Energy 333 (2025) 137130

Contents lists available at ScienceDirect

Energy

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

Smart buildings energy consumption forecasting using adaptive evolutionary bagging extra tree learning models

Mehdi Neshat [a,*], Menasha Thilakaratne [b], Mohammed El-Abd [c], Seyedali Mirjalili [d,e], Amir H. Gandomi [a,e], John Boland [f]

[a] Faculty of Engineering and Information Technology, University of Technology Sydney, Sydney, 2007, NSW, Australia
[b] School of Computer Science, The University of Adelaide, Adelaide, 5005, Australia
[c] College of Engineering and Applied Sciences, American University of Kuwait, Kuwait
[d] Center for Artificial Intelligence Research and Optimization, Torrens University Australia, Brisbane, QLD 4006, Australia
[e] University Research and Innovation Center (EKIK), Obuda University, Budapest, 1034, Hungary
[f] Industrial AI Research Centre, UniSA STEM, University of South Australia, Mawson Lakes, 5095, Australia

ARTICLE INFO

Keywords:
Smart building
Energy forecasting
Deep learning
Ensemble learning
Extra tree
Optimisation
Hyper-parameter tuning

ABSTRACT

Smart buildings are gaining popularity because they have the capability to enhance energy efficiency, lower costs, improve security, and provide a more comfortable and convenient environment for building occupants. A considerable ratio of the global energy supply has been consumed in building sectors and plays a pivotal role in the future decarbonisation pathways. In order to manage energy consumption and improve energy efficiency in smart buildings, developing reliable and accurate energy demand forecasting is crucial and meaningful. However, extending an effective predictive model for the total energy use of appliances at the buildings' level is challenging due to temporal oscillations and complex linear and non-linear patterns. This paper proposes three hybrid ensemble predictive models, incorporating Bagging, Stacking, and Voting mechanisms combined with a fast and effective evolutionary hyper-parameters tuner. The performance of the proposed energy forecasting model was evaluated using a hybrid dataset of meteorological parameters, energy use of appliances, temperature, humidity, and lighting energy consumption from different sections collected by 18 sensors in a building located in Stambruges, Mons in Belgium. In order to provide a comparative framework and investigate the efficiency of the proposed predictive model, 15 popular machine learning (ML) models, including two classic ML models, three Neural Networks (NN), a Decision Tree (DT), a Random Forest (RF), two Deep Learning (DL) and six Ensemble models, were compared. The prediction results indicate that the adaptive evolutionary bagging model surpassed other predictive models in both accuracy and learning error. Notably, it delivered accuracy gains of 12.6%, 13.7%, 12.9%, 27.04%, and 17.4% when compared to Extreme Gradient Boosting (XGB), Categorical Boosting (CatBoost), Gradient Boosting Machine (GBM), Light Gradient Boosting Machine (LGBM), and RF.

Nomenclature

See Table 1.

1. Introduction

One-third of the world's primary energy is approximately consumed by buildings [1]. Buildings are a significant contributor to carbon dioxide (CO2) emissions, accounting for nearly 39% of such emissions [2]. Due to this high level of buildings' energy consumption contribution to global energy demand, developing smart buildings is crucial. There are numerous advantages in advancing smart buildings, such as enhanced energy optimisation, augmented residents' satisfaction and productivity [3], as well as improved health and well-being [4]. These benefits have been achieved due to hiring cutting-edge technologies such as artificial intelligence (AI)-based methods, deep neural networks (DNNs) [5], and adaptive learning controls in smart buildings [6], which enable such facilities to control various systems (cooling, heating, cooking, etc. [7]) to evolve more efficient in terms of energy and comfort [8]. Furthermore, smart buildings prioritise indoor air quality, ensuring thermal, acoustic, and visual comfort.

* Corresponding author.
E-mail addresses: mehdi.neshat@uts.edu.au (M. Neshat), menasha.thilakaratne@adelaide.edu.au (M. Thilakaratne), melabd@auk.edu.kw (M. El-Abd), ali.mirjalili@laureate.edu.au (S. Mirjalili), gandomi@uts.edu.au (A.H. Gandomi), John.Boland@unisa.edu.au (J. Boland).

https://doi.org/10.1016/j.energy.2025.137130
Received 16 December 2024; Received in revised form 11 June 2025; Accepted 13 June 2025
Available online 11 July 2025

Table 1
Summary of key abbreviations used in the manuscript for clarity.

Abbreviation	Full name
AI	Artificial intelligence
ANN	Artificial Neural networks
Bi-LSTM	Bidirectional Long short-term memory network
BIM-DB	Building information modelling-design builder
BIM	Building Information Modelling
BS	Batch size
CART	Classification and regression tree
CatBoost	Categorical Boosting
CR	Probability crossover rate
CL	Cooling load
CMA-ES	Covariance matrix adaptation evolution strategy
CNN	Convolutional neural network
DDPG	Deep Deterministic Policy Gradient
DE	Differential evolution
DNN	Deep neural networks
DT	Decision Tree
EA	Evolutionary Algorithm
ELM	Extreme Learning Machine
EVS	Explained variance score
GA	Genetic algorithm
GBT	Gradient boosting tree
GBM	Gradient Boosting Machine
GC	Generalised correntropy
GPT	Generative Pre-trained Transformers
GRU	Gated recurrent unit
HGBR	Histogram-Based Gradient Boosting Regressor
HL	Heating load
HVAC	Heating, Ventilation, and Air Conditioning
IoT	Internet of Things
LGBM	Light Gradient Boosting Machine
LOF	Local outlier factor algorithm
LRD	Local reachability density
LHTES	Latent heat thermal energy storage
LSTM	Long short-term memory network
MAE	Mean absolute error
ML	Machine learning
MLP	Multi-layer perceptron
MSE	Mean square error
NSGA	Non-dominated Sorting Genetic Algorithm
NM	Nelder–Mead simplex direct search method
PSO	Particle Swarm Optimisation
PHPP	Passive House Planning Package
RF	Random Forest
RIME	Rime optimisation algorithm
RMSE	Root mean square error
RNN	Recurrent neural networks
SCO	Sine cosine optimisation
SMAPE	Symmetric mean absolute percentage error
SVM	Support vector machines
XGB	Extreme Gradient Boosting

In smart buildings, to enhance communication and information sharing, technologies such as the Internet of Things (IoT), Building Information Modelling (BIM), and Blockchain have been incorporated to improve security and management [9]. Another significant advantage of developing smart buildings is their contribution to the energy sector decarbonisation [10] by supporting the electrical grid through providing demand response functionality [11] and balancing electricity demand with non-dispatchable renewable energy sources [12].

In the last two decades, various ML techniques have experienced significant growth, particularly in modelling energy consumption in smart buildings. This surge of interest can be attributed to the remarkable efficacy and robustness exhibited by ML predictors in this field. Impressively, ML models have demonstrated exceptional generalisation and flexibility abilities [13], making them widely pertinent to a diverse range of problems. They have been hailed as "universal function approximators" because of their unparallelled adaptability. A comprehensive review of the rapid advancements in Artificial Intelligence (AI) and ML models within the context of smart buildings has yielded a meaningful conclusion [14] and determined that the overall adaptability of buildings to unforeseen changes can be significantly enhanced through the enactment of AI-driven learning processes. Moreover, integrating adaptability solutions on the timescales of heating, ventilation, and air conditioning (HVAC) control and electricity market participation has been identified as the most promising avenue for achieving substantial improvements in energy efficiency.

One pivotal advantage of employing ML models lies in their aptitude for analysing extensive datasets and uncovering intricate patterns that elude traditional statistical methodologies. By considering an array of factors, such as construction characteristics, occupancy patterns, and weather states, these models offer accurate predictions of energy usage within buildings [15]. This capability stems from their capacity to process vast volumes of data and discern hidden correlations that would otherwise remain inconspicuous. Moreover, the prevalence of multiple sensors for data collection in smart buildings necessitates the development of real-time systems for monitoring, controlling, predicting, and optimising total power consumption. ML models excel in this arena by continuously analysing sequential data and constructing precise models of these dynamic systems [16]. Through incessant monitoring and data analysis, these models can adapt control settings for Heating, Ventilation, and Air Conditioning (HVAC) systems, lighting, and other building components to attain desired energy efficiency targets. Recently, Lie et al. [17] proposed a novel HVAC control system for intelligent buildings that uses a multi-step predictive deep learning model to reduce power consumption costs while maintaining user satisfaction. The system combines Long Short-term Memory (LSTM), generalised correntropy (GC) loss function, and Deep Deterministic Policy Gradient (DDPG) for predicting house temperature and dynamic power adjustment. Simulation results showed over 12% cost savings compared to alternative approaches.

Another compelling rationale for incorporating machine learning (ML) models in energy demand modelling for smart buildings lies in their forecasting capabilities. By leveraging historical data, weather forecasts, and other relevant characteristics, ML aids in accurately predicting future energy demands [18]. This proficiency in demand forecasting facilitates superior planning for energy generation, distribution, and load management, culminating in a more dependable and efficient energy supply. These factors collectively enable the optimisation of energy utilisation, enhance operational efficiency, and contribute to the establishment of sustainable [19] and intelligent building systems.

Somu et al. [20] proposed a hybrid building power consumption model (kCNN-LSTM) consisting of LSTM, a Convolutional neural network (CNN) combined with a K-means clustering method and sine cosine optimisation (SCO) algorithm [21] to tune the hyper-parameters of LSTM. The kCNN-LSTM model outperforms existing demand forecast models and offers precise energy consumption prediction. An automated building energy load forecasting methodology [22] has recently been introduced based on Generative Pre-trained Transformers (GPT) in combination with prompt optimisation, external knowledge use, and self-correction. The method effectively mitigates technical barriers to entry for non-experts and permits precise low-budget energy prediction. It was compared with actual test buildings and proved to have a mean R2 of 0.95, demonstrating the engineering viability of mass language models for smart building energy management innovation.

While ML models have been shown to be promising for the prediction of building energy consumption, current research focuses mainly on short-term prediction. It seldom introduces new parameters to improve prediction accuracy. To address this gap, a team developed a data-driven method [23] to predict the hourly energy consumption of a university office building by integrating meteorological, temporal, and an introduced meta-parameter, air conditioning demand. Five ML algorithms (Random Forest (RF), Gradient Boosted Trees (GBT), Support Vector Machines (SVM), Artificial Neural Networks (ANN), and Deep Neural Networks (DNN)) are compared and experimental results show that DNN provide the best performance (Root mean square

error (RMSE) = 4.796 kWh, Mean Absolute Percentage Error (MAPE) = 5.738%), outperforming existing methods. Incorporating the air conditioning demand parameter significantly enhances model accuracy for every algorithm.

Ensemble models offer excellent benefits in building energy prediction [24] by exploiting the strengths of different algorithms, enhancing prediction accuracy and generalisability compared to individual models. While much attention is being given now, most prior studies have focused on single ML models or basic ensemble techniques without fully harnessing stacked architectures for heating and cooling load (HL and CL) prediction. Furthermore, there has been limited research in the literature on integrating hyperparameter-tuned models with heterogeneous base models for predicting residential building energy. Closing these gaps, in a recent work [25], a stacked ensemble model was introduced integrating XGB, DT, RF, and Bayesian optimisation for hyperparameter tuning. Closing these gaps, a recent work [25] introduced a stacked ensemble model that integrates XGB, DT, RF, and Bayesian optimisation for hyperparameter tuning. The suggested model performed considerably better than the traditional techniques, providing better performance (RMSE of 0.484 for HL and 0.948 for CL). Another example of ensemble models is [26] which proposes a stacked learning model for predicting the dynamic performance of PCM-based double-pipe latent heat thermal energy storage (LHTES) units. Main contributions include sensitivity analysis for variable selection, a two-stage ensemble model combining Regression Trees, SVR, and Linear Regression, and comprehensive validation across datasets and phase change stages. The proposed infrastructure demonstrated a 7.82% improvement in MAPE, a 25.6% increase in stability, and a 9.7% reduction in peak demand for heating, ventilation, and air conditioning (HVAC) systems, contributing to more flexible, data-driven building energy management. Another study [27] suggested a stacking ensemble learning model for home net load-interval prediction, which combines k-means user clustering, LRIME-based optimisation, and bootstrap interval estimation. Their main contributions included developing interpretable interval forecasts, recommending the rime optimisation algorithm (LRIME) for improved performance, and adding LSTM, XGBoost, and ELM as optimised base learners. Australian Ausgrid data tests confirm the model's improved accuracy, robustness, and uncertainty estimation over state-of-the-art models.

Combining ML models with optimisation methods is one of the popular techniques used to forecast energy consumption in buildings. To address the lack of integrated prediction and optimisation methods in green building design, a recent study [28] proposed a framework combining BIM-DB simulation, Bayesian-Random Forest (Bayesian-RF) prediction, and Non-dominated Sorting Genetic Algorithm (NSGA-III) optimisation. BIM-DB efficiently generates building performance data, while Bayesian-RF achieves high prediction accuracy ($MSE < 0.08$, $R2 > 0.85$). The prediction model guides NSGA-III to optimise energy use, emissions, cost, and thermal comfort. A case study conducted on a teaching building demonstrated reductions of 7.68% in energy consumption, 6.48% in carbon emissions, and 1.77% in operational costs while also enhancing occupant comfort. Current approaches to optimising public building sustainability struggle to reconcile competing goals and integrate expert knowledge with data-driven forecasting. A recent study [29] suggested a hybrid approach that blended building information modelling-design builder (BIM-DB) simulations with a BO-CatBoost-NSGA-III algorithm to overcome these limitations. Their major contributions included a two-stage knowledge a data-driven approach to secure dataset generation, a BO-optimised CatBoost model with $R2 > 0.97$ across targets, and finally, multiobjective optimisation using NSGA-III, which delivered 32.20% lower energy consumption, 48.77% lower CO2 emissions, 60.69% improved thermal comfort, and 15.45% less glare.

1.1. Research gaps

Sequential ML models, such as LSTM, BiLSTM, CNN-LSTM, etc., have gained recognition for their success in these specific domains [30]. However, they do come with certain drawbacks that need to be considered as follows. One notable disadvantage is the complex architecture of these models, which can result in extensive training runtimes, mainly when dealing with large-scale datasets. Consequently, the computational requirements for training these models can be substantial. Moreover, achieving optimal performance with these models heavily relies on careful design and parameter tuning. Improper choice of hyper-parameters can lead to suboptimal performance or overfitting, underscoring the need for meticulous attention during the model configuration phase. Another drawback is the need for more interpretability of LSTM and its family models. These models are often considered black boxes, making it challenging to comprehend the underlying reasoning behind their predictions. Interpreting the learned representations and understanding the critical features becomes a non-trivial task. Furthermore, when faced with limited data, these sequential models may struggle to extract meaningful patterns and achieve optimal performance [31]. Uncovering hidden patterns and dependencies relies heavily on the availability of sufficient training examples, which can be a limitation in scenarios where data is scarce.

Considering these drawbacks is crucial when deciding whether to employ LSTM, BiLSTM, or CNN-LSTM models. The trade-off between their success in specific domains and the associated challenges of training runtime, parameter tuning, interpretability, and data limitations should be carefully evaluated to ensure the most suitable approach for a given application.

Furthermore, despite notable advancements in ML and ensemble-based approaches for smart building energy forecasting, several research challenges and gaps remain unresolved:

- Limited integration of heterogeneous ensemble strategies: While individual ensemble techniques such as bagging, boosting, and stacking have shown promise, most existing studies rely on singular strategies. Few attempts have been made to systematically combine these approaches within a unified hybrid framework to leverage their complementary strengths.
- Insufficient use of advanced hyper-parameter optimisation: Many prior works employ default or manually-tuned parameters, which may result in suboptimal model performance. The integration of meta-heuristic optimisation algorithms, such as Genetic Algorithms (GA), Differential Evolution (DE), or 1+1 Evolutionary Algorithms for automated and adaptive hyperparameter tuning remains underexplored in this domain.
- Neglect of real-world temporal and environmental complexity: Existing models often oversimplify input features or overlook dynamic environmental factors, such as temporal variability, sensor heterogeneity, and inter-feature dependencies. There is a need for models that can robustly learn from multivariate, high-resolution data collected via Internet of Things (IoT) sensors in actual smart building environments.
- Lack of comprehensive benchmarking with modern deep and ensemble models: Although deep learning models (e.g., CNNs, LSTMs) and gradient-boosting methods (e.g., XGBoost, CatBoost, LGBM) are increasingly adopted, few studies conduct extensive comparative analyses involving a broad spectrum of baseline models across classical ML, deep networks, and ensemble methods under consistent evaluation metrics.
- Moreover, limited focus on model generalisability and robustness: Many forecasting models are tailored to specific datasets or settings, raising concerns about their adaptability across different buildings or climatic regions. There is a gap in assessing generalisability through cross-validation techniques and testing on diverse time periods or unseen environments.

- Last but not least, sparse consideration of interpretability and computational trade-offs: Highly accurate models such as deep networks or ensemble stacks often lack interpretability and incur high computational costs. Few studies explicitly address the trade-off between model complexity, transparency, and real-time applicability, especially in the context of building management systems.

1.2. Key contributions

To address the aforementioned challenges, in this study, we propose a hybrid learning model specifically designed for predicting the total power usage of compliances in a Stambruges, Mons, Belgium building. The model incorporated three ensemble mechanisms: Bagging, Stacking and Voting models, as well as a fast and effective Evolutionary framework. The study's primary objective was to develop a robust and accurate model for predicting power consumption in smart buildings. To achieve this, data collected from 18 sensors installed in the building was used to capture meteorological parameters, energy use of appliances, temperature, humidity, and lighting energy consumption of different sections. The main contributions of this study are listed as follows:

- Comprehensive data analysis was conducted to extract various characteristics and correlations among the collected features and power consumption. This analysis provided valuable insights into the relationships between different variables, helping to inform the development of the predictive model.
- A wide range of machine and deep learning models were implemented and compared to ensure the most efficient learning model. This included classic ML models such as DT and RF, as well as various Neural Networks (NN) and Ensemble models. By developing this comprehensive comparative framework, the designers will be able to identify the most effective learning model for predicting power consumption in the smart building context.
- Further, the study addressed the challenge of hyper-parameter tuning initialisation, which can significantly impact the model's performance. To overcome this challenge, four optimisation methods were tested and compared to improve prediction accuracy and reduce modelling training errors. The aim was to find a practical and smart hyper-parameter tuner that would enhance the overall performance of the power consumption prediction model.
- Finally, this study contributes to the field of smart buildings by proposing an adaptive evolutionary ensemble learning model that leverages the power of various ML and tree-based techniques combined with a fast and effective Evolutionary algorithm. To this end, we developed and evaluated six Voting models, eight Bagging models, and ten Stacking architectures, each composed of different configurations of decision trees, gradient-boosted methods, and neural learners. The comprehensive data analysis, extensive model comparison, and optimisation methods employed in this study provide valuable insights and techniques for accurately predicting power consumption in similar smart building scenarios.

