A

DISSERTATION REPORT

ON

Stacked Ensemble Learning with Heap Based Optimization for Building Energy Consumption

Submitted in partial fulfilment of the requirements for the award of degree of

MASTER OF TECHNOLOGY

in

POWER SYSTEM MANAGEMENT

Submitted By:
Puneet Pahadia
(2021PSM5643)

Supervised By:
Mr. Ashok Kumar Agarwal
Dr. Prerna Jain
Associate Professor

DEPARTMENT OF ELECTRICAL ENGINEERING
MALAVIYA NATIONAL INSTITUTE OF TECHNOLOGY JAIPUR, INDIA

JUNE, 2023

© PUNEET PAHADIA

© Malaviya National Institute of Technology Jaipur-2023

All rights reserved

© PUNEET PAHADIA

MALAVIYA NATIONAL INSTITUTE OF TECHNOLOGY JAIPUR

CERTIFICATE

This is to certify that the dissertation entitled **"Stacked Ensemble Learning with Heap Based Optimization for Building Energy Consumption"** submitted by **Mr. Puneet Pahadia (2021PSM5643)** at **Malaviya National Institute of Technology Jaipur** towards partial fulfillment of the requirements for the award of the degree of **Master of Technology in Power System Management at Department of Electrical Engineering** is a bonafide record of the work carried out by him under my supervision.

Mr. Ashok Kumar Agarwal
Associate Professor
Department of Electrical Engineering
MNIT Jaipur
Jaipur -302017, India

Dr. Prerna Jain
Associate Professor
MNIT Jaipur
Department of Electrical Engineering
Jaipur -302017, India

Place: Jaipur

Date: 28 June,2023

© PUNEET PAHADIA

MALAVIYA NATIONAL INSTITUTE OF TECHNOLOGY JAIPUR

DECLARATION

I hereby declare that the dissertation entitled **"Stacked Ensemble Learning with Heap Based Optimization for Building Energy Consumption"** being submitted by me in partial fulfilment of the requirements for the award of the Degree of **Master of Technology in Power Systems Management at Department of Electrical Engineering, Malaviya National Institute of Technology Jaipur** is an authentic record of my own work, carried out under the supervision of **Dr. Prerna Jain**, Associate Professor, Malaviya National Institute of Technology Jaipur and **Mr. Ashok Kumar Agarwal**, Associate Professor, Malaviya National Institute of Technology Jaipur.

The matter presented in this dissertation embodies the result of my own work and studies carried out by me. The contents of this dissertation work, in full or in parts, have not been submitted for the award of any other degree of this or any other Institute.

Place: Jaipur **Puneet Pahadia**

Date: **(2021PSM5643)**

© PUNEET PAHADIA

ACKNOWLEDGEMENT

I take this opportunity to express my gratitude to my supervisors **Dr. Prerna Jain**, Associate Professor in Department of Electrical Engineering at Malaviya National Institute of Technology and **Mr. Ashok Kumar Agarwal**, Associate Professor in Department of Electrical Engineering at Malaviya National Institute of Technology for their invaluable guidance and encouragement during the research and the preparation of this thesis. They have been the principal motivating force behind this work and provided all kind of support. I am very much thankful to them for extending maximum possible help at times of need.

I express my indebtedness to **Dr. Harpal Tiwari**, Head, Department of Electrical Engineering. Malaviya National Institute of Technology Jaipur, for creating an encouraging and pleasant research environment.

I would like to thank DPGC convener **Dr. Praveen Kumar Agrawal** for giving valuable suggestions regarding my work and encouraged me to keep my enthusiasm high to complete my work.

I am deeply thankful to PG-Coordinator **Dr. Rohit Bhakar** who gave me moral as well as technical support which helped me to keep my enthusiasm high to complete my work.

I would also like to express my thanks to the PhD scholars **Abhishek Harit, Ashish Prajesh and Amit Kumar** for introducing me the research area and helping me to get through this dissertation.

And many thanks to **Piyush Kaloriya**, **Harshit Rathore and Swati Sharma** for support, co-operation, and discussion throughout the project work.

It gives me immense pleasure in acknowledging the love and encouragement showered on me by parents. Their financial and moral support helped me to accomplish my work.

