DEVELOPMENT OF AN INTELLIGENT MODEL FOR THE ENERGY OPTIMIZATION OF A SMALL HOUSE

Javier Asensio Blázquez

Universidad Pontificia Comillas - ICAI, Madrid, España

Abstract:

This article presents a model capable of establishing the optimal operation of a Heating, Ventilation, Air Cooling system (HVAC system) in order to achieve energy savings and thus reduce electricity costs. The model was designed for a small house to simplify the results, but with proper parameter adjustment, it could be extrapolated to any building, provided the necessary resources are available. The model combines Machine Learning (data ingestion, processing, and future predictions) developed in Python, along with C language code that synchronizes actuators and sensors. The data is stored and interconnected between both codes through an SQL database.

Keywords: HVAC system, Machine Learning, Python, SQL.

1. Introduction:

Nowadays, electricity prices in Spain and across Europe are highly volatile due to both the diversity of the energy mix and the market structure itself. Electricity supplied to users is generated from various sources, such as solar radiation (solar energy), wind speed (wind energy) natural gas combustion (Combined cycles), potential and kinematic energy of water (hydropower), and the energy contained in atoms (nuclear energy).

Adding to everything just mentioned, we have got a marginalist system, in which the last technology needed to meet the demand for the following day will set the price for all others, which gives as a result higher fluctuations in electricity prices. This has resulted in extreme situations, such as those recorded in 2025: a maximum price of 225 € / MWh on January 15th and a minimum price of 15 € / MWh on May 11th, representing a difference of 1400% in just a few months, which is a quite significant difference for some. Moreover, on March 8th, the highest historical electrical price was recorded in Spain, reaching a mean electricity price of 544,98 € / MWh, amid the international energy crisis.

Such high variability in electricity prices creates uncertainty for companies and users, which makes them look for different strategies in order to cut expenses. The solution proposed in this article is to use an intelligent model for electricity consumption management. The application consists of using Big Data Tools, predictive algorithms from Machine Learning and advances control that will allow every user to anticipate high prices, optimize the use of resources such as energy storage, and prioritize the cheapest time slots for equipment operation. In this way, a balance will be achieved between user comfort and energy savings.

1.1. Objective:

The main objective of this project is to maximize energy savings without compromising user comfort conditions. To achieve this goal, a system composed by two interconnected modules is proposed, described in detail in the System Architecture section.

This model will have historical data and real-time data. Both data are quite important for the predictive model, due to the historical data will train the model and help adjust the most significant parameters and real-time data will be needed to predict the future and anticipate any kind of fluctuation.

1.2. System Operation Steps

To achieve the proposed objective, the systems is structured in the following way:

1. Data collection (Historical and real-time):

The system first gathers historical data such as local weather, climate records for your location, electricity price trends and home schedules. In addition, real-time information is incorporated, including meteorological forecast from AEMET and electricity demand curves and hourly electrical price forecasts from Red Eléctrica.

2. Data Analysis

The collected information is processed to remove inconsistencies, normalize values and extract useful patterns. This step is essential to provide reliable input for the prediction algorithm. When the information is not processed properly, the algorithm only learns existing patterns, it cannot anticipate to any other pattern and the model will be useless.

3. Optimization and decision-making

Once data is transformed into information, which means, data is now processed and can be used properly; the system will predict which is the optimal operation strategy for the intelligent system for the upcoming hours. These predictions involve: setting airflow rates and supply air temperatures for each zone, when to charge any existing battery… to minimize energy costs while maintaining user comfort and needs.

4. Real-Time Monitoring

A distributed sensor network continuously monitors indoor conditions to ensure compliance with user-defined comfort parameters. In case deviations or any unexpected behaviour is captured, the system immediately takes corrective action by adjusting these new relevant parameters.

5. Data Storage

At the end of each operational cycle or working day or just day, the system will upload all sensor data and operational outcomes into the SQL database. This information will be used to improve future predictions, avoid recurring issues, recalculate parameters… which will result into a continuous quality improvement.

6. Reports and recommendations

The system generates detailed reports comparing forecasted results with actual performance. These reports provide valuable feedback, support continuoi¡us improvement and guide both system administrators and end users in making informed energy management decisions.