The remainder of this paper is organised as follows. Section 2 introduces the dataset and presents a detailed statistical analysis. Section 3 outlines the methodological framework, encompassing outlier detection, ensemble learning strategies, and optimisation techniques. Section 4 presents the experimental results and compares model performance. Moreover, Section 5 discusses the key findings and their implications. Finally, Section 6 concludes the study by summarising the research contributions, acknowledging its limitations, and outlining directions for future work.

2. Data sets and statistical analysis

The hybrid dataset utilised in this study was obtained from a residential property in Stambruges, Belgium, approximately 24 km from the City of Mons [32]. The house's construction was completed in December 2015, incorporating entirely new mechanical systems. The architectural design followed the principles of passive house certification [33], which entails limiting the annual heating and cooling loads to a maximum of 15 kWh/m^2 per year, as determined by design software (Passive House Planning Package (PHPP)). It is worth highlighting that in September 2016, the building's air leakage was assessed and measured to be 0.6 air changes per hour at 50 Pa. A heat recovery ventilation unit with an efficiency ranging between 90% and 95% is employed to ensure proper ventilation. The total floor area of the house amounts to 280 m^2, with the heated area encompassing 220 m^2. The map of two floors of the building [32] with the location of sensors to record temperature and humidity (see Fig. 1).

Electrical energy consumption in the passive house was monitored using M-BUS energy counters, which captured data every 10 min. This tracking included individual power loads from the domestic hot water, devices, lighting, heat recovery ventilation unit, and electric baseboard heaters. The energy devices used correspond to the list given in Ref. [32]. An internet-based energy monitoring system collects the energy data, keeps it and dispatches notifications via email every 12 h. Lighting energy consumption constituted between 1% and 4% of the total, predominantly due to LED fixtures. Temperature and humidity conditions within the house were tracked using a wireless sensor network (ZigBee) constructed with XBeeradios, Atmega328P microcontrollers, and DHT-22 sensors. The house's large size and solid construction necessitated the inclusion of two additional XBee radios functioning as routers to facilitate effective communication from the end nodes to the coordinator. Battery-powered sensor nodes relayed information approximately every 3.3 min. The list of variables, along with their locations in the dataset, is presented in Table S1.

Table 2 presents a statistical analysis of the dataset's variables, providing a brief overview of the dataset and highlighting key characteristics, including coverage, prominent trends, and variability.

Figure S3 illustrates the distribution of the energy consumption profile over five months. The graph displays a significant variance in energy usage, ranging from zero to 1000 Wh. From a broad perspective, no discernible pattern is observed, presenting a challenging scenario for the accurate estimation of power utilisation by ML models.

Fig. 2 is a plot of the daily average time series profiles of temperature and humidity data recorded by nine sensors mounted across the interior and exterior of the smart building. Out of these, T6 and T-out represent outdoor conditions, and the remaining ones represent indoor climate measurements. The result shows that the indoor sensors display a consistent and stable thermal trend over the four-month observation period, indicating a well-managed indoor environment. On the other hand, T6 and T-out are more diverse, reflecting the effect of outside weather volatility. Overall, the average outdoor temperature, at approximately 15 °C, is considerably lower than indoor temperatures, a reflection of the quality of the building's insulation and the effectiveness of internal climate control.

In Fig. 3b, we observe the descriptive statistics of power consumption across the five-month period, specifically from January to May. Remarkably, the average power consumption in January and April is the highest among the months considered. This information provides insights into the varying power usage levels throughout the months. Besides, when comparing weekdays and weekends, Fig. 3c reveals that Thursday and Saturday are the days with the highest energy consumption. This data further highlights the distinction between energy consumption patterns on different days of the week. These graphical representations contribute to a comprehensive understanding of the energy consumption dynamics, highlighting the challenges faced by the ML model in accurately estimating power utilisation.

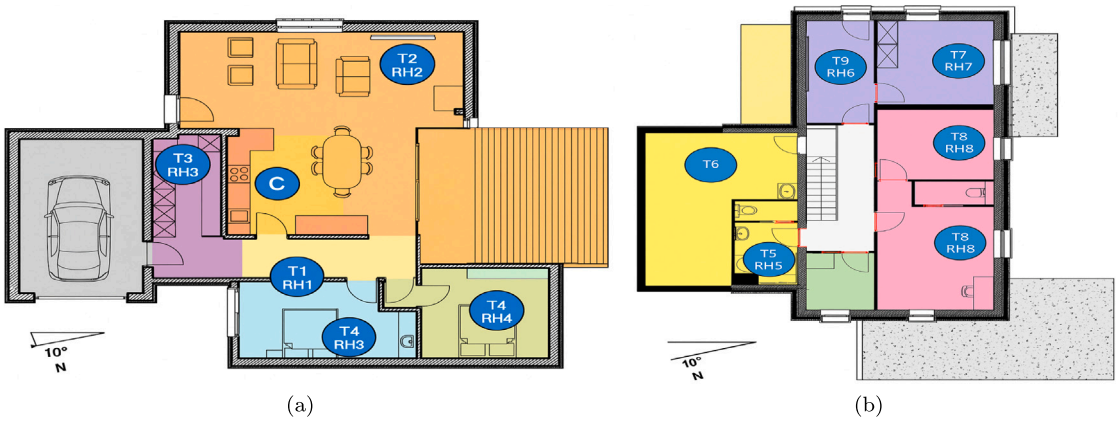

Fig. 1. The building map of (a) the First and (b) the Second floor and temperature and humidity sensors position.

Table 2
Statistical analysis of total energy consumption of the building and other features.

	Appliances	Lights	T1	RH_1	T2	RH_2	T3	RH_3	T4	RH_4	T5	RH_5	T6	RH_6
Min	10.000	0.000	16.790	27.023	16.100	20.463	17.200	28.767	15.100	27.660	15.330	29.815	−6.065	1.000
Max	1080.000	70.000	26.260	63.360	29.857	56.027	29.236	50.163	26.200	51.090	25.795	96.322	28.290	99.900
Mean	97.695	3.802	21.687	40.260	20.341	40.420	22.268	39.243	20.855	39.027	19.592	50.949	7.911	54.609
Median	60.000	0.000	21.600	39.657	20.000	40.500	22.100	38.530	20.667	38.400	19.390	49.090	7.300	55.290
STD	102.525	7.936	1.606	3.979	2.193	4.070	2.006	3.255	2.043	4.341	1.845	9.022	6.090	31.150
	T7	RH_7	T8	RH_8	T9	RH_9	T_out	Press_mm_hg	RH_out	Windspeed	Visibility	Tdewpoint	rv1	rv2
Min	15.390	23.200	16.307	29.600	14.890	29.167	−5.000	729.300	24.000	0.000	1.000	−6.600	0.005	0.005
Max	26.000	51.400	27.230	58.780	24.500	53.327	26.100	772.300	100.000	14.000	66.000	15.500	49.997	49.997
Mean	20.267	35.388	22.029	42.936	19.486	41.552	7.412	755.523	79.750	4.040	38.331	3.761	24.988	24.988
Median	20.033	34.863	22.100	42.375	19.390	40.900	6.917	756.100	83.667	3.667	40.000	3.433	24.898	24.898
STD	2.110	5.114	1.956	5.224	2.015	4.151	5.317	7.399	14.901	2.451	11.795	4.195	14.497	14.497

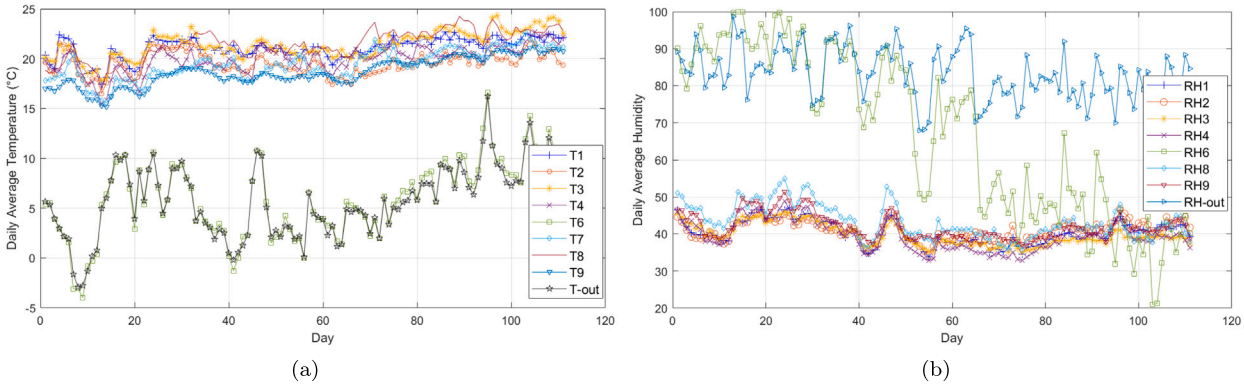

Fig. 2. Time series of daily average (a) temperature and (b) humidity recorded from sensors.

Fig. 4 depicts the average electricity usage of both devices and lights at different times. The graph reveals a considerable correlation between the two variables. Particularly, a high correlation is observed throughout the time range. However, it is noteworthy that between 12:00 PM and 6:00 PM, the average power consumption of devices surpasses that of lights. This finding aligns with expectations, as daytime usage typically involves increased activity and higher demand for related to device electricity. After 6:00 pm, a shift in the pattern becomes evident with the average power consumption of lights increases, likely corresponding to the evening hours when lighting requirements typically become more prominent. Consequently, during this period, the average power consumption of lights surpasses that of devices.

Figure S4 presents the correlation coefficient analysis between temperature variables recorded by ten sensors and the power consumption of appliances. Two noteworthy observations can be made from this analysis. Firstly, a positive correlation is observed between all indoor temperature variables and power consumption. This indicates that as indoor temperatures rise, the power consumption of appliances also tends to increase. Furthermore, there is a positive correlation among the indoor temperature variables themselves, suggesting that similar changes in the others accompany changes in one temperature variable. In contrast, the outdoor temperature variable negatively correlates with power consumption and the other indoor temperature variables. This observation implies that as the outdoor temperature rises, there is an inclination to decline in power consumption and indoor temperatures. This negative correlation likely stems from cooling systems or strategies to maintain comfortable indoor conditions despite higher outdoor temperatures. Last but not least, the highest correlation between appliances and the temperature variable T2 indicates a strong relationship between these two factors. Further, the second-largest correlation between appliances and temperature variable T6 is observed, further highlighting their interdependence.

To explore the correlation between temperature and humidity variables, and power consumption, we analysed as depicted in Figure

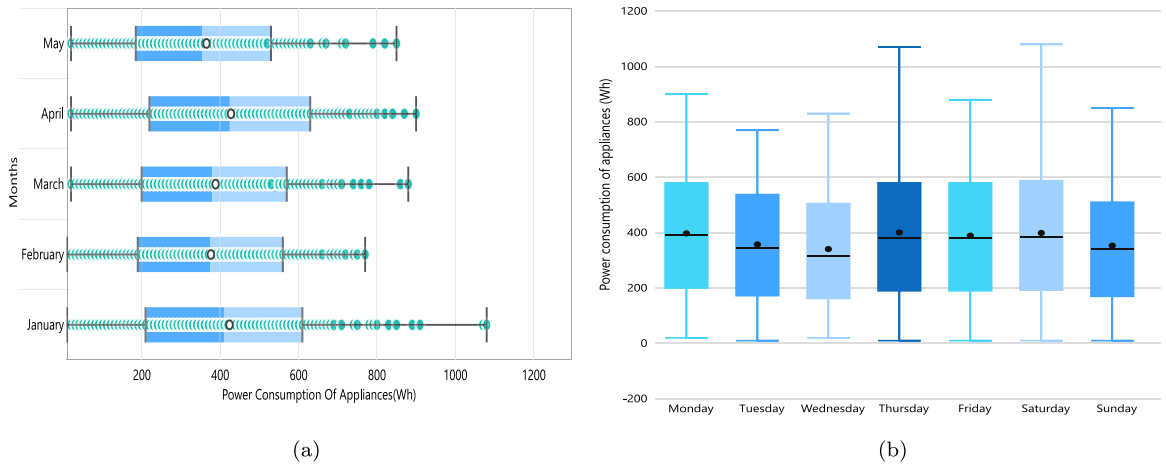

Fig. 3. (a) The distribution of consumption through five months. (b) The statistical observations for energy consumption in five months as a box-plot.

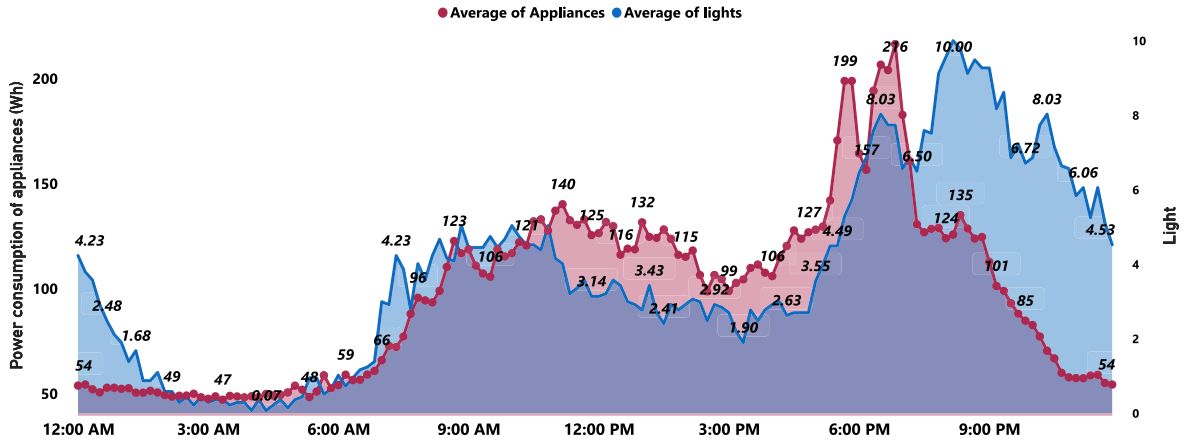

Fig. 4. The average power usage of appliances and lights between 12:00 AM and 11:59 PM.

S5. This line chart provides insights into the relationships between these variables. The chart reveals a positive correlation pattern among temperature variables, with correlations higher than those observed for humidity features. Nevertheless, most humidity variables exhibit a negative correlation with power consumption, which implies that as humidity levels increase, power consumption tends to decrease. The negative correlations observed for humidity variables highlight the influence of humidity on energy usage patterns. This negative correlation could be attributed to the impact of moisture on cooling requirements, ventilation systems, or other factors affecting power consumption.

Building on the insights gained from the statistical analysis of the smart building dataset, the next section outlines the methodological framework employed to construct and optimise predictive models.

3. Methods

In this section, the technical approaches adopted in this research are presented. Firstly, the Local Outlier Factor algorithm is presented (Section 3.1) to filter and remove outlying data points and present a high-quality dataset for model building. Secondly, the meta-heuristic algorithms (Section 3.2), including GA, DE, and the (1+1) Evolutionary Algorithm, and their details in optimisation and search abilities are emphasised. Next, ensemble learning strategies (Section 3.3) such as Stacking, Bagging, Voting, and Boosting are outlined, sharing their advantages in predictive precision, stability, and generalisability. Finally, this study introduces the Adaptive Evolutionary Ensemble Learning model (Section 3.4), which highlights its novelty and advantages over ensemble learning and evolutionary algorithms for minimising the function under adverse optimisation landscapes.

3.1. Data preprocessing and outlier detection

Local outlier factor (LOF) algorithm. To detect and remove outliers, we employed the LOF method [34], one of the most popular and effective techniques for cleaning time series data. LOF is an unsupervised, neighbourhood-based algorithm and compares each observation with k-nearest Neighbours estimates, finding the ratio density that estimates the local reachable of observation versus that over its neighbourhood; therefore, it calculates this LOF score, corresponding to an observation's average density to those neighbours. Thus, it considers outlier points whose densities are significantly lower than those of their neighbours, which is why LOF effectively identifies anomalies within datasets with varying density distributions. Eq. (1) shows the LOF computed for x observation [35]. Also, variable o is an observation to an individual nearest observation from among the k-nearest neighbours of data point x.

$$LOF_i(x) = \frac{1}{|N_i(x)|} \sum_{o \in N_i(x)}^{N} \frac{\text{LRD}\, dis_i(o)}{LRD_i(x)}, \tag{1}$$

where the local reachability density shows by LRD and $|N_i(x)|$ denotes the number of samples in the neighbourhood of x observation. To compute the rate of reachability distance for each sample in the dataset, Eq. (2) was introduced.

$$\tilde{dis}_i(x, o) = \max\left(dis_i(o), dis_i(x, o)\right), \tag{2}$$

It is noted that $dis_i(o)$ mentions the shortest distance among the neighbours of observation o. Therefore, the LRD of observation x is defined

as follows.

$$\mathrm{LRD}_i(x) = 1/\frac{\sum_{o\in N_i(x)} \tilde{dis}_i(x,o)}{|N_i(x)|} \tag{3}$$

The formula is calculated as the inverse of the average reachability distance between x and its k-nearest neighbours $o \in N_i(x)$. The term $dis_i(x,o)$ represents the reachability distance from x to neighbour o, which accounts for both the actual distance and the neighbourhood radius of o. This measure quantifies how densely x is located with respect to its local neighbourhood—higher LRD values indicate that x resides in a denser region.

The LOF algorithm is suited for detecting outliers in datasets, including different distributions concerning density because it uses a relative measure of the density at every point concerning its surrounding neighbours instead of a general threshold value [36]. LOF is also resistant to differing data scales and able to handle both clustered and nonuniform data.

3.2. Evolutionary optimisation algorithms

Evolutionary Optimisation Algorithms have proved highly effective in optimising the performance of hybrid learning strategies such as ensemble models. These algorithms, namely GA, DE, and (1+1)EA, are optimally employed in challenging optimisation problems where traditional gradient-based or exhaustive search strategies are not applicable [37]. Ensemble learning algorithms typically have several base learners and a number of control parameters, such as learning rates, tree depths, and voting weights, whose manual tuning is time-consuming and inefficient. Meta-heuristic algorithms solve this issue by intelligently searching and exploring the parameter space in order to avoid premature convergence, making a balance between global and local search [38]. Their ability to operate without derivative information and adapt to very nonlinear, high-dimensional objective landscapes renders them highly beneficial for hyperparameter optimisation by hand and the improvement of ensemble model accuracy, stability, and generalisation.

3.2.1. Differential Evolution (DE) algorithm

DE [39] is an evolutionary computational method, population-based, inspired by biological processes that use a stochastic search strategy to find the global optimum of a given problem. DE generates and maintains a population of candidate solutions, and each solution is designated as a vector of decision variables (binary, discrete or continuous values) in the optimisation problems. In order to evaluate the fitness of each solution, an objective function is introduced and based on this fitness, the solutions can be sorted. In the following generations, DE algorithm develops new vectors (offspring) by integrating and mutating individuals in the current population. The primary evolutionary operators of DE include crossover, mutation and selection.