PUNEET PAHADIA

(2021PSM5643)

© PUNEET PAHADIA

ABSTRACT

Energy Management Systems (EMS) can generate a vast quantity of data-points, which can be used to forecast the power consumption using machine learning-based techniques. In recent years, especially ensemble techniques, demonstrated prominent performance in time-series forecasting and prediction applications. The utilisation of the stacking method is considered to be a highly effective approach within the domain of Ensembling techniques. We used it in this research to predict Building Energy Consumption based on electrical and climatic datasets. We utilized the latest metaheuristic optimization algorithm, Heap-Based Optimization (HBO), to train the ensemble model and optimize its parameters, aiming to improve its accuracy. HBO is a novel human-behaviour-based metaheuristic algorithm used for solving complex optimisation and engineering dilemmas. It was motivated by corporate rank hierarchy. In the present research, HBO has been employed for training the ensemble model, and the ensemble prediction performance is significantly improved. For assessing the developed HBO-Ensemble, we made use of two datasets. In addition to existing models, we compared several optimised models using various optimisation algorithms, as well as several optimised models. The outcome of the comparison affirmed HBO's capacity to improve the forecasting accuracy of the stacked ensemble model.

© PUNEET PAHADIA

LIST OF CONTENTS

© PUNEET PAHADIA

© PUNEET PAHADIA

© PUNEET PAHADIA

LIST OF TABLES

© PUNEET PAHADIA

LIST OF FIGURES

© PUNEET PAHADIA

LIST OF ABBREVIATIONS

Abbreviation	Full form
AI	Artificial Intelligence
ANN	Artificial Neural Network
CNN	Convolutional Neural Networks
CRH	Corporate Rank Hierarchy
DL	Deep Learning
EL	Ensemble Learning
ERT	Extremely Randomized Trees
FFNNs	Feedforward Neural Networks
GBM	Gradient Boosting Machine
HBO	Heap Based Optimization
HVAC	Heating, Ventilation, and Air Conditioning
Iot	Internet of Things
KNN	K-Nearest Neighbour
LSTM	Long Short Term Memory
MAPE	Mean Absolute Percentage Error
ML	Machine Learning
MLR	Multiple Linear Regression
RBF	Radial Basis Function
RF	Random Forest
RMSE	Root Mean Square Error
RNNs	Recurrent Neural Networks
RR	Ridge Regression
SVM	Support Vector Machine

© PUNEET PAHADIA

Chapter: 1

Introduction

1.1 General

Building energy consumption forecasting using artificial intelligence (AI) involves leveraging historical energy data and contextual information to develop accurate models for forecasting energy usage. AI methodologies, such as deep learning and machine learning, enable analysis of complex datasets and provide valuable insights for energy management.

Machine learning algorithms, including regression models, SVM, RF, and neural networks, are commonly used for energy consumption prediction. Deep learning methodologies, such as RNNs and LSTMs, excel at capturing temporal dependencies in time-series data.

The application of AI in building energy prediction offers benefits such as optimized energy usage, proactive maintenance planning, and implementation of energy-saving strategies. AI models help with load balancing, demand response management, and efficient scheduling of energy resources.

AI also aids in identifying anomalies and energy waste, enabling proactive interventions and energy conservation measures. Real-time energy monitoring and feedback systems empower occupants to make informed decisions regarding energy usage.

Building energy consumption prediction using AI holds promise for optimizing energy usage and promoting sustainability. By leveraging advanced AI algorithms, accurate forecasting models can be developed, leading to efficient energy management in the built environment.

1.2 Energy Management

Energy management strategies are essential for optimizing energy usage, reducing energy waste, and achieving energy efficiency goals. One of the key reasons for the need for energy management is the limited availability of natural resources and the environmental impact associated with energy generation and consumption. The International Energy Agency (IEA) predicts that global energy demand will continue to

© PUNEET PAHADIA

rise in the coming decades, making it imperative to adopt measures to enhance energy efficiency and conservation.