2. Problem Statement and Assumptions

The central problem address here is the optimization of energy consumption in any kind of building through an intelligent management of HVAC systems. HVAC systems commonly operate by static schedules or manual adjustments; this type of management does not adapt to electricity prices fluctuations, climate changes nor occupancy variations. This results in suboptimal performance, higher operating costs and unnecessary energy consumption. The challenge is therefore to design a predictive and adaptative system capable of minimizing electricity costs while maintaining users thermal configurations.

2.1. Input Data and Constraints

The model is based on both historical and real-time data sources, which are stored and processed through an SQL database. The main categories of inputs are:

- Historical data: Weather conditions, electricity prices, occupancy records and previous system operation logs.

- Real-time data: Meteorological forecasts, hourly electricity predictions and sensors measurements of temperature, humidity and airflow inside the building.

- System operating limits: predefined technical constraints of HVAC systems, such as minimum and maximum airflow, temperature ranges and safety thresholds.

The system must comply with the following hard constraints:

- Equipment feasibility: setpoints must remain within manufacturer specifications to avoid unsafe operations

- User comfort conditions: indoor temperature and humidity must remain within defined comfort ranges.

- Temporal restrictions: optimization must operate on an hourly basis, anticipating short-term fluctuations in both demand and prices.

2.2. Optimization Objectives

The optimization seeks to:

1. Minimize electricity costs by anticipating high-price periods and shifting energy consumption to lower-price intervals.

2. Maintain user comfort by ensuring compliance with predefined thermal comfort standards.

3. Ensure efficient resource usage, preventing unnecessary operation of HVAC equipment and avoiding oversizing of control actions.

2.3. Problem Complexity

The HVAC optimization problem is a high-dimensional, dynamic and multi-objective task. It requires simultaneously handling prediction (room temperature conditions), decision making (optimal setpoints), and control (real-time actuator commands). A monolithic deterministic formulation would be computationally expensive and difficult to adapt to evolving operating conditions.

To address this, the proposed model adopts a hybrid strategy:

Module 1 (Python + Machine Learning): forecasts thermal demand and energy prices, generating optimal setpoints.

Module 2 (C + Control Logic): validates and applies setpoints in real-time, ensuring compliance with system and comfort constraints.

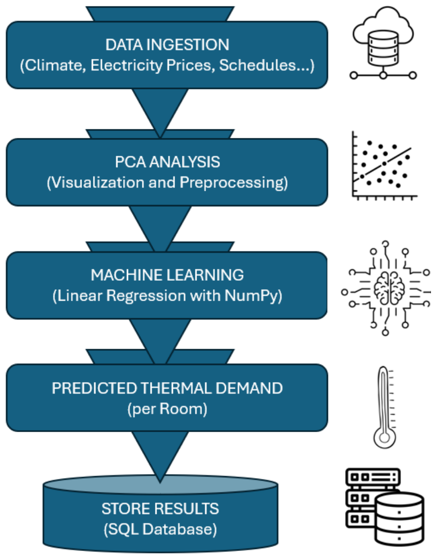
SQL integration: ensures continuous feedback, retraining of predictive models and seamless communication between predictive and control modules.

3. System Architecture

The proposed system architecture is based o a modular design which integrates three fundamental components: a predictive and optimization module (MODULE 1), a control and execution module (MODULE 2) and a central communication and storage layer (SQL Database). This architecture ensures scalability, robustness and transparency, while allowing integration of advanced data analytics with real-time actuation.

At a high level, the model is designed to transform raw environmental and energy data into optimized operational strategies that are then executed by the HVAC equipment in a safe and efficient manner. The modular approach allows each layer to evolve independently while maintaining interoperability through the SQL layer.

MODULE 1: Climate and Energy Market Analysis + Decision-Making (Python + Machine Learning):



This module represents the intelligence of the system, focusing on data ingestion, preprocessing, prediction and decision-making. Implemented in Python, it leverages scientific libraries (NumPy, Pandas, Scikit-learn) and Machine Learning techniques to forecast both thermal demand and electricity prices fluctuations.