Mutation operator:. In DE algorithms, the most significant operator is the mutation that stochastically perturbs a solution in the population to generate a new candidate solution [40]. The popular type of DE mutation is entitled "DE/rand/1/bin" (Eq. (4)). This strategy often intuitively supports a stronger exploration ability but almost shows a low convergence speed, promoting global exploration and reducing the risk of premature convergence. As a result, this strategy can usually be used to optimise problems with multi-modal attributes.

$$\vec{V}_g = \vec{D}_{r1} + \omega \times (\vec{D}_{r2} - \vec{D}_{r3}) \tag{4}$$

where $\vec{V}_g$ is the differential vector of three candidates ($\vec{D}_{r1}$, $\vec{D}_{r2}$, and $\vec{D}_{r3}$) chosen randomly from the current population, and to tune the exploration step size, ω is introduced as the mutation factor.

Crossover operator:. The binomial crossover strategy of DE enjoys several advantages that result in its effectiveness and wide use in real-world problems of continuous optimisation. Its computationally efficient and simple construction relies on random sampling and component-wise replacement, hence making it scalable to high-dimensional problems. Crossover rate (C_R) is a direct control parameter that facilitates flexible balancing between exploitation and exploration by regulating the fraction of the mutant vector in the trial solution. This promotes population diversity and avoids premature convergence. In addition, binomial crossover typically includes a mechanism to ensure that at least one component of the mutant vector is incorporated into the trial vector, thereby preventing cyclical solutions and enhancing the algorithm's ability to avoid local optima. Its generality in various problem spaces and robustness to diverse objective function topologies also speaks volumes about its effectiveness in solving complicated optimisation problems. The crossover operator combines the mutated solution with another one in the current population to form a trial solution. One of the well-known types of crossover is binomial [40], formulated based on Eq. (5).

$$\vec{S}^{i,j} = \begin{cases} \vec{V}^{i,j} & \text{if } (\mathrm{r} \le C_R) \text{ or } (j == C_n), \\ \vec{D}^{i,j} & \text{otherwise.} \end{cases} \quad j = 1, 2, \ldots N_D \tag{5}$$

where S and C_R are the trial vector and the rate of probability crossover defined in the range of [0–1], respectively. C_n is the index of solutions chosen in the crossover.

Selection strategy:. In Differential Evolution (DE), the selection strategy plays a crucial role in guiding the evolution process to optimal solutions. After a trial vector is created through mutation and crossover, DE applies a greedy selection strategy to determine whether the new solution should be retained. The new solution (S^i) is generated and combined with its parent (D^i) to replace the offspring as follows.

$$\vec{D}_i^{g+1} = \begin{cases} \vec{S}_i^g & \text{if } f\left(\vec{S}_i^g\right) \le f\left(\vec{D}_i^g\right), \\ \vec{D}_i^g & \text{otherwise.} \end{cases} \tag{6}$$

DE exhibits outstanding power in solving optimisation problems and has advantages such as simplicity, reliability, and robustness, and is particularly useful for solving complex optimisation problems where the objective function is non-linear, non-convex [41] and may have multiple local optima. However, DE has weaknesses, including slow convergence speed, difficulty adjusting parameters for different problems, and performance deterioration with increasing search space dimensionality.

3.2.2. Genetic Algorithms (GA)

GAs are population-based stochastic optimisation techniques that emulate the process of evolutionary biology to identify the best solutions [42]. GAs start with an array of feasible solutions; each expressed as a series of decision parameters. These candidate solutions are then subjected to selection, crossover, and mutation processes to generate new offspring solutions. Each resultant solution is then assessed by an objective function to determine its fitness level. Those with higher fitness are more likely to persist into subsequent generations, while those with lower fitness are phased out over time. The cycle repeats until a termination criterion is satisfied, such as reaching a predetermined number of cycles or finding an acceptable solution.

GAs achieve a delicate balance between the exploratory and exploitative aspects of optimisation [43]. Exploration involves surveying the search space to find new areas that might house superior solutions. Exploitation, on the other hand, is about improving the solutions located in promising regions. This equilibrium is realised through selection, crossover, and mutation. Selection favours the survival of fitter solutions. Crossover merges the genetic information of chosen solutions to create new offspring exhibiting a blend of characteristics. Mutation triggers random alterations in the offspring, fostering exploration by bringing unique genetic variations.

Crossover operator:. The geometric crossover technique [44] has been strategically chosen for its remarkable ability to identify and uncover potential solutions that lie precisely on the edge of what can be considered a feasible solution space, as referenced in the source. Moreover, this operation enables smooth transition in the search space, enhancing exploitation while preserving diversity. It is particularly beneficial for real-valued and continuous optimisation problems since it guarantees feasibility and enables convergence towards optimal regions with higher precision. Envision two parent chromosomes, represented mathematically as $A = \{a_1, a_2, \ldots, a_n\}$ and $B = \{b_1, b_2, \ldots, b_n\}$, from which the offspring are derived through a specific calculation method outlined below.

$$C = \left\{ \sqrt{a_1 \cdot b_1}, \sqrt{a_2 \cdot b_2}, \ldots, \sqrt{a_n \cdot b_n} \right\} \tag{7}$$

$$C_i = (A_i)^{\alpha} \cdot (B_i)^{1-\alpha}. \tag{8}$$

In this context, the variable i denotes the number of individual indexes associated with each chromosome. At the same time, α is confined to the interval [0,1], indicating the proportion that influences the merging of the parent chromosomes. Specifically when the value of α is set to $\frac{1}{2}$, thereby illustrating a balanced combination of both parent genes. Two offspring are created by swapping parent positions during the second calculation, adding variety to genetic mixing. This method also supports multiple parents, increasing genetic diversity and innovation as follows.

$$C_i = \left(A_i^1\right)^{\alpha_1} \left(A_i^2\right)^{\alpha_2} \left(A_i^3\right)^{\alpha_3} \ldots \left(A_i^n\right)^{\alpha_n}, \quad \text{where, } \sum_{i=1}^{n} \alpha_i = 1 \tag{9}$$

Mutation operator:. A crucial mechanism in the realm of genetic algorithms is a mutation, which plays a significant role in altering one or more genes within a given population, thereby enhancing the overall variability and diversity of that population in an effort to explore the vast landscape of potential solutions more thoroughly.

To illustrate this concept, let us consider an individual represented as $A_1 = (a_1, a_2, \ldots, a_n)$, where each variable in a solution a_i is confined within a specific range, defined by the lower bound $Low_b(i)$ and the upper bound $Up_b(i)$, which respectively set the limits for that variable's potential values.

A non-uniform mutation operator was used, which is designed to alter the selected variables in a manner that is not uniform across the population but rather varies depending on certain criteria. Eq. (10) shows the formulation of this mutation where *iter* and $iter_{max}$ are the current and maximum generation number, ϑ is a random number between 0 and 1, and β is a system parameter determining the degree of non-uniformity equal to 6 in this research.

$$a_i' = \begin{cases} a_i + (Up_b(i) - a_i)\left(\vartheta \cdot \left(1 - \frac{iter}{iter_{max}}\right)\right)^{\beta} & if\ rand \le \alpha \\ a_i - (a_i - Low_b(i))\left(\vartheta \cdot \left(1 - \frac{iter}{iter_{max}}\right)\right)^{\beta} & if\ rand > \alpha \end{cases} \tag{10}$$

Population size importance:. Population size that determines the number of solutions in it is a critical factor in determining the effectiveness of GAs. A large population promotes greater diversity and exploration, but it also results in higher computational expense. Small populations, conversely, might converge more quickly but have the potential to get stuck in suboptimal solutions. Problem complexity, search space, and computing resources can determine the optimal selection of a population size. It should measure the objectives, boundaries, and requirements specific to the problem and establish how close a solution is to global or local optimal. These fitness values are utilised by the GA to guide the search process, favouring solutions with greater fitness values. GAs can also be hybridised in a hybridisation with other optimisation methods in an attempt to enhance performance. Hybridisation strategies leverage the strengths of various algorithms while mitigating their weaknesses. For instance, genetic algorithms can be blended with local search techniques to enhance the performance of the genetic algorithm and convergence to improved solutions.

3.2.3. Single-parent evolutionary algorithm

The Single-Parent evolutionary algorithm known as 1+1EA is an optimisation method [45] that begins with a starting solution, X and generates a new solution, Y, in each iteration by randomly altering one or more selected variables in X ($X_{iter} \in \{LB, UB\}^N$), where UB and LB represent the upper and lower bounds of the variable, respectively, and N denotes the number of variables. Unlike the standard 1+1EA, which employs a uniform distribution for mutation, resulting in a local search that is both non-curved and noisy, we prefer to utilise a normally distributed transformation [46]. Next, the new solution generated is evaluated and compared with its parent. If the fitness of the new solution dominates the previous one, it will be replaced. Otherwise, the new solution will be removed, and another solution generates from the parent candidate.

Mutation operator:. Contrary to the default 1+1EA with uniform random mutation, leading to non-curved and noisy search behaviour, our implementation employs a Gaussian (normally distributed) mutation scheme to enable better local search in the vicinity of the parent solution. Specifically, the mutation for each decision variable i is defined as:

$$Y_i = \mathcal{N}\left(\mu = X_i, \sigma^2 = 0.2 \times (UB - LB)\right) \tag{11}$$

The parameter $\mu = X_i$ defines the mean of the distribution, ensuring that mutations occur locally around the current solution. The standard deviation σ is derived from the problem's variable bounds and is computed as $\sqrt{0.2 \times (UB - LB)}$. This adaptive normal distribution enables a more refined search around the parent solution, allowing for better exploitation of promising regions in the search space. The use of Gaussian noise is particularly effective in real-valued continuous optimisation problems where smooth convergence is desirable [46].

Selection strategy:. Once the new solution Y is generated, it undergoes fitness evaluation. The selection mechanism in 1+1EA follows a greedy strategy:

$$X^{(iter+1)} = \begin{cases} Y, & \text{if } f(Y) \le f\left(X^{(iter)}\right) \\ X^{(iter)}, & \text{otherwise} \end{cases} \tag{12}$$

1+1EA offers the advantage of changing only a small number of variables in each iteration. This characteristic allows for a gradual approach towards a nearly optimal solution. However, for large-scale optimisation problems, this can incur significant costs. Empirical evidence suggests that simpler EAs can occasionally outperform more complex methods. Additionally, 1+1EA proves to be a suitable choice when the fitness function involves a combinatorial optimisation problem [47].

3.3. Ensemble learning architectures

In machine learning, ensemble models combine several different models, resulting in much better overall predictions [48]. The procedure compensates for the weaknesses that may result from overfitting or bias. Now, this can be looked at broadly under three sections: first, bagging; second, boosting; and third, stacking. As the model trains on varied subsets to reduce variance, an exemplary model created with the procedure of '*bagging*' is a Random Forest [49]. Boosting, in a manner similar to that of Adaptive Boosting (AdaBoost) [50], XGBoost [51] and Gradient Boosting [52], tries to decrease the bias by iteratively correcting the model's past mistakes. Stacking takes an approach to meta-learning by making use of a high-order model to combine the predictions of lower-level base learners. Through their ability to aggregate diverse models, these ensemble methods have been successful in generalising and providing performance when single-model techniques fail. Among others, three significant advantages can be identified as more important than the benefits created by more traditional ML methods in this work: enhanced accuracy, robustness, and adaptability. This generally leads to better overall performances,

as ensemble methods aggregate several models to minimise both bias and variance in error [53]. It is much more robust towards noise and outliers among the data points. Above all, most ensemble methods can be seamlessly integrated with almost all data and problem types, whether classification or regression, making them natural selections for complex, practical applications—advantages that fully implement their valuable contribution to achieving state-of-the-art machine learning tasks.

3.3.1. Stacking ensemble models

Stacking ensemble models typically combines a set of base models — usually referred to as the level-0 learners — predictions via a higher-level meta-model, commonly referred to as the level-1 learner, for improving results [54]. Each of the base models uses different algorithms in their training with the same dataset with the aim of ensuring diversity that would utilise each of their unique strengths. It is crucial that the meta-model learns how to effectively combine the output from these base learners in a refined and generally more accurate final prediction [55]. The key insight to stacking is when different models specialise in combining strengths and other aspects of the problem. The advantages include increased predictive accuracy arising due to the combination of various models and immense versatility regarding the handling of complex challenges.

The framework of stacking can be described in the following steps [48]: The base models are trained using certain algorithms, the choice of which depends on the problem domain and requirements of the user. This step involves preparing the base learners using the provided training data. These, in turn, are used to develop a new dataset. This new dataset will contain the predicted outputs of the base models as new features and the actual target labels as corresponding target values. For example, any instance in the original dataset R if of the form $a_i, f(a)_i$, then the same instance in the new dataset created will be in the form $\hat{a}_i, f(a)_i$ where $\hat{a}_i$ is composed of the various outputs $h_1(a_i), h_2(a_2), \ldots, h_T(a_i)$ from the different base models. The meta-learner is then trained using this new dataset, hence learning how to integrate the predictions of the base models [48]. The meta-model is then deployed to combine the outputs from the base models for new, unseen data. In stacking, for an out-of-sample instance a, the ultimate prediction is a function from the meta-learner: $\hat{h}(h_1(a), h_2(a), \ldots, h_T(a))$, with respect to outputs from the base models—the level-0 models. However, despite its potential for high accuracy, stacking is not as widely adopted as either bagging or boosting due to the complexity of its implementation and the potential for data leakage if not handled appropriately.

3.3.2. Bagging ensemble models

Bagging, short for bootstrap aggregating, is an ensemble technique aimed at reducing the variance of model predictions and improving generalisation by combining multiple models [56]. These models are trained independently on diverse, randomly generated subsets of either the training data or input features. Each is trained separately on a different, random subset of the training data or input features. Bootstrapping refers to creating M sets of data $\{D_1, D_2, \ldots, D_M\}$ with size n, each drawn with replacement from the original training set D. Mathematically, for each dataset D_m, with $m = 1, 2, \ldots, M$, we have:

$$D_m = \{(x_i, y_i)\}_{i=1}^{n}, \quad D_m \sim D \tag{13}$$

Each subset D_m is used to train a base model $h_m(x)$. The final prediction is made by aggregating the outputs of these base models: For regression tasks, the prediction is given by the average:

$$\hat{y} = \frac{1}{M}\sum_{m=1}^{M} h_m(x) \tag{14}$$

The final prediction is made by aggregating the outputs of these models, using majority voting for classification tasks or averaging for regression tasks [57]. A prominent application of Bagging is the Random Forest algorithm, which builds numerous decision trees and combines their results to produce stable and accurate predictions.

Relative to stacking and boosting, Bagging possesses distinct advantages. Unlike boosting, which sequentially trains models with the emphasis being placed on rectifying errors from the previous iterations, Bagging trains its base models in parallel and independently from one another [58]. The parallel approach reduces the risk of overfitting and enhances computational efficiency. In addition, while stacking combines the heterogeneous algorithm predictions using a meta-learner, Bagging tends to employ a single algorithm type to create homogeneous models, which are simpler to implement. Another significant benefit of Bagging is that it is robust to noisy data and outliers because boosting does not assign extra weight to difficult instances. Bagging is particularly valuable in applications where variance reduction and generating consistent, generalised predictions are key goals.

3.3.3. Voting ensemble models

Voting is one of the most straightforward ensemble learning techniques, and the underlying principle is that combining predictions from multiple models yields overall improvements in performance. This approach works by aggregating base model outputs by majority vote or averaging [59]. Voting ensembles can be composed of homogeneous models (i.e., models of the same type trained on different data subsets) or heterogeneous models (i.e., models based on different algorithms). There are two main types of voting: majority voting for classification tasks and averaging for regression tasks. In an N_C class in a classification problem with N_e base classifiers, the output of the it h classifier for class c is denoted as $O_{i,c} \in \{0, 1\}$, where $O_{i,c} = 1$ if the classifier h_i predicts class c_r and $O_{i,c} = 0$ otherwise. With majority voting, the ensemble prediction ω_{c^*} is the class label that receives the most votes:

$$c^* = \arg \max_{c \in \{1,\ldots,N_C\}} \sum_{i=1}^{N_e} O_{i,c} \tag{15}$$

In weighted majority voting, every classifier h_i is assigned a weight w_i, which is its estimated reliability or generalisation ability. The class c^* is predicted by computing the weighted sum of votes across all classifiers:

$$c^* = \arg \max_{c \in \{1,\ldots,N_C\}} \sum_{i=1}^{N_e} w_i \cdot O_{i,c} \tag{16}$$

For regression, voting is replaced by averaging. Each base model produces a real-valued output $h_i(x)$, and the final prediction $\hat{y}$ is taken to be the average (or weighted average) of all base outputs:

$$\hat{y} = \frac{1}{N_e}\sum_{i=1}^{N_e} h_i(x) \quad \text{(unweighted)} \quad \text{or} \quad \hat{y} = \sum_{i=1}^{N_e} w_i \cdot h_i(x) \quad \text{(weighted)} \tag{17}$$

This ensemble process is simple yet effective, particularly when the base learners are heterogeneous because it tends to reduce variance while enhancing robustness.

3.3.4. Boosting ensemble models

Extreme gradient boosting XGBoost is an innovative machine-learning methodology that enhances tree-based models through an assembly of classification and regression trees (CART) [51]. This methodology is structured on a gradient-boosting framework, which enables simultaneous tree boosting. The tree assembly model merges numerous weak learners to forecast the output by applying an incremental training approach. The steps of this incremental training are as follows: initially, the full scope of input data is adjusted by the first learner, after which the residuals, which are used to rectify the deficiencies of a weak learner, are modified by a subsequent learner. This adjustment procedure is repeated multiple times until the termination condition is met. The final prediction of the model is then derived as the cumulative

prediction of all learners. The parallel procedures are autonomously executed during the training phase, thereby facilitating the efficient use of computational resources [60]. Moreover, in order to deal with over-fitting issues, an advanced regularised formulation is applied as follows:

$$L(\omega) = \sum_{i}^{N} d(y_i', y_i) + \sum_{k} \lambda(f_k) \tag{18}$$

$$\lambda(f) = \alpha T + 1/2\beta\|s\|^2 \tag{19}$$

where d plays the role of the loss function to calculate the difference between the predicted value and true value. λ is the regularisation function to penalise the ld complexity of the model. α is a threshold to extend the leaf node. The weight of the leaf and regularisation parameter are shown by s and β, and T is the number of tree leaves.

XGBoost offers several advantages contributing to its widespread adoption and success in various domains. It can be used in a wide range of data types, including numerical, categorical, and text data. Additionally, XGBoost allows customising loss functions, enabling users to specify their objective functions and tailor the model to distinct conditions. Another benefit of XGBoost is offering valuable information about the significance of features, qualifying the users to comprehend how different predictors contribute to the model's overall performance [61]. Assessing feature importance simplifies the identification of influential variables, facilitates feature selection, and enhances the understanding of the underlying data.

3.4. Proposed adaptive evolutionary ensemble learning model

This section outlines the technical aspects of the proposed neuro-evolutionary model for forecasting energy consumption in smart buildings. The methodology comprises six main steps: baseline model comparison, hyper-parameter selection, and optimisation of the chosen model.