Energy management also addresses the economic aspect of energy consumption. Businesses and industries incur significant costs related to energy usage, making it essential to optimize energy consumption to reduce operational expenses. A study conducted by the United Nations Industrial Development Organization (UNIDO) estimates that energy efficiency measures can result in substantial cost savings for industries.

Furthermore, energy management plays a vital role in achieving sustainability goals and mitigating climate change. The reduction of greenhouse gas emissions and the transition towards renewable energy sources are key objectives of energy management. The United Nations Sustainable Development Goals (SDGs) emphasize the importance of sustainable energy practices to combat climate change.

To implement effective energy management, the utilization of advanced technologies and data-driven approaches is crucial. Energy management systems (EMS) equipped with sensors, automation, and machine learning techniques enable real-time monitoring, analysis, and optimization of energy consumption.

1.3 Building Energy Management

In recent years, the need for effective building energy management has become increasingly critical due to growing concerns about energy efficiency, environmental sustainability, and cost optimization. Buildings play a crucial role in energy consumption, constituting share of the overall global energy usage. Therefore, implementing robust energy management strategies is essential for reducing energy waste, minimizing carbon emissions, and achieving long-term energy sustainability goals.

Building energy management systems (BEMS) play a pivotal role in addressing these challenges by enabling real-time monitoring, analysis, and control of energy consumption within buildings. BEMS utilise cutting-edge technologies such as IoT and ML to offer significant insights and streamline energy optimisation procedure.

Efficient building energy management offers numerous benefits. Firstly, it helps reduce energy costs by identifying energy-saving opportunities, optimizing HVAC systems, and implementing demand response strategies. Secondly, it enhances occupant comfort

© PUNEET PAHADIA

and productivity through improved indoor air quality, temperature control, and lighting management. Thirdly, effective energy management contributes to environmental conservation by reducing greenhouse gas emissions and overall environmental footprint.

To achieve these benefits, research efforts have focused on developing innovative methodologies and algorithms for building energy prediction, optimization, and control. Machine learning techniques, such as ensemble methods, have shown promising results in accurately forecasting building energy consumption and optimizing energy usage patterns. Moreover, novel optimization algorithms, such as Heap-Based Optimization (HBO), have emerged as effective tools for training ensemble models and optimizing their parameters, resulting in improved accuracy and performance.

1.4 Artificial Intelligence for Building Energy Management

AI has emerged as a method for optimizing energy management in buildings, leading to improved energy efficiency and cost savings. This write-up aims to provide a concise overview of the applications of AI in building energy management.

AI models, such as ML and data analytics, enable the analysis of large volumes of data generated by energy management systems (EMS) in buildings. These data points encompass a wide range of parameters, including weather conditions, occupancy patterns, equipment performance, and energy consumption profiles. By leveraging machine learning algorithms, accurate models can be developed to forecast energy consumption patterns and optimize building operations.

One notable application of AI in building energy management is demand response. AI algorithms can effectively analyze real-time energy demand and price data to optimize energy usage by adjusting heating, cooling, and lighting systems accordingly. This enables buildings to participate in demand response programs, contributing to grid stability and reducing energy costs.

Furthermore, AI can enhance fault detection and diagnostics in building systems. Through the analysis of sensor data, artificial intelligence (AI) algorithms have the capability to detect anomalies and deviations from typical operational states. This enables facility managers to be promptly notified of potential equipment malfunctions or instances of energy wastage. Proactive maintenance and prompt remediation based

© PUNEET PAHADIA

on AI-driven insights can significantly improve energy efficiency and minimize downtime.

Moreover, AI-based occupant behaviour modelling can provide valuable insights into individual energy consumption patterns, allowing for personalized energy management strategies. By understanding occupant preferences and usage patterns, buildings can optimize HVAC settings, lighting controls, and other energy-consuming systems to maximize comfort while minimizing energy waste.

1.5 Hyperparameters and Metaheuristic Algorithms

Hyperparameters play a crucial role in machine learning models as they define the configuration and behaviour of the algorithms. Optimizing hyperparameters is essential to achieve optimal model performance. In recent years, metaheuristic algorithms have gained attention as effective tools for hyperparameter optimization in various domains.