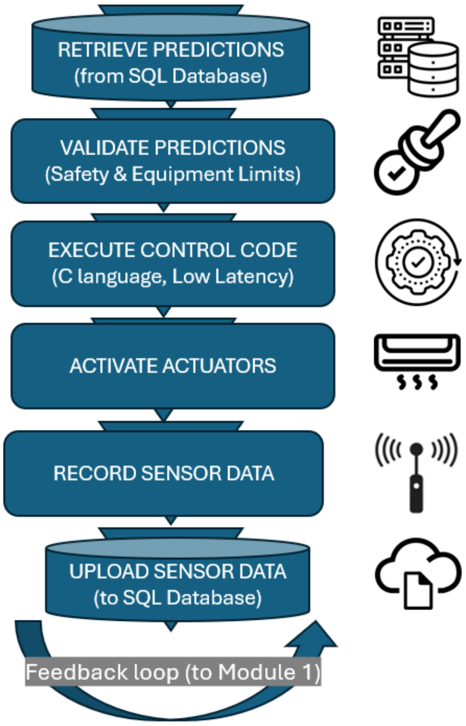
- Scope of Data: Module 1 manages heterogeneous sources, including historical weather records, electricity prices, occupancy patterns and operational logs of the HVAC system. In parallel, it incorporates real-time data streams such as meteorological forecast, hourly electricity prices updates and sensor measurements from Module 2.

- Core Functions:
  - Preprocessing and feature extraction to ensure data quality via Principal Component Analysis (PCA).
  - Development of predictive models to estimate near-future thermal loads in each zone.
  - Generation of optimal setpoints (temperature, humidity, airflow, schedules), which are validated against manufacturer limitations and minimize energy costs while maintaining thermal comfort.
  - Publication of these setpoints into the SQL Database, ensuring they are available for real-time execution in Module 2.

- Design Considerations: The module is conceived as scalable and adaptable, allowing additional algorithms, such an example could be neural networks, to be integrated in the future without altering the global architecture.

MODULE 2: Local Management and Real-Time Control (C + Sensors/Actuators):

This module is the execution arm of the system. Implemented in C to guarantee low-latency performance, it is directly connected to the network of sensors and actuators installed in the building.



- Sensors: Measure indoor temperature, relative humidity and airflows; providing continuous feedback on environmental conditions.

- Actuators: Control HVAC equipment (fans, valves, compressors) according to the setpoints defined in Module 1.

- Control Logic: Second validation of the predicted setpoints against manufacturer limitation and adding safety thresholds. Once validated, this control logic executed them in real-time, ensuring a safe and efficient system operation.

- Monitoring Logic: It looks for deviations or unexpected behaviours, generating corrective actions locally when necessary.

- Data Interface: All collected sensor data and equipment states are uploaded to the SQL Database, where they become available for analysis and model retraining in Module 1.

This module 2 also acts as a protective layer, ensuring that no unsafe or unfeasible commands are applied to the system while providing high resolution monitoring of real conditions.

SQL DATABASE: Central Communication and Data Repository:

The SQL Database acts as the integration backbone of the architecture. It provides a shared, structured and persistent data layer that enables reliable communication between Modules 1 and 2.

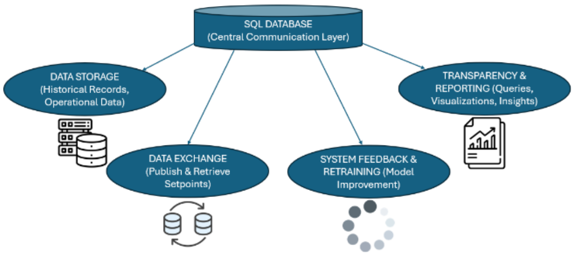
- Data Storage: Maintains long-term historical records (weather, occupancy, electricity prices, system performance) as well as real-time operational data. This kind of information is stored in structural relational tables. This dual storage ensures that predictive models have access to both enriched historical datasets and up-to-date real conditions.

- Data Exchange: Operates as the central interface for publishing and retrieving setpoints, forecasts and control signals. Python Code in Module 1 generates optimized strategies and publish them into the SQL tables, while C Code in Module 2 receives the published information, reads , validates and executes it.

- Feedback and Continuous Improvement: Sensor fededback, performance indicators and actual energy consumption are constantly uploaded to the database, enabling Module 1 to refine its predictive models through continuous retraining.

- Transparency and Reporting: The SQL layer also serves as a source for generating reports and visualization, comparing predicted values with real measurements. System administrators and end users can query the database to obtain insights into building performance, energy/economic savings and anomaly detection.

3.1. Architecture Strength



This modular system architecture is designed with several key strengths:

1. Separation concerns: Each module has a clearly defined role (prediction, control and communication). This reduces complexity and improves maintainability.