- Initially, we selected 15 diverse ML models for evaluation, including four traditional algorithms: Support Vector Machine (SVM), Logistic Regression (LR), Bayesian Linear Regression (BR), and k-nearest Neighbours (KNN). Additionally, we incorporated three neural network architectures: Multi-Layer Perceptron (MLP), Dense Neural Network (DNN), and Convolutional Deep Neural Network (CDNN). To further enhance diversity, three tree-based models, RF [62], DT, and Extra Tree (ET) were included. Lastly, seven ensemble models were trained and assessed: XGBoost [51], AdaBoost [63], Gradient Boosting Regressor (GBR) [64], Histogram-Based Gradient Boosting Regressor (HGBR) [65], Categorical Boosting (CatBoost) [66], and Light Gradient Boosting Machine (LGBM). The specific configurations [52] used for training these models are detailed in Table S4.
- We developed a robust hybrid ensemble framework that incorporates three strategies: stacking, bagging, and voting, to enhance the learning capability of an individual model by improving its predictive accuracy. This effectively fuses the strengths of each method in leveraging their complementary mechanisms towards a more accurate and reliable predictive model.
- In the stacking ensemble model, the best-performing model among 15 candidates was selected as the initial base learner, with linear regression as the meta-learner. Additional base learners were identified using a greedy search approach, incrementally adding models that improved performance metrics such as accuracy or error reduction. At each step, the combination of base learners yielding the highest performance was retained, ensuring the inclusion of only the most effective models while avoiding redundancy. The same technique was applied to optimise the meta-learner, further enhancing the ensemble's predictive capability (See Fig. 5). The details of the stacking ensemble model procedure can be seen in Algorithm 1 (In Appendix).
- In the second proposed ensemble model, we began with six superior-performing ML methods embedded in a weighted majority vote framework. Then, the models went one by one into the removing process, and the performance of the resultant ensemble was re-evaluated in the absence of the model that was being removed. The process was reiterated to see if this improved the accuracy of the prediction result. Then, weights within the final resultant ensemble were also optimised using the Nelder–Mead local search. The result was an optimal voting model, which only contained two methods, XGB and LGBM, having equal weights (See Table 6 and Fig. 9).
- Leveraging the unique advantages of bagging ensemble models — such as reducing variance, preventing overfitting, and improving stability — we developed an adaptive bagging framework. This approach involved evaluating nine models trained and tested within the bagging framework. The best-performing model, Extra Trees, was then selected for further optimisation. To enhance its performance, we applied a fast and robust optimisation algorithm, 1+1 Evolutionary Algorithm (1+1EA), to fine-tune its hyper-parameters, ensuring optimal predictive accuracy and efficiency (See Table 5 and Fig. 8).
- Finally, we implemented and compared four widely used meta-heuristic algorithms — GA, DE, Particle Swarm Optimisation (PSO), and 1+1EA — to optimise the hyper-parameters of the proposed ensemble models, assessing their effectiveness and performance. Meta-heuristic algorithms explore and find the optimal and feasible combination of parameters to maximise the prediction accuracy of total power consumption using IoT-collected information. The formulation is represented as follows.

$$f^*(h) = argmax_{h\in\Psi} f(h)$$
$$\text{Subject to} \tag{20}$$
$$[h_i] \in \Lambda, \quad i = 1, \ldots, N_h$$

where Ψ and N_h are the search space and number of hyper-parameters listed in Table S3. $f(h)$ evaluates the machine learning effectiveness with the set of hyper-parameters h that should be maximised. The fitness function ($f(h)$) is subjected to the boundary constraints (Λ) listed in Table S4.

For the stacking ensembles' meta-learner, we selected the top ten models performing on cross-validation metrics (R-value, MAE, RMSE). This ensured that only those models with very high individual predictive ability were chosen for the second-level learning process. To construct the sub-learner block in stacking and voting ensembles, we employed a greedy forward selection strategy. This strategy begins with the top-performing model and gradually includes the subsequent candidates one by one, only retaining a model if its addition leads to a gain in average performance for all measures of evaluation. The procedure is iterated until no more models can further enhance the predictive performance of the ensemble. Using this method, we prepared and tested ten stacked scenarios, each being compared in terms of performance gains. Similarly, in the case of bagging ensembles, we created eight models with the ensemble of the top-performing individual learners under a single feature space. In the case of voting ensembles, a greedy selection strategy demonstrated that gains in performance plateaued after two base models at maximum, so we had six fine-tuned voting models. Such systematic selection also ensures that resultant ensemble structures are not only high-performing but also efficient in computation and non-redundant.

In order to ensure guarantees of convergence and stability of individual learners in the ensemble, our proposed framework (Fig. 5) contains several precautions designed to mitigate the impact of non-converging models on the overall process of training. Each candidate learner is first independently tested with K-fold cross-validation, thereby separating any instability or non-convergence associated with that specific model so that it does not contaminate the integrity of the

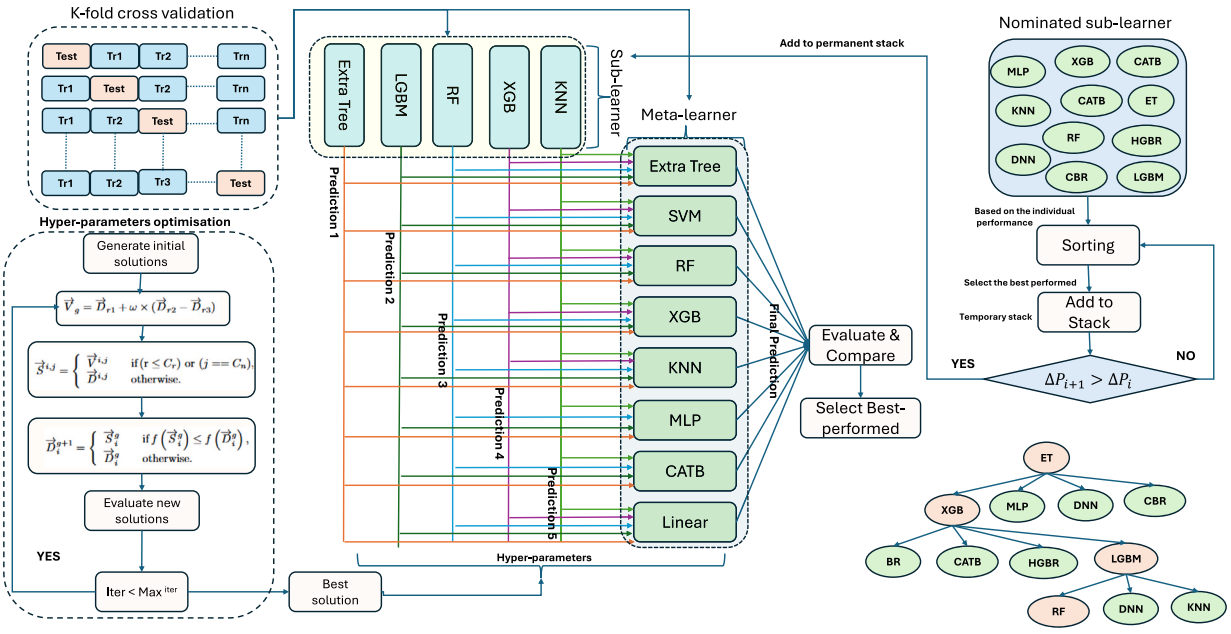

Fig. 5. Schematic flowchart illustrating the workflow of the proposed adaptive evolutionary stacking ensemble model, highlighting the ensemble tree model's performance as determined by the greedy search method.

ensemble. Suppose a learner fails to converge or has a score below some threshold. In that case, the greedy selection strategy, illustrated on the right of the schematic, removes it systematically from the stacking structure. This is based on the difference in performance (ΔP), and only those sub-learners that enhance the ensemble's overall predictive accuracy are retained in the transient and subsequently in the permanent stack. Moreover, the hyperparameter optimisation module (in the top centre of the figure) enhances the likelihood of convergence through the application of a metaheuristic search strategy to incrementally tune each learner's parameters adaptively. This serves the purpose of bypassing local optimum areas of parameter space, which else could induce training instability or divergence. Finally, the meta-learner is trained only after the sub-learner block has been completed from converged and validated models. Therefore, any non-converging learner is naturally excluded from the final ensemble, and the pipeline for training is stable, robust, and driven by validated performance improvement.

With the modelling framework and optimisation techniques established, the following section presents the experimental results. Here, we evaluate the predictive performance of the proposed hybrid ensemble models in comparison with baseline machine learning and deep learning algorithms using a range of statistical metrics.

4. Experimental results

This study presents the outcomes achieved through the utilisation of the proposed three hybrid evolutionary ensemble strategies and 15 popular ML models in predicting the total power consumption of appliances based on a hybrid dataset of meteorological parameters, energy use of appliances, temperature, humidity, and lighting energy consumption of different sections collected by 18 sensors in a building which is located in Stambruges, Mons in Belgium. Additionally, a concise analysis of the key findings from this research is presented. With regard to developing a comprehensive and robust comparative prediction framework, 14 effective ML models were selected. Each model was independently trained ten times based on 10-fold cross-validation, with the percentages of training, validation, and testing set at 80%, 10%, and 10%, respectively. We employed a parallelised K-fold cross-validation strategy to address computational demands associated with training advanced ensemble models through K-fold cross-validation. Because every fold in cross-validation is independent, model training and validation for every fold were executed in parallel on multiple CPU cores. This significantly reduced the overall runtime without sacrificing cross-validation's strengths in robustness and generalisability. Specifically, we utilised parallel computing abilities in Python's scikit-learn package (via n_jobs = −1) and tuned our model pipelines to enable parallel processing without compromising reproducibility.

4.1. Evaluation metrics

To assess the performance of the proposed hybrid models alongside the other 15 ML models, we utilised seven widely recognised evaluation metrics [67], as outlined in Table S2. Among these, MSE, RMSE, MAE, MSLE, and SMAPE are metrics where lower values indicate better performance. Conversely, higher values are more desirable for EVS and R-value, as they reflect greater predictive accuracy and a stronger linear relationship between predictions and true values. Where N_s represents the total number of samples, $f_e(k)$ denotes the estimated (predicted) output of the model for the k^{th} sample, and $f_t(k)$ is the corresponding true (target) value.

4.2. Quantitative evaluation and statistical analysis

This section provides a detailed quantitative comparison of the proposed models on the basis of statistical performance metrics. Certain error measures and R-values are used to compare the accuracy, robustness, and generalisation capacity of the models with various experimental configurations. Comparative statistical analysis with conventional methods is also included to reasonably validate the excellence of the proposed framework in predicting energy consumption in smart buildings.

4.2.1. Baseline models experimental results

Table 3 presents the statistical results corresponding to 14 ML models' performance to predict the power appliances' consumption using six evaluation metrics. The analysis of the provided Table 3 reveals intriguing findings regarding various models' prediction accuracy (R-value). Notably, the XGBoost model emerged as the top performer, exhibiting an impressive average accuracy of 73% across ten runs. We

Table 3
Statistical analysis results of the appliances power consumption prediction using 14 well-known machine learning methods, neural networks, deep learning, ensemble, tree-based and hybrid methods.

	SVM							MLP					
	RMSE	MAE	MSLE	SMAPE	EVS	R-value		RMSE	MAE	MSLE	SMAPE	EVS	R-value
Min	9.45E+01	4.11E+01	2.99E−01	3.28E+01	9.46E−02	3.72E−01	Min	8.19E+01	4.24E+01	2.43E−01	3.46E+01	2.79E−01	5.29E−01
Max	1.05E+02	4.47E+01	3.30E−01	3.49E+01	1.10E−01	4.15E−01	Max	8.83E+01	5.46E+01	4.60E−01	4.71E+01	3.59E−01	6.00E−01
Mean	1.01E+02	4.30E+01	3.14E−01	3.37E+01	1.01E−01	3.85E−01	Mean	8.55E+01	4.69E+01	3.26E−01	4.02E+01	3.20E−01	5.67E−01
Median	1.02E+02	4.32E+01	3.15E−01	3.36E+01	1.00E−01	3.81E−01	Median	8.56E+01	4.67E+01	3.05E−01	4.06E+01	3.19E−01	5.66E−01
STD	2.61E+00	1.09E+00	9.25E−03	5.24E−01	4.36E−03	1.16E−02	STD	2.16E+00	3.05E+00	6.57E−02	3.05E+00	2.15E−02	1.86E−02
	DNN							CDNN					
	RMSE	MAE	MSLE	SMAPE	EVS	R-value		RMSE	MAE	MSLE	SMAPE	EVS	R-value
Min	7.22E+01	3.72E+01	2.04E−01	3.10E+01	3.17E−01	5.82E−01	Min	7.38E+01	3.33E+01	1.77E−01	2.68E+01	3.56E−01	6.35E−01
Max	8.55E+01	4.37E+01	2.38E−01	3.47E+01	4.80E−01	7.13E−01	Max	8.29E+01	3.66E+01	1.97E−01	2.80E+01	4.53E−01	6.91E−01
Mean	8.17E+01	4.08E+01	2.23E−01	3.30E+01	3.89E−01	6.43E−01	Mean	7.80E+01	3.48E+01	1.87E−01	2.72E+01	4.01E−01	6.59E−01
Median	8.25E+01	4.15E+01	2.21E−01	3.34E+01	3.77E−01	6.38E−01	Median	7.77E+01	3.49E+01	1.87E−01	2.72E+01	3.96E−01	6.56E−01
STD	4.06E+00	1.99E+00	1.29E−02	1.32E+00	4.57E−02	3.58E−02	STD	2.71E+00	1.04E+00	7.20E−03	3.95E−01	2.95E−02	1.70E−02
	HGBR							DT					
	RMSE	MAE	MSLE	SMAPE	EVS	R-value		RMSE	MAE	MSLE	SMAPE	EVS	R-value
Min	7.11E+01	3.61E+01	1.81E−01	3.02E+01	4.26E−01	6.54E−01	Min	8.56E+01	3.56E+01	2.13E−01	2.60E+01	1.54E−01	5.85E−01
Max	7.93E+01	3.96E+01	2.05E−01	3.17E+01	5.12E−01	7.23E−01	Max	9.50E+01	3.94E+01	2.43E−01	2.81E+01	3.05E−01	6.46E−01
Mean	7.53E+01	3.76E+01	1.94E−01	3.10E+01	4.64E−01	6.84E−01	Mean	9.01E+01	3.77E+01	2.23E−01	2.70E+01	2.27E−01	6.17E−01
Median	7.57E+01	3.78E+01	1.95E−01	3.09E+01	4.61E−01	6.82E−01	Median	9.03E+01	3.77E+01	2.22E−01	2.70E+01	2.37E−01	6.17E−01
STD	2.31E+00	8.54E−01	6.40E−03	3.88E−01	2.18E−02	1.73E−02	STD	2.38E+00	9.20E−01	8.54E−03	4.24E−01	4.64E−02	2.01E−02
	EBM							XGB					
	RMSE	MAE	MSLE	SMAPE	EVS	R-value		RMSE	MAE	MSLE	SMAPE	EVS	R-value
Min	6.61E+01	3.16E+01	1.57E−01	2.55E+01	4.58E−01	6.87E−01	Min	6.76E+01	3.15E+01	1.48E−01	2.46E+01	4.84E−01	7.03E−01
Max	7.74E+01	3.54E+01	1.75E−01	2.70E+01	5.47E−01	7.41E−01	Max	7.52E+01	3.48E+01	1.67E−01	2.59E+01	5.58E−01	7.49E−01
Mean	7.15E+01	3.37E+01	1.64E−01	2.62E+01	5.06E−01	7.14E−01	Mean	7.09E+01	3.27E+01	1.55E−01	2.52E+01	5.28E−01	7.30E−01
Median	7.11E+01	3.36E+01	1.64E−01	2.62E+01	5.06E−01	7.13E−01	Median	7.09E+01	3.25E+01	1.54E−01	2.52E+01	5.27E−01	7.30E−01
STD	3.02E+00	9.74E−01	5.03E−03	4.00E−01	2.35E−02	1.53E−02	STD	1.93E+00	7.34E−01	5.23E−03	3.80E−01	1.97E−02	1.24E−02
	AdaB							CatB					
	RMSE	MAE	MSLE	SMAPE	EVS	R-value		RMSE	MAE	MSLE	SMAPE	EVS	R-value
Min	1.00E+02	6.62E+01	5.12E−01	5.16E+01	−2.83E−01	3.46E−01	Min	6.85E+01	3.43E+01	1.71E−01	2.92E+01	4.79E−01	6.94E−01
Max	1.82E+02	1.67E+02	1.68E+00	1.03E+02	1.02E−01	4.01E−01	Max	7.48E+01	3.64E+01	1.87E−01	3.06E+01	5.52E−01	7.45E−01
Mean	1.31E+02	1.02E+02	8.96E−01	6.88E+01	−7.98E−02	3.69E−01	Mean	7.10E+01	3.55E+01	1.75E−01	2.98E+01	5.21E−01	7.23E−01
Median	1.26E+02	9.56E+01	8.26E−01	6.62E+01	−7.00E−02	3.66E−01	Median	7.07E+01	3.56E+01	1.75E−01	2.98E+01	5.20E−01	7.24E−01
STD	2.04E+01	2.40E+01	2.76E−01	1.21E+01	9.37E−02	1.40E−02	STD	1.72E+00	6.65E−01	3.98E−03	3.08E−01	1.61E−02	1.17E−02
	BR							RF					
	RMSE	MAE	MSLE	SMAPE	EVS	R-value		RMSE	MAE	MSLE	SMAPE	EVS	R-value
Min	9.00E+01	5.20E+01	3.73E−01	4.57E+01	1.52E−01	3.90E−01	Min	6.83E+01	3.36E+01	1.64E−01	2.78E+01	4.67E−01	6.84E−01
Max	1.00E+02	5.48E+01	4.34E−01	4.80E+01	1.88E−01	4.35E−01	Max	7.61E+01	3.69E+01	1.82E−01	2.94E+01	5.26E−01	7.34E−01
Mean	9.46E+01	5.32E+01	3.90E−01	4.67E+01	1.66E−01	4.07E−01	Mean	7.20E+01	3.49E+01	1.74E−01	2.85E+01	4.96E−01	7.07E−01
Median	9.46E+01	5.30E+01	3.86E−01	4.66E+01	1.64E−01	4.04E−01	Median	7.18E+01	3.47E+01	1.75E−01	2.86E+01	4.97E−01	7.08E−01
STD	2.52E+00	7.13E−01	1.54E−02	5.37E−01	1.23E−02	1.54E−02	STD	2.42E+00	8.78E−01	5.55E−03	4.10E−01	1.99E−02	1.55E−02
	GBM							LGBM					
	RMSE	MAE	MSLE	SMAPE	EVS	R-value		RMSE	MAE	MSLE	SMAPE	EVS	R-value
Min	6.80E+01	3.28E+01	1.57E−01	2.72E+01	4.98E−01	7.06E−01	Min	7.71E+01	4.25E+01	2.60E−01	3.89E+01	3.13E−01	6.16E−01
Max	7.40E+01	3.55E+01	1.73E−01	2.85E+01	5.53E−01	7.44E−01	Max	8.68E+01	4.55E+01	2.81E−01	4.06E+01	3.59E−01	6.78E−01
Mean	7.08E+01	3.42E+01	1.66E−01	2.80E+01	5.28E−01	7.28E−01	Mean	8.33E+01	4.43E+01	2.72E−01	4.00E+01	3.35E−01	6.47E−01
Median	7.02E+01	3.42E+01	1.67E−01	2.80E+01	5.31E−01	7.30E−01	Median	8.40E+01	4.45E+01	2.72E−01	4.01E+01	3.37E−01	6.48E−01
STD	1.93E+00	9.19E−01	4.96E−03	3.74E−01	1.45E−02	1.07E−02	STD	2.81E+00	8.44E−01	4.72E−03	4.12E−01	1.07E−02	1.48E−02

can see that in the best-case scenario, this model achieved a remarkable accuracy of 75%. Furthermore, the GBM, CatBoost, and EBM models also demonstrated considerable accuracy levels, with respective values of 72.8%, 72.3%, and 71.4%. It is worth mentioning that, in general, the performance of neural networks and deep learning models, such as Dense (DNN) and convolutional (CDNN) deep models, fell slightly behind ensemble models in terms of average accuracy. However, it is noteworthy to investigate that the AdaBoost model proved to be an oddity to this trend. These findings shed light on the comparative performance of different models, providing valuable insights for future analysis and decision-making processes.