Metaheuristic algorithms are optimization techniques inspired by natural processes or human behaviour. They offer a robust approach to finding near-optimal solutions in complex optimization problems. These algorithms explore the search space and adaptively refine the hyperparameter values to improve model performance.

One such metaheuristic algorithm is Heap-Based Optimization (HBO), which is motivated by corporate rank hierarchy. HBO has shown promising results in solving complex optimization and engineering dilemmas. In the context of machine learning, HBO has been employed to train ensemble models and optimize their hyperparameters.

Recent studies have examined the application of HBO in ensemble models for the purpose of forecasting building energy consumption. The study incorporated electrical and climatic datasets to develop an ensemble model. HBO was employed to train the model and optimize its hyperparameters, resulting in improved accuracy in energy consumption prediction.

To evaluate the effectiveness of HBO, the developed HBO-Ensemble model was compared with existing models and various optimized models using different optimization algorithms. The comparison affirmed HBO's ability to enhance the forecasting accuracy of the stacked ensemble model.

The combination of ensemble techniques and metaheuristic algorithms, such as HBO, holds promise for improving the performance of machine learning models in various

© PUNEET PAHADIA

domains. By efficiently searching the hyperparameter space, these algorithms contribute to the optimization and fine-tuning of models, leading to more accurate predictions.

1.6 Motivation for the Present Work

Efficient energy management in buildings is crucial for sustainable development and reducing carbon emissions. Predicting building energy consumption accurately plays a pivotal role in optimizing energy usage, improving energy efficiency, and implementing effective demand-side management strategies. Traditional approaches to energy consumption estimation have limitations in capturing the complex dynamics of energy usage patterns. Therefore, the motivation arises to utilize machine learning-based techniques for forecasting building energy consumption, leveraging the vast quantity of data generated by Energy Management Systems (EMS).

1.6.1 Machine Learning for Energy Consumption Prediction

In last decades, ML methods have shown promise in time-series forecasting and prediction applications. Ensemble techniques, in particular, have demonstrated remarkable performance by combining the strengths of multiple models. The stacking method, a popular ensemble technique, has been increasingly used for its effectiveness in capturing the intricate relationships between electrical and climatic datasets.

1.6.2 Heap-Based Optimization (HBO) Algorithm

To optimize the ensemble model's accuracy, this research employs the Heap-Based Optimization (HBO) algorithm. HBO is a novel metaheuristic optimization algorithm inspired by the hierarchical structure of corporate ranks. This algorithm is known for its ability to solve complex optimization and engineering dilemmas efficiently. By utilizing HBO, the ensemble model's parameters are fine-tuned, leading to improved accuracy in predicting building energy consumption.

1.6.3 Validation and Comparison

To assess the effectiveness of the HBO-Ensemble model, two datasets were utilized. Additionally, various optimization algorithms and optimized models were compared against the existing models. The comparison revealed HBO's capability to enhance the forecasting accuracy of the stacked ensemble model.

© PUNEET PAHADIA

1.7 Contribution of the Present Work

This study contributes to the field of building energy consumption prediction by showcasing the efficacy of the HBO-Ensemble technique. By leveraging the HBO algorithm's metaheuristic optimization capabilities, the ensemble model achieved improved accuracy in forecasting building energy consumption. The findings highlight the potential of HBO as a valuable tool in optimizing energy management systems and enhancing resource utilization in buildings.

1.8 Organization of Thesis

This thesis is organized into several chapters to give a comprehensive understanding of research conducted on building's energy consumption forecasting using HBO-ensemble technique. The chapters are structured as follows:

- **Chapter 1:** Provides an overview of the research topic, the significance of predicting building energy consumption, and the objective.
- **Chapter 2:** Gives the Review of the relevant literature and discusses concepts, theories, and previous studies related to energy management systems, artificial intelligence, and machine learning techniques in predicting building energy consumption.
- **Chapter 3:** Presents the related theories and concepts, including various machine learning algorithms and ensembling techniques, as well as an in-depth exploration of the Heap-Based Optimization (HBO) algorithm and its mathematical modelling.
- **Chapter 4:** Describes the case study conducted, including the description of datasets used, data pre-processing techniques applied, outlier detection and removal, correlation analysis, and feature selection methods employed.
- **Chapter 5:** Presents the results obtained from the experiments and evaluations conducted, including the evaluation metrics employed, model evaluation results, comparison of proposed HBO-Ensemble model along with the traditional EM, and the feasibility analysis of the proposed model.
- **Chapter 6:** Summarizes the research findings, discusses the implications of the study, and provides recommendations for future research.