2. Scalability: Additional data sources, predictive algorithms or building subsystems can be incorporated without altering the global framework.
3. Robustness and Safety: The validation layer is contained in both Modules which ensures that only feasible commands are executed, protecting both equipment and users.

4. Continuous adaptation: The feedback loop supported by the SQL database enables continuous learning, allowing the system to improve performance over time.

5. Transparency: The architecture facilitates traceability of decisions, making it possible to audit predictions, control actions and achieved results.

4. End-To-End WorkFlow

The interaction between the modules and the SQL database can be summarized as follows:

1. Training (Python Code + Machine Learning): Historical data stored in SQL is used to calibrate model parameters.

2. Prediction (Python Code + Machine Learning): Once the model is trained, real-time information is used in order to predict and generate optimal setpoints.

3. Publish (Python to SQL): As mentioned communication between modules in quite important, without communication, the project is impossible. Actuators parameters are predicted, published and stored in SQL database.

4. Receiving information (SQL to C): When the parameters are uploaded/published, in can be downloaded/received by the C code,

read and change any parameters out of range.

5. Execution (C Code): Setpoints/parameters are retrieved, validated and applied to the system equipment via actuators, while real-time measurements are collected from sensors.

6. Feedback (C to SQL to Python): Sensor data and performance results are generated, published to the SQL database, send to th Python Code as real-time data. This creates a feedback loop with continuous predictions and communication.

7. Continuous improvement: The predicted model is periodically retrained using the enriched dataset, increasing system robustness and effectiveness.

In conclusion, this architecture leverages the strengths of each technology, which is: Python for analytics, C for control and fast executions and SQL for integration. Every section combined delivers an intelligent, adaptive and energyefficient management system.

5. Experimental Setup

The proposed model was validated through simulations on a small house in Madrid. The setup included:

- Building model: Calculus of thermal loads (summer and winter) following RITE and CTE standards.
- Modules
  - Module 1: Climate and Energy Market Analysis + Decision-Making (Python + Machine Learning).
  - Module 2: Local Management and Real-Time Control (C + Sensors/Actuators).
  - SQL Database.
- Data sources:
  - Historical datasets: Weather conditions, electricity prices, and energy consumption or the last 5 years.
  - Real-time and future data: AEMET meteorological forecast and occupancy expected for the next week; hourly price curves and electrical demand for the next day (information obtained from Red Electrica); and indoors sensors measurements (temperature, humidity, airflow).
- Constraints applied:
  - Comfort ranges: 25 ºC ± 1 ºC for summer; 21 ºC ± 1 ºC for winter; Relative Humidity between 45-50%.
  - Equipment limits: 1 inverter air-to-air heat pump with 5 kW cooling and 6 kW heating nominal; Supply air temperature: cooling 10-16ºC and heating 30-45ºC; airflow limits 300 m^3 /h.
  - Temporal restrictions: hourly optimization cycles or changing to manual when needed.
- Evaluation metrics:
  - Energy cost reduction vs baseline static schedule, as well as energy consumption.
  - Comfort compliance: percentage of time indoor temperature and humidity remain within predefined comfort ranges.
  - Execution time per optimization cycle.
  - Adaptability under weather and market fluctuations.
  - Prediction Accuracy: mean error between the temperature predicted and actual temperature measured by sensors.

6. Results and Evaluation

The intelligent HVAC optimization model was tested through several simulations and optimizations having as base model a small house in Madrid, using both historical and real-time datasets. The evaluation focused on how the model affects the energy efficiency, comfort compliance, prediction accuracy and system adaptability.

- Energy savings: The system was able to reduce 15% of electricity costs compared to a baseline static schedule. Savings were significantly seen during peak price periods, where load shifting to cheaper hours reduced operational costs.

- Comfort compliance: Indoor temperature remained within the predefined comfort range during 96% of the total operating time. Relative humidity was controlled within the 45-50% range, with only minor deviations during abrupt weather changes.

- Prediction Accuracy: The average error between the predicted and measured indoor temperature was ± 1 ºC. Forecasts of short-term energy demand aligned well with actual consumption, supporting reliable decision-making. - Computational Performance: Each optimization cycle was executed in under a minute, demonstrating feasibility for near real-time applications. SQL-based communication ensured fast data exchange communication between Python-Module 1 and C-Module 2.

- System Robustness and Adaptability: The model successfully adapted to unexpected fluctuations in both electricity prices and weather conditions.
  Continuous retraining improved the reliability of the predictions made over time.