4.2.2. Ensemble learning models result

In this section, we present a detailed discussion, analysis, and comparison of the performance of the three proposed evolutionary ensemble models: stacking, bagging, and voting. Finally, we evaluate these strategies against one another and identify the most effective approach, providing recommendations based on the results.

Stacking ensemble models finding. We evaluated the performance of various stacking models by combining multiple ML models as base learners and integrating them with meta-learners, as detailed in Table 4. The highest average accuracy, 80.3%, was achieved with a combination of ExtraTree, LGBM, RF, and KNN as base learners, paired with meta-learners such as Linear Regression or MLP, both yielding similar results. On average, stacking models demonstrated approximately a 10% improvement in prediction accuracy compared to individual ML models. Regarding MAE, the stacking model comprising ExtraTree, LGBM, RF, and KNN with Linear Regression as the meta-learner outperformed XGB, LGBM, and RF, with improvements of 91.2%, 159.0%, and 102.3%, respectively.

Table 4
Statistical analysis results of the appliances power consumption prediction using 10 stacking ensemble methods.

Stacking (ExtraTree+LGBM+RF+KNN/Cat)							Stacking (ExtraTree+LGBM+RF+KNN/linear)						
Metric	RMSE	MAE	R_value	MSLE	EVS	SMAPE	Metric	RMSE	MAE	R_value	MSLE	EVS	SMAPE
Min	3.04E+01	1.62E+01	7.58E−01	6.96E−02	5.56E−01	1.88E+01	Min	2.70E+01	1.51E+01	7.53E−01	6.63E−02	5.67E−01	1.84E+01
Max	3.74E+01	1.91E+01	8.29E−01	9.86E−02	6.86E−01	2.08E+01	Max	3.75E+01	1.89E+01	8.45E−01	9.20E−02	7.10E−01	2.08E+01
Mean	3.45E+01	1.78E+01	7.91E−01	8.46E−02	6.22E−01	1.95E+01	Mean	3.29E+01	1.71E+01	**8.03E−01**	8.02E−02	6.44E−01	1.93E+01
Median	3.46E+01	1.78E+01	7.85E−01	8.46E−02	6.16E−01	1.95E+01	Median	3.36E+01	1.73E+01	8.06E−01	8.05E−02	6.47E−01	1.94E+01
STD	2.04E+00	8.05E−01	2.03E−02	6.98E−03	3.42E−02	5.45E−01	STD	2.62E+00	9.82E−01	2.10E−02	6.78E−03	3.33E−02	5.94E−01
Stacking (ExtraTree+LGBM+RF+KNN/MLP)							Stacking (ExtraTree+LGBM+RF+KNN+XGB/CBR)						
Metric	RMSE	MAE	R_value	MSLE	EVS	SMAPE	Metric	RMSE	MAE	R_value	MSLE	EVS	SMAPE
Min	3.05E+01	1.56E+01	7.35E−01	6.23E−02	5.39E−01	1.74E+01	Min	3.20E+01	1.69E+01	7.41E−01	7.71E−02	5.42E−01	1.85E+01
Max	3.82E+01	1.98E+01	8.40E−01	9.68E−02	6.98E−01	2.14E+01	Max	3.87E+01	1.99E+01	8.32E−01	1.01E−01	6.87E−01	2.11E+01
Mean	3.38E+01	1.75E+01	**8.03E−01**	8.24E−02	6.43E−01	1.95E+01	Mean	3.47E+01	1.81E+01	7.88E−01	8.62E−02	6.17E−01	1.97E+01
Median	3.33E+01	1.75E+01	8.08E−01	8.33E−02	6.49E−01	1.95E+01	Median	3.44E+01	1.78E+01	7.93E−01	8.49E−02	6.28E−01	1.96E+01
STD	2.05E+00	1.08E+00	2.56E−02	7.41E−03	3.98E−02	9.44E−01	STD	1.90E+00	8.33E−01	2.46E−02	5.86E−03	4.07E−02	6.10E−01
Stacking (ExtraTree+LGBM+RF+KNN+XGB/KNN)							Stacking (ExtraTree+LGBM+RF+KNN+XGB/linear)						
Metric	RMSE	MAE	R_value	MSLE	EVS	SMAPE	Metric	RMSE	MAE	R_value	MSLE	EVS	SMAPE
Min	3.61E+01	1.89E+01	6.68E−01	1.05E−01	3.86E−01	2.19E+01	Min	3.19E+01	1.69E+01	7.23E−01	7.80E−02	5.16E−01	1.89E+01
Max	4.32E+01	2.22E+01	7.56E−01	1.27E−01	5.48E−01	2.37E+01	Max	4.19E+01	2.17E+01	8.12E−01	1.11E−01	6.58E−01	2.21E+01
Mean	4.00E+01	2.11E+01	7.20E−01	1.16E−01	4.85E−01	2.29E+01	Mean	3.53E+01	1.88E+01	7.77E−01	8.99E−02	6.00E−01	2.04E+01
Median	3.98E+01	2.11E+01	7.26E−01	1.15E−01	4.94E−01	2.30E+01	Median	3.47E+01	1.85E+01	7.82E−01	8.82E−02	6.07E−01	2.03E+01
STD	1.84E+00	8.30E−01	2.38E−02	6.74E−03	4.41E−02	5.32E−01	STD	2.55E+00	1.15E+00	2.57E−02	8.58E−03	4.11E−02	7.80E−01
Stacking (ExtraTree+LGBM+RF+KNN+XGB/RF)							Stacking (ExtraTree+LGBM+RF+KNN+XGB/SVM)						
Metric	RMSE	MAE	R_value	MSLE	EVS	SMAPE	Metric	RMSE	MAE	R_value	MSLE	EVS	SMAPE
Min	3.17E+01	1.73E+01	7.40E−01	7.83E−02	5.28E−01	1.96E+01	Min	3.12E+01	1.59E+01	7.56E−01	7.18E−02	5.58E−01	1.80E+01
Max	3.83E+01	2.03E+01	8.14E−01	1.05E−01	6.60E−01	2.19E+01	Max	3.85E+01	1.92E+01	8.34E−01	9.02E−02	6.74E−01	2.03E+01
Mean	3.50E+01	1.88E+01	7.83E−01	9.23E−02	6.08E−01	2.06E+01	Mean	3.44E+01	1.72E+01	8.00E−01	7.97E−02	6.30E−01	1.90E+01
Median	3.48E+01	1.87E+01	7.81E−01	9.32E−02	6.07E−01	2.07E+01	Median	3.43E+01	1.71E+01	8.04E−01	7.98E−02	6.33E−01	1.90E+01
STD	1.85E+00	8.25E−01	1.97E−02	6.92E−03	3.45E−02	6.10E−01	STD	1.93E+00	8.29E−01	1.74E−02	5.79E−03	2.69E−02	6.61E−01
Stacking (ExtraTree+LGBM+RF+KNN+XGB/XGB)							Stacking (ExtraTree+LGBM+RF+KNN+XGB/ExtraTree)						
Metric	RMSE	MAE	R_value	MSLE	EVS	SMAPE	Metric	RMSE	MAE	R_value	MSLE	EVS	SMAPE
Min	3.02E+01	1.67E+01	7.15E−01	7.36E−02	5.05E−01	1.92E+01	Min	3.17E+01	1.71E+01	7.41E−01	7.95E−02	5.30E−01	1.96E+01
Max	4.16E+01	2.11E+01	8.44E−01	1.07E−01	7.10E−01	2.19E+01	Max	3.79E+01	2.07E+01	8.21E−01	1.10E−01	6.73E−01	2.26E+01
Mean	3.54E+01	1.88E+01	7.85E−01	8.99E−02	6.13E−01	2.05E+01	Mean	3.48E+01	1.87E+01	7.83E−01	9.12E−02	6.09E−01	2.07E+01
Median	3.54E+01	1.86E+01	7.80E−01	9.01E−02	6.07E−01	2.05E+01	Median	3.52E+01	1.86E+01	7.81E−01	9.07E−02	6.06E−01	2.06E+01
STD	2.43E+00	9.42E−01	2.79E−02	7.63E−03	4.57E−02	6.40E−01	STD	1.98E+00	1.00E+00	2.08E−02	7.84E−03	3.44E−02	7.71E−01

To evaluate the contribution of each component within the best-performing stacking model (ST3), a series of ablation experiments were conducted by incrementally excluding and including individual learners. The prediction accuracy and corresponding MAE for each configuration are illustrated in Figure S2. Initially, the stacking model was tested using only KNN as the base learner, achieving an average R-value of 0.76. When Random Forest (RF) was incorporated into the ensemble, the model's accuracy improved by 2.63%, indicating its significant complementary effect. Further enhancement was observed upon adding LightGBM (LGBM), resulting in an additional 2.56% increase in accuracy. Finally, the inclusion of ExtraTree yielded a substantial improvement of 5.00%, confirming its valuable contribution to the ensemble. These results collectively highlight the additive performance gains achieved through a carefully structured stacking approach.

Bagging ensemble models finding. In the second prediction scenario, we developed eight bagging ensemble models selected from 15 ML models based on their individual prediction accuracy. As summarised in Table 5, Bagging Extra-Trees outperformed all other bagging models, achieving an average accuracy of 82.1%, representing a 9% improvement over the standalone Extra-Tree base model. The high performance of Bagging Extra-Trees can be attributed to their randomised splitting mechanism, which enhances generalisation and reduces the risk of overfitting. In contrast, models like XGBoost, CatBoost, and GBR are more susceptible to overfitting, particularly on noisy or imbalanced datasets, unless carefully regularised.

To assess the effect of the number of estimators on the performance of bagging ensembles, we conducted a detailed experiment using Bagging with Extra Trees (Bag-ExtraTree, which performed best) and XGBoost (Bag-XGB) as base learners. Each model was evaluated across a range of ensemble sizes, varying the number of estimators from 1 to 30. As can be illustrated in Figure S1, increasing the number of estimators initially leads to improvements in both prediction accuracy (R-value) and MAE, indicating enhanced generalisation and reduced prediction error. However, this trend does not persist until 30. In the case of Bag-ExtraTree, performance gains plateau after 14 estimators, while Bag-XGB shows diminishing returns beyond 24 estimators. These observations highlight the importance of selecting an optimal ensemble size to avoid unnecessary computational complexity without compromising model accuracy.

Voting ensemble models finding. Table 6 presents the statistical prediction results of six voting ensemble models. Among these, the combination of Extra-Trees and LGBM in a bagging framework achieved the highest average accuracy of 80.6%. This superior performance can be attributed to the complementary strengths of the two algorithms, as their diversity and aggregation enhance overall predictive capabilities. The box-and-whisker plot in

4.3. Visual interpretation and model performance insights

This section presents a qualitative overview of the most significant experimental results, complementing the quantitative findings in the previous section. Using various plots, model behaviour comparisons, and performance visualisations, we aspire to provide deeper insight into the predictive ability and interpretability of the proposed ensemble learning models. The visualisations help identify temporal trends, model robustness, and relative performance of different configurations under actual real-world smart building conditions.

Table 5
Statistical analysis results of the appliances power consumption prediction using four proposed neuro-evolutionary methods.

Bag-XGB							Bag-CATB						
Metric	RMSE	MAE	R_value	MSLE	EVS	SMAPE	Metric	RMSE	MAE	R_value	MSLE	EVS	SMAPE
Min	2.96E+01	1.57E+01	7.42E−01	6.29E−02	5.47E−01	1.78E+01	Min	3.35E+01	1.88E+01	7.25E−01	9.40E−02	5.20E−01	2.18E+01
Max	3.85E+01	1.96E+01	8.66E−01	9.12E−02	7.39E−01	1.99E+01	Max	4.16E+01	2.25E+01	7.95E−01	1.20E−01	6.21E−01	2.43E+01
Mean	3.31E+01	1.71E+01	8.09E−01	7.73E−02	6.51E−01	1.90E+01	Mean	3.80E+01	2.08E+01	7.53E−01	1.08E−01	5.56E−01	2.32E+01
Median	3.24E+01	1.69E+01	8.13E−01	7.70E−02	6.56E−01	1.90E+01	Median	3.82E+01	2.09E+01	7.49E−01	1.08E−01	5.46E−01	2.32E+01
STD	2.68E+00	1.05E+00	2.87E−02	7.07E−03	4.43E−02	6.03E−01	STD	2.23E+00	9.13E−01	2.34E−02	6.09E−03	3.26E−02	6.44E−01
Bag-DT							Bag-ExtraTree						
Metric	RMSE	MAE	R_value	MSLE	EVS	SMAPE	Metric	RMSE	MAE	R_value	MSLE	EVS	SMAPE
Min	2.85E+01	1.56E+01	7.69E−01	6.50E−02	5.90E−01	1.76E+01	Min	2.96E+01	1.62E+01	7.85E−01	6.70E−02	6.14E−01	1.81E+01
Max	3.72E+01	1.92E+01	8.48E−01	9.21E−02	7.17E−01	2.06E+01	Max	3.72E+01	1.81E+01	8.56E−01	8.71E−02	7.30E−01	1.96E+01
Mean	3.26E+01	1.69E+01	8.09E−01	7.69E−02	6.52E−01	1.89E+01	Mean	3.28E+01	1.70E+01	**8.21E−01**	7.66E−02	6.72E−01	1.88E+01
Median	3.29E+01	1.70E+01	8.11E−01	7.70E−02	6.56E−01	1.91E+01	Median	3.28E+01	1.71E+01	8.22E−01	7.71E−02	6.75E−01	1.88E+01
STD	2.39E+00	9.73E−01	2.39E−02	6.82E−03	3.77E−02	6.87E−01	STD	2.04E+00	5.64E−01	1.95E−02	5.40E−03	3.16E−02	4.58E−01
Bag-GBR							Bag-LGBM						
Metric	RMSE	MAE	R_value	MSLE	EVS	SMAPE	Metric	RMSE	MAE	R_value	MSLE	EVS	SMAPE
Min	3.86E+01	2.30E+01	5.96E−01	1.24E−01	3.50E−01	2.58E+01	Min	3.23E+01	1.91E+01	6.85E−01	9.71E−02	4.67E−01	2.21E+01
Max	4.73E+01	2.66E+01	7.19E−01	1.59E−01	4.94E−01	2.85E+01	Max	4.43E+01	2.38E+01	7.85E−01	1.30E−01	6.08E−01	2.55E+01
Mean	4.27E+01	2.46E+01	6.61E−01	1.43E−01	4.27E−01	2.73E+01	Mean	3.83E+01	2.14E+01	7.36E−01	1.14E−01	5.35E−01	2.38E+01
Median	4.31E+01	2.45E+01	6.60E−01	1.43E−01	4.27E−01	2.73E+01	Median	3.83E+01	2.14E+01	7.43E−01	1.14E−01	5.41E−01	2.39E+01
STD	2.60E+00	1.00E+00	2.45E−02	9.92E−03	2.88E−02	6.90E−01	STD	2.93E+00	1.25E+00	2.55E−02	8.43E−03	3.54E−02	7.63E−01
Bag-RF							Bag-KNN						
Metric	RMSE	MAE	R_value	MSLE	EVS	SMAPE	Metric	RMSE	MAE	R_value	MSLE	EVS	SMAPE
Min	3.01E+01	1.65E+01	7.67E−01	6.98E−02	5.80E−01	1.88E+01	Min	2.98E+01	1.55E+01	7.90E−01	7.01E−02	6.21E−01	1.76E+01
Max	3.69E+01	1.91E+01	8.33E−01	9.61E−02	6.71E−01	2.13E+01	Max	3.74E+01	1.79E+01	8.35E−01	8.62E−02	6.89E−01	1.91E+01
Mean	3.36E+01	1.79E+01	7.99E−01	8.38E−02	6.29E−01	2.01E+01	Mean	3.26E+01	1.66E+01	8.08E−01	7.58E−02	6.52E−01	1.85E+01
Median	3.39E+01	1.80E+01	8.01E−01	8.50E−02	6.30E−01	2.02E+01	Median	3.26E+01	1.65E+01	8.08E−01	7.48E−02	6.51E−01	1.85E+01
STD	1.94E+00	7.82E−01	1.80E−02	7.32E−03	2.57E−02	6.90E−01	STD	1.99E+00	6.66E−01	1.32E−02	4.79E−03	2.02E−02	4.72E−01

Table 6
Statistical analysis results of the appliances power consumption prediction using six proposed voting ensemble methods.