© PUNEET PAHADIA

Chapter: 2

Literature Survey

2.1 Introduction

Building energy consumption accounts for a significant portion of global energy consumption and carbon emissions. About 40% of worldwide energy use and 30% of the world's carbon footprint are attributable to buildings [1]. This has led to increased attention towards building energy efficiency and sustainability. The forecasting of building energy consumption is a crucial aspect in optimizing energy usage and enhancing sustainability within the built environment. Accurate energy prediction can help building owners and managers to optimize energy usage and lower energy costs.

Traditionally, building energy prediction relies on energy simulation software, which requires detailed building models and significant computational resources. However, energy simulation software needs accurate inputs for the building model, which can be challenging to obtain, and the computational requirements for energy simulations can be time-consuming and costly.

In recent years, AI techniques, such as ML, DL, and EL, have emerged as effective tools for building energy prediction. AI techniques are capable of analysing vast volumes of data and identifying patterns and correlations that are difficult to discern using conventional methods.

Numerous studies have been conducted using AI techniques for building energy prediction. ML techniques have been employed to forecast energy usage based on a number of variables, such as weather conditions, building characteristics, and occupancy rates [2]. DL techniques, such as convolutional neural networks (CNN) and recurrent neural networks (RNN), have been used for feature extraction and time-series prediction in building energy consumption [3]. EL methods, include randomized forest (RF) and XGBoost, have been used to enhance the accuracy of building energy prediction models [4].

Notwithstanding the prospective advantages of utilising AI methodologies for developing energy forecasting models, there exist certain obstacles and constraints that necessitate resolution. Data quality and availability can be an issue, especially in older buildings that lack modern sensors and data acquisition systems. Additionally, the

© PUNEET PAHADIA

interpretability of AI models can be challenging, and it is essential to understand how these models work to gain insights into building energy usage and identify opportunities for optimization.

2.2 AI in Building Energy Prediction: Literature Review

Several recent studies have explored various machine learning (ML) techniques for predicting building energy consumption. Using historical data, a k-nearest neighbour (KNN) algorithm was used to predict hourly building energy consumption [5]. The results revealed that the KNN algorithm outperformed other ML algorithms, such as artificial neural networks (ANNs) and support vector regression (SVR). Similarly, [6] utilized a decision tree algorithm to predict the daily energy use of a building using occupancy as well as weather information. The findings of the research demonstrate that the decision tree algorithm exhibited a notable level of predictive precision, with an accuracy rate of up to 95%.

Apart from ML, DL techniques have also been employed for building energy prediction. Study [7] proposed a hybrid CNN-LSTM model that relied on weather data to predict daily building energy consumption. Given model demonstrated a high prediction accuracy of 90%, surpassing other DL models such as standard LSTM and CNN models. Additionally, [8] presented a deep residual network (ResNet) to predict the hourly energy usage of a building based on its characteristics, occupancy, and weather data. The ResNet model obtained a high prediction accuracy of up to 93%.

Ensemble learning (EL) techniques have also been explored for building energy prediction. EL integrates multiple ML or DL models to enhance prediction accuracy. For instance, [9] employed an ensemble model composed of a random forest and an LSTM model to predict hourly building energy consumption using weather data. The findings of the study demonstrated that the ensemble model exhibited superior performance compared to the individual models, achieving a maximum prediction accuracy of 94%.

2.3 ML in Building Energy Prediction: Literature Review

ML is a subset of AI that enables unprogrammed data-driven learning and prediction by computers [10]. ML techniques are often used to predict how much energy a building will use because they can look at a lot of data and find patterns and relationships that are hard to find with traditional methods.

© PUNEET PAHADIA

In study [9], ML techniques were used to forecast the energy usage of a building over short term. The study compared various ML algorithms, including linear regression, DT, KNN, SVM, and ANN to predict consumption of energy based on weather conditions, building characteristics, and occupancy rates. The study found that the artificial neural network outperformed the other algorithms, with a mean absolute percentage error of 2.32%.