7. Future Works

While the proposed model has demonstrated promising results in reducing energy consumption and maintaining comfort, there are several potential extensions which could enhance its effectiveness and broaden its applicability:

- Increased Energy Savings:

With further system upgrades, the model could achieve up to 25% reduction in energy consumption under typical residential conditions.

When integrated with battery storage systems, energy savings could increase to 30-35% as the optimizer would be able to store energy during low-price periods and release it during peaks.

- Integration of Advanced Algorithms:

More sophisticated models, such as deep learning or neural networks, could be employed to improve forecasting accuracy of thermal loads and electricity prices. However, these approaches would likely increase computational requirements, with optimization cycles potentially lasting several minutes instead of seconds.

- Renewable Energy Integration:

Incorporating on-site renewable sources such as solar PV could enhance the optimizer by aligning HVAC consumption with local generation.

This could reduce dependency on the grid and provide additional flexibility in managing costs.

- Cybersecurity and Data Privacy:

As the system depends on real-time data integration and SQL storage, ensuring data integrity privacy and resilience against cyberattacks will be an important area of future work.

8. Conclusions

This work presented the design, implementation, and validation of an intelligent model for the energy optimization of HVAC systems in residential buildings. The proposed architecture integrated three complementary components: a predictive and optimization module developed in Python using Machine Learning techniques, a real-time control module implemented in C for execution and safety validation, and an SQL database for communication, storage, and feedback.
The model was tested on a small house scenario in Madrid, considering both historical and real-time datasets. The results demonstrated that:

- The system achieved 15% reduction in energy costs compared to a baseline static schedule.
- Comfort conditions (temperature and humidity) were maintained within predefined ranges during over 96% of operating time. - Prediction accuracy was satisfactory, with an average error below ±1 °C between predicted and measured values.
- Each optimization cycle was executed in under 1 minute, proving the feasibility of real-time application.
- The modular design enabled robustness and adaptability, allowing the system to respond effectively to price volatility and unexpected weather fluctuations.

These findings confirm that intelligent energy management strategies can significantly reduce HVAC energy consumption while maintaining user comfort, even in small-scale residential contexts. Furthermore, the combination of Machine Learning, low-latency control, and structured database integration proved to be a solid framework for ensuring scalability, transparency, and continuous improvement.

In conclusion, the developed model successfully balances the dual objectives of cost reduction and comfort assurance. Its design makes it a promising solution not only for small houses but also for larger residential or commercial buildings, provided proper parameter adjustments and additional computational resources are made available. This work represents a step forward toward the digitalization and intelligent management of building energy systems, aligning with broader goals of energy efficiency, sustainability, and smart-grid integration.

9. References

[1] *[BOE-A-2007-15820 Real Decreto 1027/2007, de 20 de julio, por el que se aprueba el Reglamento de Instalaciones Térmicas en los Edificios. (s. f.) (BOE núm. 207, de 29 de agosto de 2007).* https://www.boe.es/buscar/doc.php?id=BOE-A-2007-15820

[2] *BOE-A-2006-5515 Real Decreto 314/2006, de 17 de marzo, por el que se aprueba el Código Técnico de la Edificación (CTE) (BOE núm. 74, de 28 de marzo de 2006).* https://www.boe.es/buscar/act.php?id=BOE-A-2006-5515

[3] Red Eléctrica de España (REE). *Hourly electricity demand and day-ahead market price forecasts*. Available: https://www.ree.es

[4] Agencia Estatal de Meteorología (AEMET). *Spanish Meteorological Forecasting Service*. Available: https://www.aemet.es

[5] ASHRAE. *Fundamentals Handbook*. American Society of Heating, Refrigerating and Air-Conditioning Engineers, Atlanta, GA, 2001.

[6] Carrier. *Load Calculation Manual* (Cooling and Heating Load Estimation). Carrier Corporation, Technical Manual, 2001.

[7] Pedregosa, F., et al. *Scikit-learn: Machine Learning in Python*. Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011.

[8] McKinney, W. *Data Structures for Statistical Computing in Python (pandas)*. Proceedings of the 9th Python in Science Conference, 2010.

[9] Oliphant, T.E. *A guide to NumPy*. Trelgol Publishing, 2006.

[10] Van Rossum, G., & Drake, F.L. *Python 3 Reference Manual*. Addison-Wesley, 2009.