Voting (XGB+LGBM)							Voting (XGB+CATB)						
Metric	RMSE	MAE	R_value	MSLE	EVS	SMAPE	Metric	RMSE	MAE	R_value	MSLE	EVS	SMAPE
Min	3.23E+01	1.73E+01	7.69E−01	7.61E−02	5.87E−01	1.92E+01	Min	2.86E+01	1.69E+01	7.04E−01	7.74E−02	4.94E−01	1.93E+01
Max	3.68E+01	1.97E+01	8.42E−01	9.46E−02	7.06E−01	2.10E+01	Max	3.84E+01	2.04E+01	8.24E−01	1.02E−01	6.74E−01	2.18E+01
Mean	3.42E+01	1.82E+01	7.92E−01	8.41E−02	6.27E−01	2.00E+01	Mean	3.47E+01	1.85E+01	7.85E−01	8.79E−02	6.14E−01	2.05E+01
Median	3.40E+01	1.81E+01	7.89E−01	8.49E−02	6.22E−01	1.99E+01	Median	3.56E+01	1.83E+01	7.87E−01	8.67E−02	6.17E−01	2.04E+01
STD	1.43E+00	6.35E−01	2.05E−02	4.70E−03	3.24E−02	4.71E−01	STD	2.64E+00	1.14E+00	2.60E−02	8.02E−03	3.94E−02	6.92E−01
Voting (XGB+KNN)							Voting (ExtraTree+LGBM)						
Metric	RMSE	MAE	R_value	MSLE	EVS	SMAPE	Metric	RMSE	MAE	R_value	MSLE	EVS	SMAPE
Min	3.02E+01	1.60E+01	7.38E−01	7.17E−02	5.36E−01	1.84E+01	Min	2.91E+01	1.61E+01	7.65E−01	7.25E−02	5.85E−01	1.85E+01
Max	3.85E+01	1.94E+01	8.40E−01	9.86E−02	7.01E−01	2.10E+01	Max	3.67E+01	1.91E+01	8.48E−01	1.02E−01	7.17E−01	2.11E+01
Mean	3.43E+01	1.75E+01	7.98E−01	8.28E−02	6.35E−01	1.93E+01	Mean	3.34E+01	1.75E+01	**8.06E−01**	8.32E−02	6.48E−01	1.96E+01
Median	3.40E+01	1.73E+01	7.95E−01	8.27E−02	6.32E−01	1.91E+01	Median	3.31E+01	1.74E+01	8.05E−01	8.08E−02	6.43E−01	1.94E+01
STD	2.19E+00	8.50E−01	2.43E−02	7.51E−03	3.93E−02	6.37E−01	STD	2.10E+00	9.03E−01	2.15E−02	8.19E−03	3.44E−02	7.93E−01
Voting (ExtraTree+CATB)							Voting (ExtraTree+KNN)						
Metric	RMSE	MAE	R_value	MSLE	EVS	SMAPE	Metric	RMSE	MAE	R_value	MSLE	EVS	SMAPE
Min	2.99E+01	1.64E+01	7.63E−01	7.01E−02	5.79E−01	1.86E+01	Min	2.87E+01	1.54E+01	7.51E−01	6.66E−02	5.58E−01	1.79E+01
Max	3.72E+01	1.94E+01	8.23E−01	9.29E−02	6.76E−01	2.09E+01	Max	3.84E+01	1.93E+01	8.48E−01	9.69E−02	7.19E−01	2.03E+01
Mean	3.39E+01	1.80E+01	7.98E−01	8.35E−02	6.35E−01	1.99E+01	Mean	3.40E+01	1.74E+01	7.98E−01	8.27E−02	6.34E−01	1.90E+01
Median	3.42E+01	1.82E+01	8.03E−01	8.38E−02	6.41E−01	2.00E+01	Median	3.45E+01	1.77E+01	7.96E−01	8.31E−02	6.33E−01	1.89E+01
STD	1.98E+00	7.98E−01	1.83E−02	6.06E−03	2.89E−02	5.58E−01	STD	2.89E+00	1.21E+00	2.31E−02	9.19E−03	3.82E−02	7.82E−01

4.3.1. Benchmark models results

The box-and-whisker plot 6 presented in this analysis offers a comprehensive evaluation of 14 machine and deep learning techniques utilised for predicting household appliance energy consumption. This evaluation focuses on prediction accuracy and MAE. In plot 6, a box is drawn between the first and third quartiles, with a vertical line passing through the box at the median. The whiskers extend from each quartile to the minimum and maximum values. Additionally, any outliers in the dataset are represented by single red crosses on the diagram. Upon analysing the plot, it becomes evident that XGBoost consistently outperforms other models in terms of the median accuracy metric. Following XGBoost, the GBM and Catboost models exhibit comparable performances. Furthermore, an intriguing observation can be made regarding the effectiveness of adding a convolutional layer to a dense model. This enhancement significantly improves the average performance of the model, indicating its potential for achieving higher accuracy. These findings provide valuable insights into the comparative performance of different machine and deep learning techniques in predicting household appliance energy consumption. This information can aid researchers and practitioners in selecting the most suitable models for their specific purposes, thereby enhancing the accuracy of energy consumption predictions. Furthermore, Fig. 6(b) presents the average absolute validation error for a set of 14 ML models. In terms of MSE, XGBoost stands out as the top performer with an impressive score of 32. Notably, the EBM model showcases a competitive performance in MAE and secures the second rank. Moreover, the GBM, Random Forest,

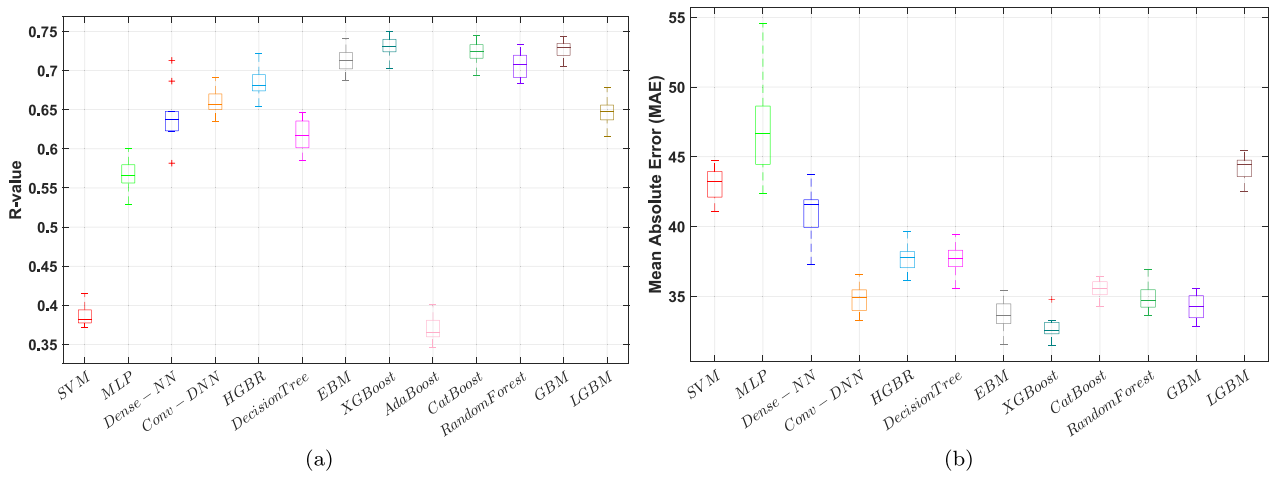

Fig. 6. The box-and-whisker plot of statistical results evaluation for 14 machine and deep learning techniques used for predicting the energy consumption of household appliances, based on (a) prediction accuracy (R-value) and (b) mean absolute error (MAE).

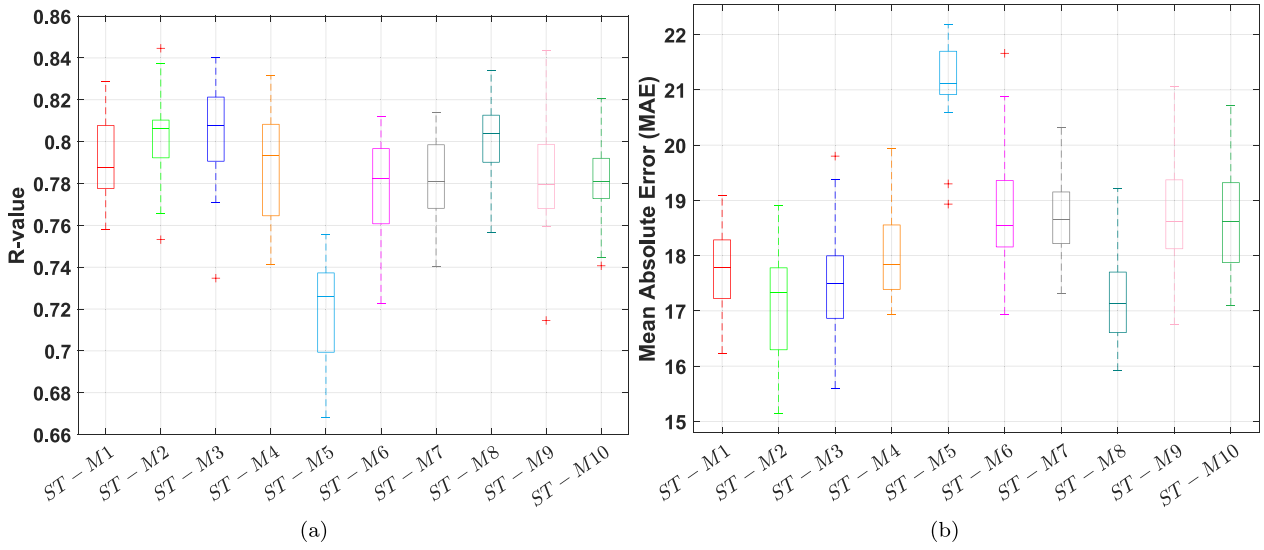

Fig. 7. The box-and-whisker plot of (a) R-value and (b) MAE statistical results for the ten stacking ensemble method in predicting the energy consumption of appliances in the smart house.

and CDNN models also demonstrate noteworthy performances, yielding acceptable results in their respective evaluations. This information provides valuable insights and highlights the strengths of XGBoost in achieving low MSE while acknowledging the competitive performance of the EBM model in MAE. The notable performances of GBM, Random Forest, and CDNN further contribute to the range of acceptable results obtained. These findings assist in understanding the efficacy of different ML models and offer guidance for selecting the most suitable approach based on the desired evaluation metric.

4.3.2. Ensemble models findings

Fig. 7 presents the statistical performance of 10 stacking models (listed in Table S5) evaluated in terms of R-value and MAE. Among these, the best-performing model in terms of median R-value accuracy is ST-M3 (ExtraTree+LGBM+RF+KNN/MLP), achieving an accuracy of 81%. Conversely, the model ST-M8 (ExtraTree+LGBM+RF+KNN+XGB/SVM) exhibits the lowest median MAE, accurately predicting appliance power consumption with a value of approximately 17.1.

Fig. 8 provides a detailed comparison of the eight bagging models in terms of R-value and MAE. While Bagging KNN demonstrated the lowest average MAE among all models, its overall accuracy was lower than that of Bagging Extra-Trees, Decision Trees, and XGBoost. This highlights a trade-off between minimising error and maximising accuracy, with Bagging Extra-Trees striking the best balance among the evaluated models.

Fig. 9 illustrates the performance of these models in terms of R-value and MAE. While Voting(XGB+LGBM) achieved the best median R-value, Voting(Extra-Tree+KNN) outperformed other models with the lowest average MAE, demonstrating its effectiveness in minimising prediction error.

4.3.3. Final comparisons

To ensure a fair comparison among the ensemble models proposed in this study, the results are presented in Fig. 10. As observed, the Bagging Extra-Trees model significantly outperformed the other ensemble methods, with a p-value less than 0.05 for both accuracy and MAE, indicating its superior predictive performance. Bagging ensembles are particularly effective in scenarios requiring variance reduction, noise handling, and robust generalisation across diverse datasets. These characteristics make bagging an ideal choice for predicting appliance power consumption, outperforming voting and stacking models in this context.

The results of the experiment demonstrate the superiority of the ExtraTree Bagging ensemble model over the Stacking (ST-M2, ST-M3,

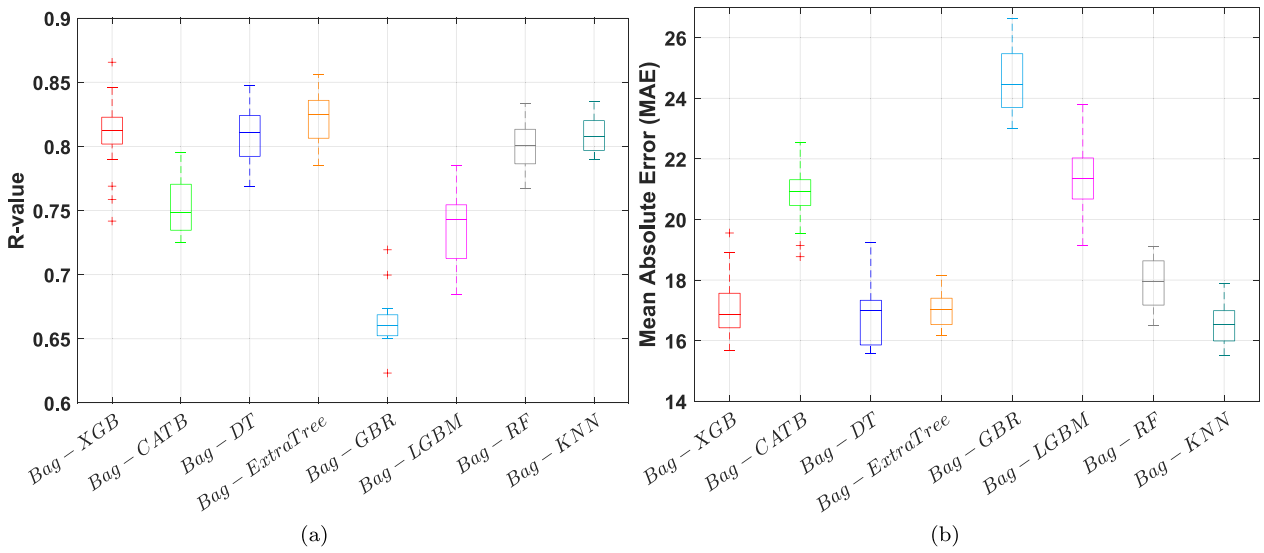

Fig. 8. The box-and-whisker plot of (a) R-value and (b) MAE statistical results for eight bagging ensemble method in predicting the energy consumption of appliances in the smart house.

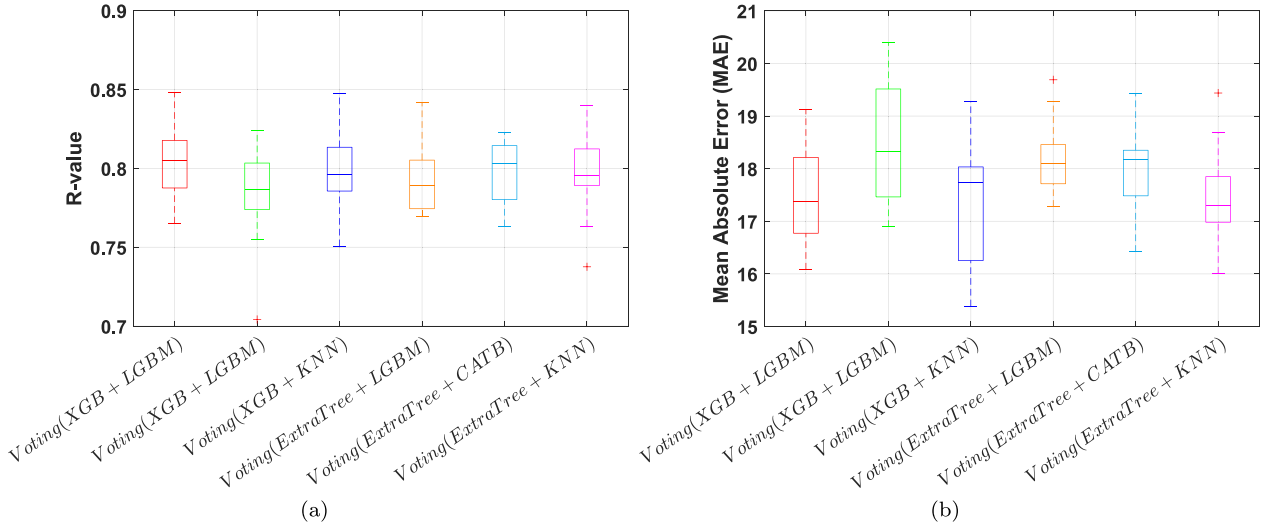

Fig. 9. The box-and-whisker plot of (a) R-value and (b) MAE statistical results for six voting ensemble methods in predicting the energy consumption of appliances in the smart house.

and ST-M8) and Voting ensemble methods. Specifically, the Bagging model achieved the best prediction accuracy rates for all the performance metrics. This is because the nature of Bagging reduces variance by combining the predictions of several decorrelated ExtraTree base models trained on different bootstrap samples. The ExtraTrees' randomness encourages model diversity and generalisation, thus, more stable and precise predictions. The Stacking model, however, relies on a meta-model to combine base models, which can sometimes introduce additional bias and be susceptible to overfitting if not carefully tuned. The Voting ensemble, similarly, treats all base learners equally without dynamically leveraging their individual strengths. These findings confirm that Bagging architecture, coupled with ExtraTrees provides a more robust and stable solution for energy consumption prediction in smart buildings.

4.3.4. Comparison with other techniques

To ensure a comprehensive comparison with previous studies using similar datasets, we evaluated 19 machine-learning models adopted from the works of Candanedo et al. [32], and Han et al. [68]. These models include Affinity Propagation Radial Basis Function (AP-RBF), standard Radial Basis Function (RBF) networks, and Backpropagation (BP) neural networks [68], each tested under varying configurations of hidden nodes to enable a robust and consistent performance assessment.

Fig. 11 provides a comparative assessment of RMSE scores of a variety of prediction models applied to the same dataset. Among all the models, our proposed model, Voting (XGB+KNN) ensemble produced the lowest RMSE, indicating superior predictive accuracy. It was closely followed by Bag-ExtraTree and ST-M3, both of which also performed well with significantly lower error rates than their standard base models. On the other hand, models such as AP-BP [68] and AP-ELM [68] possessed the highest RMSE values, which signifies poor generalisation ability and fitness to the target data. Ensemble methods, such as XGB, CatB, and HGBR, performed better than individual models, including SVM-Radial [32], GBM [32], and RF [32], consistently, reinforcing the advantage of ensemble creation in increasing predictability and robustness. Besides, neural-based architectures like MLP, DNN, and BP of reasonable sizes acted competitively but were sensitive to network size and training dynamics. Overall, the results show that ensemble and hybrid strategies are highly effective in controlling prediction errors in this application field.

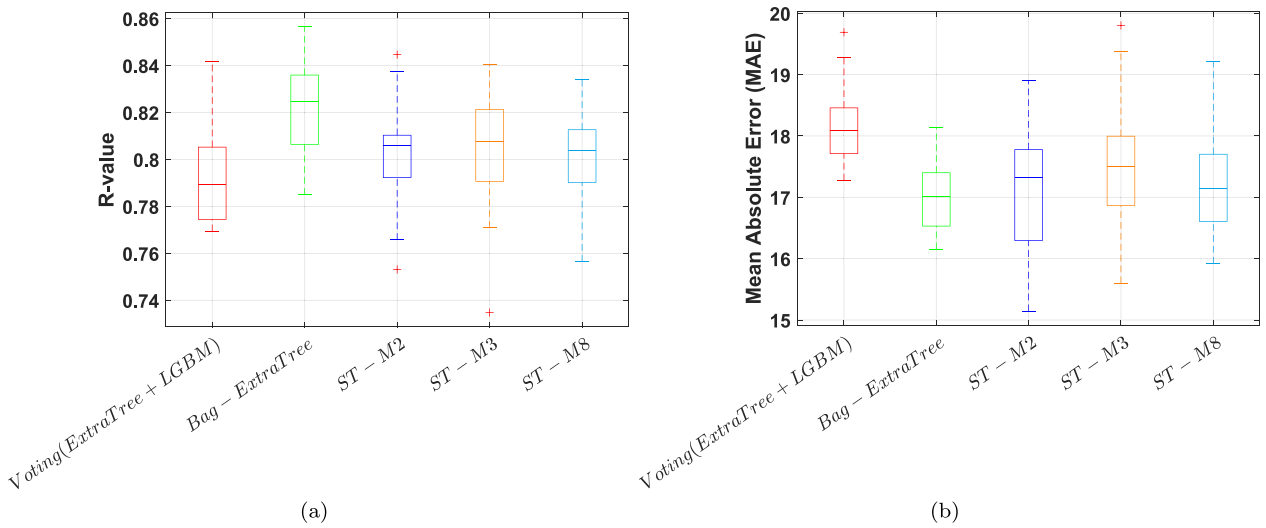

Fig. 10. The box-and-whisker plot of (a) R-value and (b) MAE statistical results for best-performed voting, bagging and stacking ensemble methods in predicting the energy consumption of appliances in the smart house.