Another study [11] compared various DL approaches for building energy prediction, including CNN, RNN, and LSTM. The study found that the LSTM model outperformed the other models, with a root mean square error of 0.64.

In addition, EL techniques, such as randomized forest and XGBoost, have been used to enhance the accuracy of building energy prediction models. Ref. [12] used gradient boosting decision trees for forecasting building's energy usage and implemented variable importance analysis to determine the most significant features affecting energy usage. According to the study, the variables that had the most significant impact on building energy consumption were temperature, relative humidity, and outdoor wind speed. Stacked generalisation, also known as stacking, is an approach to ensemble learning that leverages the combination of multiple models in order to enhance predictive accuracy. In stacking, a meta-model is trained to learn how to combine the outputs of several base models to generate the final prediction. One popular variant of stacking is the stacked generalization ensemble, which involves using a set of diverse models as base models and training a meta-model on their predictions.

© PUNEET PAHADIA

Chapter: 3

Related Theories

3.1 Ridge Regression

Ridge Regression (RR) is a method used in linear regression to handle correlation between independent variables. When predictor variables are strongly correlated, the least-squares predictor may exhibit higher variance, instability, and poor predictive accuracy, leading to overfitting. To overcome this, RR adds a penalty term to the least-squares fitness function, reducing the parameters towards zero, and thus lowering their variance and improving the model's performance.

The amount of shrinkage is determined by the regularisation parameter, λ which needs to be selected with care. A higher value causes more contraction, resulting in smaller coefficients and less variance, but potentially more bias [13]. In contrast, a lower value reduces shrinkage, resulting in larger coefficients and greater variance, but possibly lower bias. The optimal λ values can be determined using cross-validation or other methods.

RR is beneficial in situations with many highly correlated predictor variables, where the goal is to choose a subset of variables that are most predictive, avoiding overfitting, and stabilizing the least-squares estimator when there is noise or measurement error in the data [14]. However, RR assumes that all variables are equally important, and may not be appropriate when some variables are known to be more relevant than others [13]. In such cases, techniques like lasso regression or elastic net regression may be more appropriate.

The Cost function of the RR is as follows-

$$J(\Theta) = \frac{1}{m}(X\Theta - Y)^2 + \alpha \frac{1}{2}(\Theta)^2 \tag{3.1}$$

Here, $\frac{1}{m}(X\Theta - Y)^2$ is linear regression, and $\alpha \frac{1}{2}(\Theta)^2$ is new regularised weights term that fits the data using the L2 norm. If α equals zero, the model is identical to linear regression, whereas a larger α indicates a stronger regularisation. Before employing the RR, the inputs must be scaled.

© PUNEET PAHADIA

3.2 Random Forest(RF)

Ensemble learning algorithms, such as Extremely Randomised Trees and RF, belong to a category of techniques that utilise collective decision-making processes. Ensemble learning algorithm combine the capabilities of multiple learning algorithms to complete a task [15]. In classification task, for instance, an ensemble algorithm may combine the predictions of multiple classifiers to make decent prediction.

During training, a RF constructs several decision trees from distinct sets of training data. Similarly, the concept remains the same, results from multiple combined trees seem to be superior when compared with a single tree.

RF algorithm will sample sets of samples with distinct features from a dataset with multiple features. A decision tree is built using this subset. This technique of sampling sets with alternates is referred to as bootstrapping [15].

While building the decision tree, RF will select the optimal division at each node. Subsequently, the aforementioned approach is iteratively applied to a discrete subpopulation of the dataset that possesses unique attributes until the intended quantity of decision trees has been generated. After getting the results from each and every tree, the final forecast will be determined by majority of votes for classification or by arithmetic mean for regression.