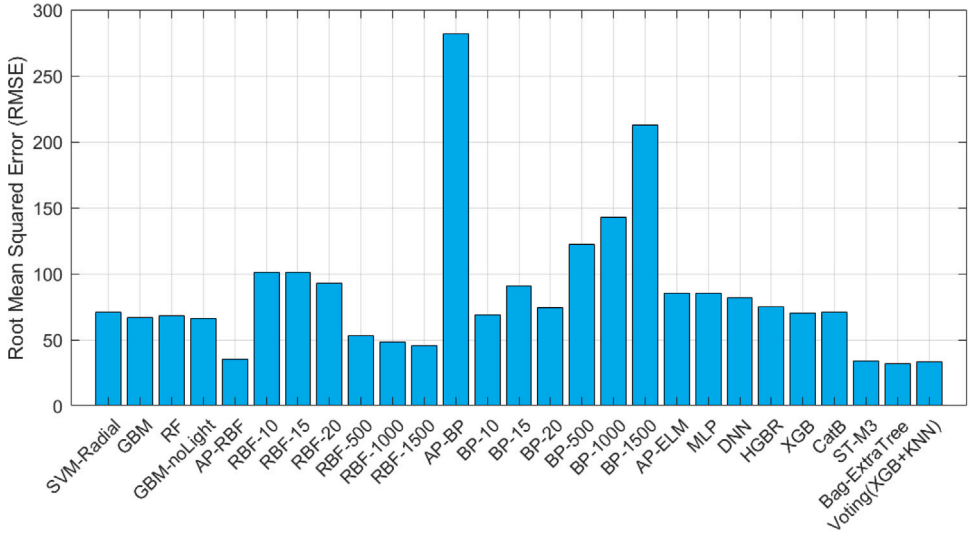

Fig. 11. Comparative analysis of energy consumption forecasting between the proposed models and prior studies.

4.4. Hyper-parameters optimisation

To evaluate the impact of hyper-parameters on model performance, we conducted an analysis using a greedy search, focusing on four key hyper-parameters: the number of estimators for Extra-Trees and Bagging, along with the maximum rate of features and samples used during training. The optimisation landscape for the number of estimators in Extra-Trees and Bagging is depicted in Fig. 12(a). For the Bagging ensemble, the number of estimators was evaluated in the range of 5 to 50, while for the Extra Trees model, the range of 10 to 100 was tested. The highest prediction accuracy was achieved with Bagging at Ns = 15 and with Extra Trees when the number of estimators exceeded 60. The results indicate that the number of estimators in the Bagging model has a more substantial influence on achieving higher accuracy compared to the number of estimators in the Extra-Trees model. Fig. 12(b) illustrates the prediction accuracy across different configurations of maximum sample rate and feature rate. The results indicate that the highest accuracy is obtained when both parameters exceed a threshold of 0.6, suggesting that retaining a larger proportion of samples and features enhances model performance. This highlights the critical role of properly tuning the Bagging model's hyper-parameters for improved predictive performance.

In this study, we employed four effective and well-known optimisation methods to adjust the hyper-parameters of ensemble models. In the first step, we focused on XGBoost hyper-parameters optimisation, and they are listed in Table S3. Fig. 13 illustrates a comparison of the average convergence speeds exhibited by these optimisation methods. It is important to note that the population size and maximum evaluation number are consistent across all methods at 25 and 1000, respectively. Upon analysis of Fig. 13, it is evident that XGB-EA demonstrates rapid convergence towards a semi-optimal configuration of hyper-parameters within the initial 20% of the total evaluation count. However, XGB-EA encounters challenges when confronted with a local optimum, and the mutation strategy employed does not effectively facilitate the exploration of alternative feasible regions. Conversely, although XGB-DE initially displayed a convergence rate lower than that of XGB-PSO and XGB-GA during the exploration phase, it ultimately managed to discover superior solutions. Considering the computational expense and time consumption associated with training the model, we recommend employing the 1+1EA meta-heuristic as a hyper-parameter optimiser.

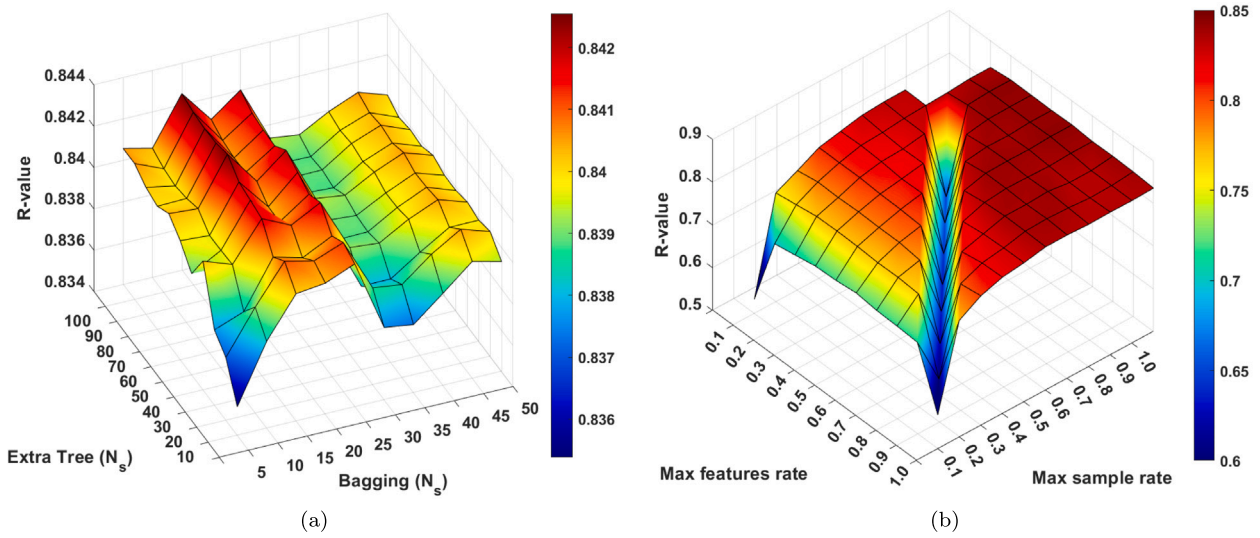

Fig. 12. Hyper-parameters tuning using grid search for bagging Extra Tree ensemble model.

Moreover, Fig. 13(b) and (c) shows the statistical performance analysis of the XGBoost with predefined hyper-parameters and four proposed neuro-evolutionary methods in terms of accuracy and MAE. It is crystal clear that the best-performing hybrid model is XGB-DE in terms of metrics, accuracy, and MAE. The accuracy and MAE improvement of XGB-DE are 3.5% and 7.6% compared with XGBoost.

Table S6 reports more technical comparison results of four evolutionary ensemble models. We can see that the XGB-DE prediction results had the minimum distance with the true power consumption values confirmed by metrics RMSE, MAE, and MSLE. In terms of the correlation coefficient (R-value), all hybrid models performed competitively; however, XGB-DE outperformed the other models. Finally, we evaluated the performance of four optimisation methods to enhance the Bagging Extra-Trees, best-performed model, as shown in Fig. 14. Among the tested methods, 1+1EA (Bag-ET-EA) demonstrated the fastest convergence during the initial iterations, highlighting its efficiency in optimisation. This experiment confirms that 1+1EA is an effective optimiser for fine-tuning hyper-parameters. Additionally, the balance between exploration and exploitation for the four hyper-parameters is illustrated in Fig. 14(b–e), providing further insights into the optimisation dynamics of each method.

As can be seen from Fig. 14(b), the optimisation process commenced by exploring a wide range of values for the number of Bagging estimators, ranging from 10 to 90. Throughout successive iterations, the search space became increasingly narrow, echoing the transition from exploration to exploitation, and ultimately converged within an optimum range of 60 to 65. An identical convergence pattern could be observed for the maximum feature rate hyperparameter, plotted in Fig. 14(c), where the search process converged around the value of 0.4. At the highest sample rate Fig. 14(d), the optimiser found good performing regions early in the search and converged rapidly to values above 0.9, finally settling at 1. Moreover, the number of estimations was subjected to an extensive and dense search over a larger space, with over 200 evaluations. Despite the wide initial range, the optimiser focused on configurations from above 60 estimators onwards and eventually settled at 80. Results like these bear testament to the optimiser's fair balance of local refinement and global search, terminating at well-chosen hyper-parameters for better model performance.

The quantitative results from Section 4 provide a solid foundation for interpreting the practical implications of the proposed approach. In this section, we delve into a critical discussion of the findings, highlighting performance trends, methodological strengths, and potential limitations based on the observed results.

5. Discussions and future directions

The proposed hybrid evolutionary ensemble models offer principal advantages in predicting total power consumption in smart buildings by effectively harnessing the merits of diverse learning algorithms and strong evolutionary optimisation. By integrating ensemble techniques such as Bagging, Stacking, and Voting with adaptive metaheuristic-based hyper-parameter tuning, the models achieve better accuracy, stability, and generalisations on highly dynamic and nonlinear energy consumption patterns. The hybrid approach enables the model to capture sophisticated dependencies between weather conditions, occupancy patterns, appliance usage, and ambient factors, typically neglected by separate algorithms. Moreover, the evolutionary optimisation process intelligence searches the hyper-parameter space, free from hand-tuning, and circumvents possible overfitting.

5.1. Scalability and dynamic pattern

The adaptive ensemble evolution learning method demonstrated in the proposal holds high scalability potential for use across various smart building environments with varying occupancy behaviour and energy use patterns. This is due to the modularity of the model, where multiple base learners (ExtraTrees, XGBoost, LGBM) are blended across ensemble frameworks (Bagging, Stacking, and Voting) and leverage evolutionary algorithms to drive optimisation of hyper-parameters. The combination of diverse learning paradigms enables the model to learn linear and nonlinear energy consumption patterns, and the evolutionary optimisation adjusts hyper-parameters according to different building-specific data distributions. These capabilities put the model in a position to generalise well beyond the current test case, particularly when retrained on new data from buildings with different spatial configurations, climate regions, or operating schedules.

Furthermore, the hybrid dataset used in this research, which includes indoor and outdoor temperature, humidity, lighting, occupancy, and appliance-level usage, represents a realistic and comprehensive sensing environment that is becoming increasingly common in modern smart buildings. The evolutionary tuning process also enables the model to adapt dynamically to changes in input feature importance, such as peak-hour demand or seasonal trends, which makes it more robust across various environments. Therefore, the model proposed is not limited to the Belgian building used for evaluation but can also be generalised to other types of buildings, such as commercial offices, schools, or housing estates. Follow-up work will focus on testing

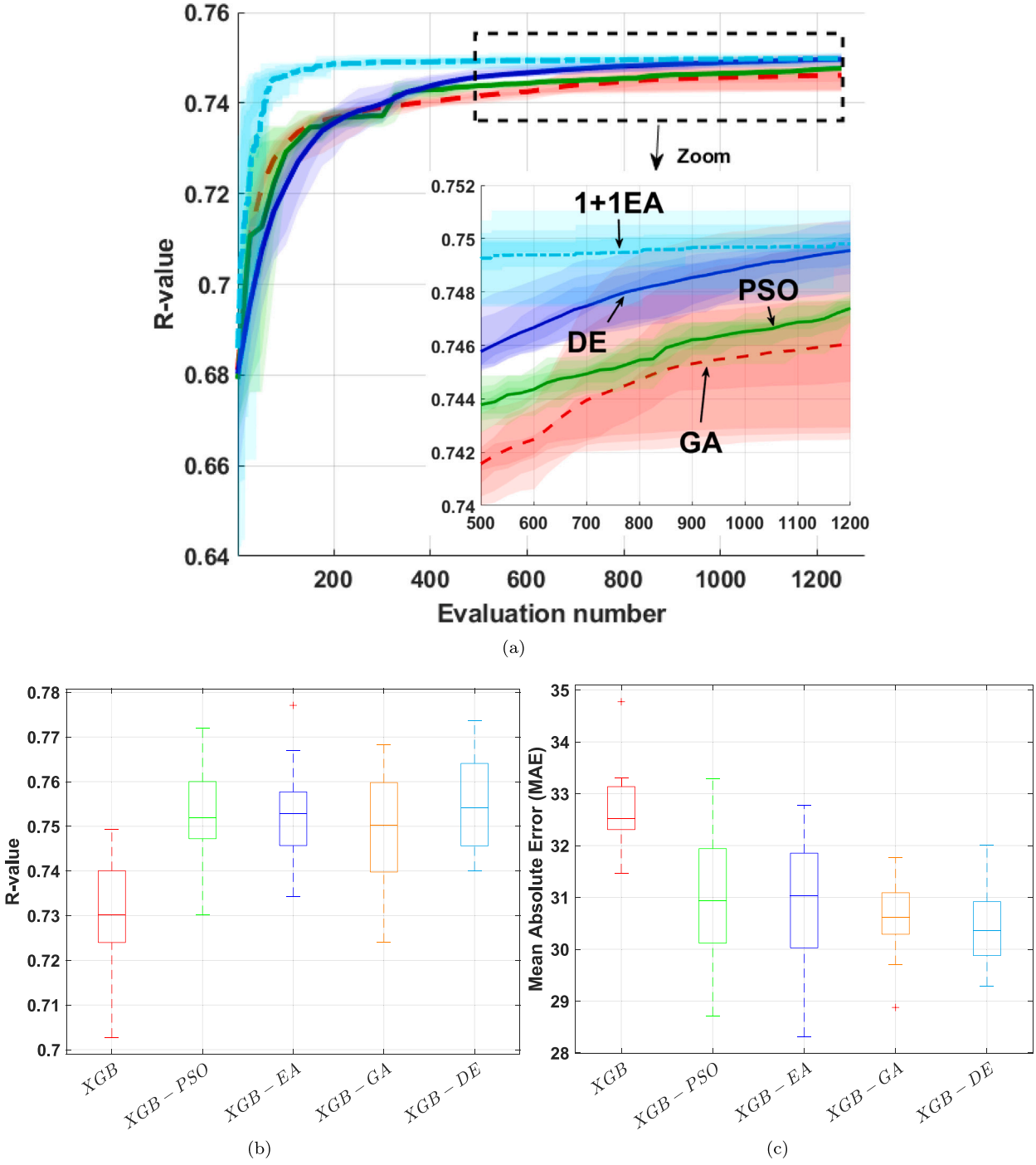

Fig. 13. (a) A convergence rate comparison for four neuro-evolutionary algorithms, including XGB-GA, XGB-DE, XGB-PSO, and XGB-EA. The lines show the average accuracy achieved by whole solutions in each generation.

the generality of the model using transfer learning techniques and cross-building training data to facilitate global deployment for energy prediction and management in various smart building setups.

5.2. Real-time and computational efficiency

The proposed adaptive models possess great potential for real-time deployment in smart building environments. By leveraging the use of lightweight learners, such as Extra Trees, within a Bagging framework and adjusting the parameters using computationally lightweight meta-heuristic algorithms, such as the 1+1 EA, the computational overhead at both the training and inference phases is significantly reduced. Due to its parallelisable nature, the Bagging framework facilitates simultaneous training and independent operation of numerous base models, making scaling simpler on multi-core or distributed systems. Additionally, the evolutionary optimisation method accelerates convergence to optimal model configurations by efficiently exploring the search space, which decreases the number of training iterations. These qualities make the proposed models highly suitable for real-time or near-real-time energy forecasting, where quick adaptation to new sensor readings is essential for dynamic energy management and demand-side response in smart buildings.

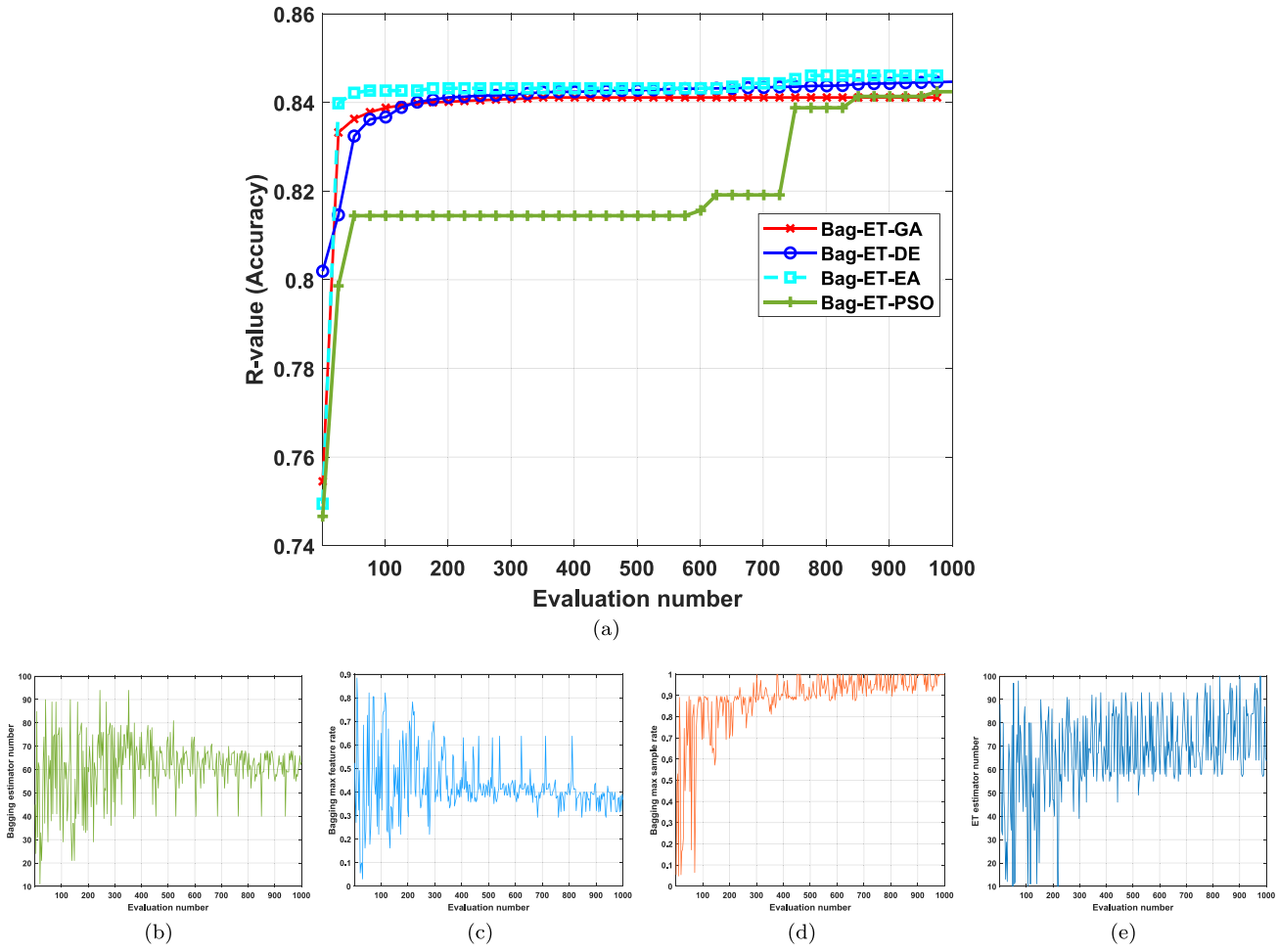

Fig. 14. (a) Convergence rate of Bagging Extra-tree's hyper-parameters tuning using four optimisation methods and the exploration of parameters search space, (b) Bagging estimator number, (c) maximum feature rate of Bagging, (d) maximum sample rate of Bagging, and (e) estimator number of Extra tree method.