The Gini Index can be defined as -

$$\text{Gini Index}= 1-\sum_{1}^{n}\left(P_i\right)^2 = 1-[(P_{(+)})^2+(P_{(-)})^2] \tag{3.2}$$

Where $P_{(+)}$ the probability of a positive is class and $P_{(-)}$ is the probability of a negative class. There is another metric called “Entropy” which is also used to measure the impurity of the split. Mathematically entropy is given as:

$$\text{Entropy}\left(E\right) = -p_{(+)}\log p_{(+)} - p_{(-)}\log p_{(-)} \tag{3.3}$$

© PUNEET PAHADIA

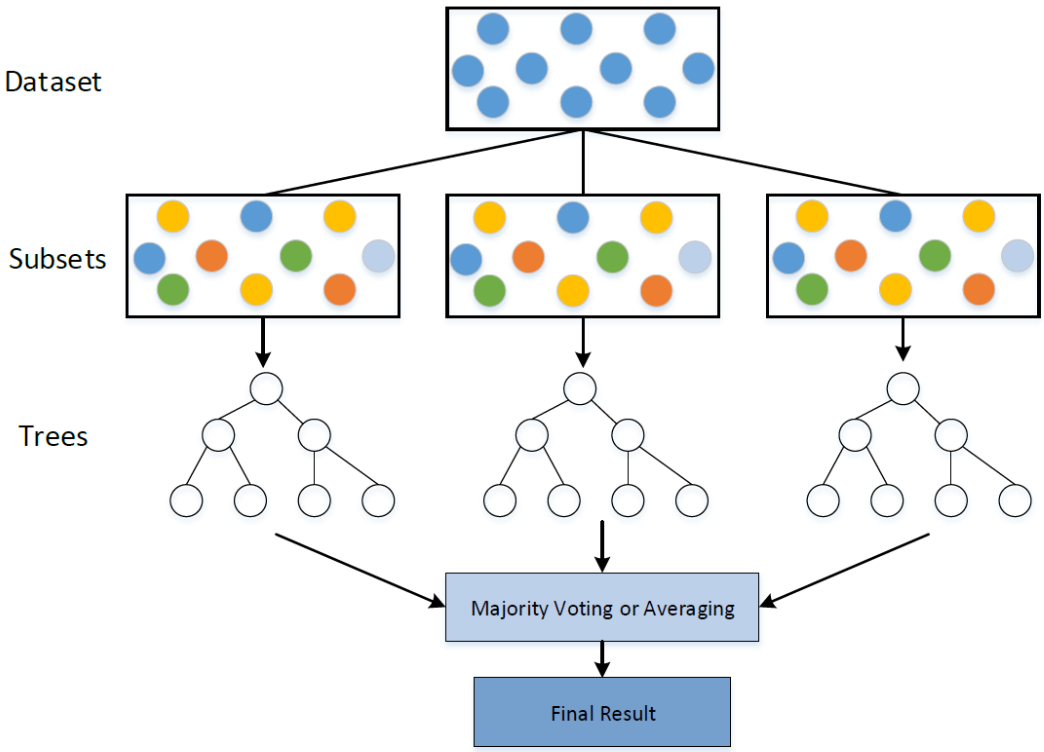


Figure 1 Structure of RF Model

3.3 Extremely Randomized Trees (ERT)

ERT algorithm generates an abundance of self-generated decision trees out from training sample. Forecasts are made by aggregating the forecasts of the decision trees throughout the regression or through majority vote in the classification [16]. The forecasts of the trees are averaged to produce the final forecast, by majority vote through arithmetic mean in regression problems and classification problems.

The algorithm has three primary hyperparameters that must be tuned: the minimum sample count required in a cluster to create an additional split point, total number of input traits to indiscriminately pick and examine for each split point, and the count of decision trees [17].

The random sample of split points reduces the correlation between the ensembled DT, but boosts the algorithm's variance. Increasing the count of trees included in the ensemble can counteract this rise in variance.

The formula for calculating the entropy is:-

$$Entropy(E) = \sum_{i=1}^{c} -p_i \log_2(p_i) \quad (3.4)$$

© PUNEET PAJA

Where c is the number of unique class labels and p_i is the proportion of rows with output label is *i*.

3.4 Gradient Boosting Machine (GBM)

Gradient Boosting is a widely used boosting algorithm in machine learning for regression and classification problems. Boosting is an ensemble learning technique that trains models sequentially, with each new model attempting to correct the previous one [18]. Gradient Boosting is an effective algorithm that transforms several weak learners into powerful ones by training each new model to minimize the loss function and other parameters like MSE or cross-entropy of the previous model through gradient descent.