Additionally, the framework's computational efficiency was verified by monitoring training and prediction run times during cross-validation experiments. Compared to traditional ensemble models such as boosting models (e.g., XGBoost, CatBoost, GBM) which involve sequential model updating and longer processing, the proposed Bagging-based model, enhanced by evolutionary tuning, consistently had lower computational costs without sacrificing predictive accuracy. This accuracy-efficiency trade-off ensures the practical viability of deploying the model in real building management systems, where timely forecasting is crucial for energy scheduling, load balancing, and integration with renewable sources. Thus, the hybrid evolutionary ensemble method improves the forecasting accuracy and meets the operational requirements of smart building applications in terms of speed, scalability, and resource efficiency.

5.3. Future directions

Future research will focus on enhancing the applicability and robustness of the proposed adaptive evolutionary ensemble models by their broader implementation in different building typologies and climatic zones. This will be realised by integrating diversified, large-scale datasets with varying occupancy schedules, appliance utilisation profiles, architectural features, types of HVAC systems, and external environmental factors such as solar irradiance, wind speed, and air quality. By including a more extensive set of input features, the model will generalise better to residential, commercial, and institutional buildings with different temporal and spatial patterns of energy consumption. Additionally, an effort will be made to integrate real-time data streaming into the prediction pipeline, allowing the model to operate in an online learning mode. This will enable the forecasting engine to adjust its parameters in real time as it receives new sensor data, thereby delivering more accurate and responsive control in dynamic energy management systems.

Advanced optimisation techniques, such as multi-objective evolutionary algorithms, cooperative coevolution, and meta-reinforcement learning, will be explored to attain further improvements in model convergence speed, scalability, and flexibility. Finally, incorporating renewable energy forecasting, such as photovoltaic and wind power generation, into the ensemble framework will help develop smart, carbon-aware decision-making systems. These enhancements will not only improve forecast accuracy but also enable real-time load balancing, demand-side management, and, ultimately, the decarbonisation and sustainability of future smart buildings.

Future research will also focus on applying the model developed to other forms of smart buildings with varying configurations and usage patterns. To enhance the objectivity and generalisability of the model, we also intend to incorporate standardised building classification systems and develop a taxonomy-based modelling process that accounts for variations in room types, appliance densities, and user usage patterns. In addition, applying the framework to multi-building datasets will provide cross-building validation and more scalable and policy-relevant energy forecasting solutions.

Having discussed the key outcomes and their relevance, the final section concludes the study by summarising the major contributions,

acknowledging inherent limitations, and outlining future directions for improving energy forecasting models in smart building environments.

6. Conclusion

In conclusion, the building sector accounts for a significant portion of global energy consumption and plays a crucial role in future decarbonisation efforts. Therefore, developing reliable and accurate energy demand forecasting models is crucial for effectively managing energy consumption and enhancing energy efficiency in smart buildings.

This paper addresses the challenges of predicting total energy use in smart buildings, complicated by temporal oscillations and complex linear and non-linear patterns. To overcome these challenges, the paper proposes three adaptive evolutionary ensemble models that integrate various bagging, stacking and voting models with a fast and effective evolutionary hyper-parameters tuner. Data filtering and automatic outlier removal techniques were also employed to extract relevant parameters and enhance prediction accuracy.

The proposed energy forecasting model was evaluated using a hybrid dataset encompassing meteorological parameters, appliance energy use, temperature, humidity, and lighting energy consumption data collected from 18 sensors in a Stambruges, Mons, Belgium building. To benchmark the performance of the proposed model, it was compared against 15 popular ML models, including classic ML models, neural networks, decision trees, random forests, deep learning models, and ensemble models. The findings demonstrate that the adaptive evolutionary bagging model outperformed the other prediction models in terms of accuracy and learning error. Specifically, it achieved accuracy improvements of 12.6%, 13.7%, 12.9%, 27.04%, and 17.4% compared to XGB, CatBoost, GBM, LGBM, and RF, respectively. These results highlight the effectiveness of the advanced evolutionary ensemble approach for energy demand forecasting in intelligent buildings. By surpassing the performance of various established ML models, the proposed model showcases its potential to enhance prediction accuracy and contribute to efficient energy management in smart buildings.

CRediT authorship contribution statement

Mehdi Neshat: Writing – original draft, Visualization, Validation, Software, Resources, Methodology, Investigation, Formal analysis, Conceptualization. **Menasha Thilakaratne:** Writing – review & editing, Resources, Investigation, Conceptualization. **Mohammed El-Abd:** Writing – review & editing, Investigation, Conceptualization. **Seyedali Mirjalili:** Writing – review & editing, Methodology, Investigation, Conceptualization. **Amir H. Gandomi:** Writing – review & editing, Methodology, Investigation, Conceptualization. **John Boland:** Writing – review & editing, Investigation, Conceptualization.

Declaration of Generative AI and AI-assisted technologies in the writing process

During the preparation of this paper, the authors utilised Grammarly and ChatGPT (GPT-4o) to improve English grammar, and overall writing quality. Following the use of these tools, the authors thoroughly reviewed and edited all content. The authors take full responsibility for the accuracy and integrity of the final version of the paper.

Declaration of competing interest

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

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.energy.2025.137130.

Data availability

Data will be made available on request.

References

[1] Wang N, Phelan PE, Harris C, Langevin J, Nelson B, Sawyer K. Past visions, current trends, and future context: A review of building energy, carbon, and sustainability. Renew Sustain Energy Rev 2018;82:976–93.

[2] González-Torres M, Pérez-Lombard L, Coronel JF, Maestre IR, Yan D. A review on buildings energy information: Trends, end-uses, fuels and drivers. Energy Rep 2022;8:626–37.

[3] Meschede H, Piacentino A, Guzovic Z, Lund H, Duic N. Integrated renewable energy systems as the basis for sustainable development of energy, water and environment systems. 2024.

[4] Hossain MF. Green science: Smart building technology to mitigate global energy and water crises. In: Climate change science. Elsevier; 2021, p. 223–46.

[5] Sulaiman MH, Mustaffa Z. Chiller energy prediction in commercial building: A metaheuristic-enhanced deep learning approach. Energy 2024;297:131159.

[6] Cheng Z, Yao Z. A novel approach to predict buildings load based on deep learning and non-intrusive load monitoring technique, toward smart building. Energy 2024;312:133456.

[7] Novosel T, Feijoo F, Duić N, Domac J. Impact of district heating and cooling on the potential for the integration of variable renewable energy sources in mild and mediterranean climates. Energy Convers Manage 2022;272:116374.

[8] Aguilar J, Garces-Jimenez A, R-moreno M, García R. A systematic literature review on the use of artificial intelligence in energy self-management in smart buildings. Renew Sustain Energy Rev 2021;151:111530.

[9] Schmidt M, Åhlund C. Smart buildings as cyber-physical systems: Data-driven predictive control strategies for energy efficiency. Renew Sustain Energy Rev 2018;90:742–56.

[10] Østergaard PA, Duic N, Noorollahi Y, Kalogirou S. Renewable energy for sustainable development. 2022.

[11] Jerominko T, Cichowicz R. Improving the energy efficiency of typical public buildings intended for education purposes located in the temperate climate zone in central and eastern Europe. Energy 2025;322:135542.

[12] Hatvani-Kovacs G, Belusko M, Pockett J, Boland J. Heat stress-resistant building design in the Australian context. Energy Build 2018;158:290–9.

[13] Yildiz B, Bilbao JI, Sproul AB. A review and analysis of regression and machine learning models on commercial building electricity load forecasting. Renew Sustain Energy Rev 2017;73:1104–22.

[14] Alanne K, Sierla S. An overview of machine learning applications for smart buildings. Sustain Cities Soc 2022;76:103445.

[15] Bourdeau M, qiang Zhai X, Nefzaoui E, Guo X, Chatellier P. Modeling and forecasting building energy consumption: A review of data-driven techniques. Sustain Cities Soc 2019;48:101533.

[16] Tien PW, Wei S, Calautit JK, Darkwa J, Wood C. Real-time monitoring of occupancy activities and window opening within buildings using an integrated deep learning-based approach for reducing energy demand. Appl Energy 2022;308:118336.

[17] Liu X, Ren M, Yang Z, Yan G, Guo Y, Cheng L, Wu C. A multi-step predictive deep reinforcement learning algorithm for HVAC control systems in smart buildings. Energy 2022;259:124857.

[18] Yang S, Wan MP, Chen W, Ng BF, Dubey S. Model predictive control with adaptive machine-learning-based model for building energy efficiency and comfort optimization. Appl Energy 2020;271:115147.

[19] Guzović Z, Duic N, Piacentino A, Markovska N, Mathiesen BV, Lund H. Recent advances in methods, policies and technologies at sustainable energy systems development. Energy 2022;245:123276.

[20] Somu N, MR GR, Ramamritham K. A deep learning framework for building energy consumption forecast. Renew Sustain Energy Rev 2021;137:110591.

[21] Mirjalili S. SCA: A sine cosine algorithm for solving optimization problems. Knowl-Based Syst 2016;96:120–33.

[22] Zhang C, Zhang J, Zhao Y, Lu J. Automated data-driven building energy load prediction method based on generative pre-trained transformers (GPT). Energy 2025;134824.

[23] Yesilyurt H, Dokuz Y, Dokuz AS. Data-driven energy consumption prediction of a university office building using machine learning algorithms. Energy 2024;310:133242.

[24] Pachauri N, Ahn CW. Weighted aggregated ensemble model for energy demand management of buildings. Energy 2023;263:125853.

[25] Mohan R, Pachauri N. An ensemble model for the energy consumption prediction of residential buildings. Energy 2025;314:134255.

[26] Liu Y, Sun Y, Gao D-c, Tan J, Chen Y. Stacked ensemble learning approach for PCM-based double-pipe latent heat thermal energy storage prediction towards flexible building energy. Energy 2024;294:130955.

[27] He Y, Zhang H, Dong Y, Wang C, Ma P. Residential net load interval prediction based on stacking ensemble learning. Energy 2024;296:131134.

[28] Zhu Y, Xu W, Luo W, Yang M, Chen H, Liu Y. Application of hybrid machine learning algorithm in multi-objective optimization of green building energy efficiency. Energy 2025;316:133581.
[29] Xu W, Wu X, Xiong S, Li T, Liu Y. Optimizing the sustainable performance of public buildings: A hybrid machine learning algorithm. Energy 2025;320:135283.
[30] Coraci D, Brandi S, Capozzoli A. Effective pre-training of a deep reinforcement learning agent by means of long short-term memory models for thermal energy management in buildings. Energy Convers Manage 2023;291:117303.
[31] Liang X, Chen K, Chen S, Zhu X, Jin X, Du Z. IoT-based intelligent energy management system for optimal planning of HVAC devices in net-zero emissions PV-battery building considering demand compliance. Energy Convers Manage 2023;292:117369.
[32] Candanedo LM, Feldheim V, Deramaix D. Data driven prediction models of energy use of appliances in a low-energy house. Energy Build 2017;140:81–97.
[33] Feist W, Pfluger R, Kaufmann B, Schnieders J, Kah O. Passive house planning package 2007. In: Specifications for quality approved passive houses, Technical information PHI-2007/1 (E). Darmstadt: Passivhaus Institut; 2007.
[34] Breunig MM, Kriegel H-P, Ng RT, Sander J. LOF: Identifying density-based local outliers. In: Proceedings of the 2000 ACM SIGMOD international conference on management of data. 2000, p. 93–104.
[35] Smiti A. A critical overview of outlier detection methods. Comput Sci Rev 2020;38:100306.
[36] Yao Q, Zhu H, Xiang L, Su H, Hu A. A novel composed method of cleaning anomy data for improving state prediction of wind turbine. Renew Energy 2023;204:131–40.
[37] Neshat M. The application of nature-inspired metaheuristic methods for optimising renewable energy problems and the design of water distribution networks [Ph.D. thesis], University of Adelaide Adelaide, Australia; 2020.
[38] Neshat M, Nezhad MM, Abbasnejad E, Groppi D, Heydari A, Tjernberg LB, Garcia DA, Alexander B, Wagner M. Hybrid neuro-evolutionary method for predicting wind turbine power output. 2020, arXiv preprint arXiv:2004.12794.
[39] Storn R, Price K. Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. J Global Optim 1997;11(4):341–59.
[40] Zaharie D. Influence of crossover on the behavior of differential evolution algorithms. Appl Soft Comput 2009;9(3):1126–38.
[41] Piotrowski AP. Review of differential evolution population size. Swarm Evol Comput 2017;32:1–24.
[42] Katoch S, Chauhan SS, Kumar V. A review on genetic algorithm: Past, present, and future. Multimedia Tools Appl 2021;80:8091–126.
[43] Costa-Carrapiço I, Raslan R, González JN. A systematic review of genetic algorithm-based multi-objective optimisation for building retrofitting strategies towards energy efficiency. Energy Build 2020;210:109690.
[44] Michalewicz Z, Nazhiyath G, Michalewicz M. A note on usefulness of geometrical crossover for numerical optimization problems.. Evol Program 1996;5(1):305–12.
[45] Friedrich T, Kötzing T, Lagodzinski G, Neumann F, Schirneck M. Analysis of the (1+ 1) EA on subclasses of linear functions under uniform and linear constraints. In: Proceedings of the 14th ACM/SIGEVO conference on foundations of genetic algorithms. 2017, p. 45–54.
[46] Neshat M, Alexander B, Simpson A. Covariance matrix adaptation greedy search applied to water distribution system optimization. 2019, arXiv preprint arXiv:1909.04846.
[47] Neumann F, Wegener I. Randomized local search, evolutionary algorithms, and the minimum spanning tree problem. Theoret Comput Sci 2007;378(1):32–40.
[48] Mienye ID, Sun Y. A survey of ensemble learning: Concepts, algorithms, applications, and prospects. IEEE Access 2022;10:99129–49.
[49] Yu Y, Chen Q, Zhi J, Yao X, Li L, Shi C. Carbon peak prediction in China based on bagging-integrated GA-BiLSTM model under provincial perspective. Energy 2024;313:133519.
[50] Bian J, Wang J, Yece Q. A novel study on power consumption of an HVAC system using CatBoost and AdaBoost algorithms combined with the metaheuristic algorithms. Energy 2024;131841.
[51] Chen T, Guestrin C. Xgboost: A scalable tree boosting system. In: Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining. 2016, p. 785–94.
[52] Neshat M, Sergiienko NY, Rafiee A, Mirjalili S, Gandomi AH, Boland J. Meta wave learner: Predicting wave farms power output using effective meta-learner deep gradient boosting model: A case study from Australian coasts. Energy 2024;304:132122.
[53] Dong X, Yu Z, Cao W, Shi Y, Ma Q. A survey on ensemble learning. Front Comput Sci 2020;14:241–58.
[54] Wolpert DH. Stacked generalization. Neural Netw 1992;5(2):241–59.
[55] Dong Y, Zhang H, Wang C, Zhou X. Wind power forecasting based on stacking ensemble model, decomposition and intelligent optimization algorithm. Neurocomputing 2021;462:169–84.
[56] Kesriklioğlu E, Oktay E, Karaaslan A. Predicting total household energy expenditures using ensemble learning methods. Energy 2023;276:127581.
[57] Kim D, Baek J-G. Bagging ensemble-based novel data generation method for univariate time series forecasting. Expert Syst Appl 2022;203:117366.
[58] González S, García S, Del Ser J, Rokach L, Herrera F. A practical tutorial on bagging and boosting based ensembles for machine learning: Algorithms, software tools, performance study, practical perspectives and opportunities. Inf Fusion 2020;64:205–37.
[59] Yang Y, Lv H, Chen N. A survey on ensemble learning under the era of deep learning. Artif Intell Rev 2023;56(6):5545–89.
[60] Qiu R, Liu C, Cui N, Gao Y, Li L, Wu Z, Jiang S, Hu M. Generalized extreme gradient boosting model for predicting daily global solar radiation for locations without historical data. Energy Convers Manage 2022;258:115488.
[61] Bentéjac C, Csörgő A, Martínez-Muñoz G. A comparative analysis of gradient boosting algorithms. Artif Intell Rev 2021;54:1937–67.
[62] Svetnik V, Liaw A, Tong C, Culberson JC, Sheridan RP, Feuston BP. Random forest: A classification and regression tool for compound classification and QSAR modeling. J Chem Inf Comput Sci 2003;43(6):1947–58.
[63] Freund Y, Schapire RE, et al. Experiments with a new boosting algorithm. In: Icml. vol. 96, Citeseer; 1996, p. 148–56.
[64] Friedman JH. Greedy function approximation: A gradient boosting machine. Ann Stat 2001;1189–232.
[65] Ke G, Meng Q, Finley T, Wang T, Chen W, Ma W, Ye Q, Liu T-Y. Lightgbm: A highly efficient gradient boosting decision tree. Adv Neural Inf Process Syst 2017;30.
[66] Prokhorenkova L, Gusev G, Vorobev A, Dorogush AV, Gulin A. CatBoost: Unbiased boosting with categorical features. Adv Neural Inf Process Syst 2018;31.
[67] Friedman D, Dieng AB. The vendi score: A diversity evaluation metric for machine learning. 2022, arXiv preprint arXiv:2210.02410.
[68] Han Y, Fan C, Geng Z, Ma B, Cong D, Chen K, Yu B. Energy efficient building envelope using novel RBF neural network integrated affinity propagation. Energy 2020;209:118414.