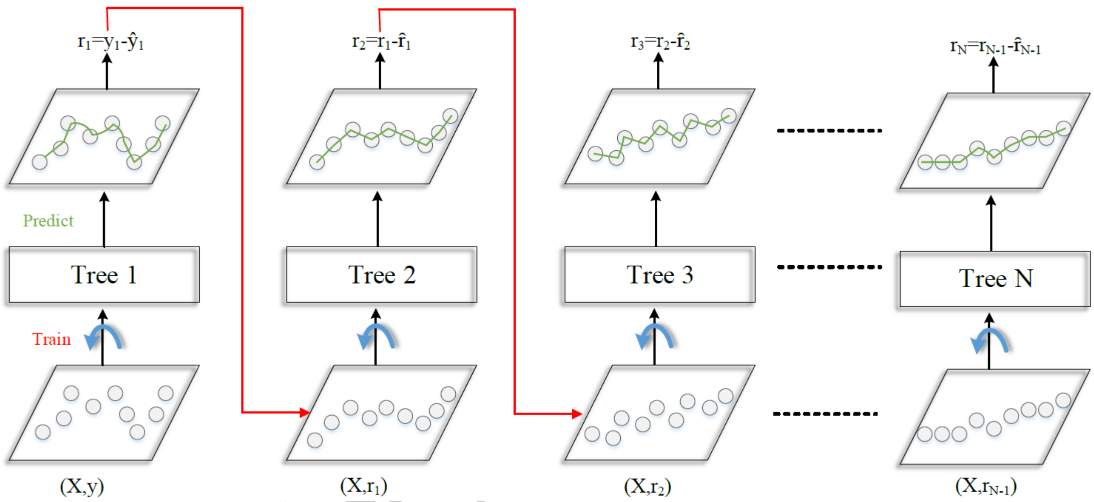


Figure.2 Structure of Gradient Boosting Machine

Fig.2 shows that the algorithm consists of an ensemble of M trees, with each tree being trained using the X feature matrix and the y label matrix. In each iteration, the previous tree's residual errors are used to train the next tree. The procedure is repeated until all M trees have been trained. Shrinkage is an important variable used in this technique, where the prediction of all trees in the ensemble is reduced after being multiplied by the learning rate (η), which varies between 0 to 1. There is a trade-off between η and the total number of estimators, where a decrease in the learning rate must be balanced by increasing the total number of estimators to achieve a specific level of model accuracy [19].

Once all the trees in the ensemble have been trained, predictions can be made. Gradient Boosting is a powerful algorithm and has been successfully used in many real-world applications, including finance and healthcare. However, it can be computationally

© PUNEL ADIA

expensive and requires careful tuning of hyperparameters to achieve optimal performance [20]. Overall, Gradient Boosting is a useful algorithm that can significantly improve the accuracy of machine learning models. Every tree forecasts a label, and the Eq. (3.5) yields the final prediction.

$$y_{pred} = y_1 + (\eta * r_1) + (\eta * r_2) + ... + (\eta * r_n) \tag{3.5}$$

3.5 Support Vector Machine (SVM)

SVM is a powerful machine learning algorithm used for both classification and regression problems. The goal of SVM is to find a hyperplane that separates data points of different classes with the largest margin, where the margin is the distance between the hyperplane and the closest data points from each class [21].

The SVM algorithm can be formulated mathematically as follows:

Given a set of labelled training data $D = \{(x_1, y_1), \ldots, (x_n, y_n)\}$., where $x_i \in R^d$ and $y_i \in \{-1, 1\}$, the goal of SVM is to find the hyperplane with the largest margin that separates the positive and negative classes [22]. This can be achieved by solving the following optimization problem:

$$\begin{aligned} \min_{\mathbf{w}, b, \xi} \quad & \frac{1}{2} W^T W + C \sum_{i=1}^{n} \xi_i \\ \textit{subject to} \quad & y_i (\mathbf{w}^T \phi(x_i) + b) \geq 1 - \xi_i, \\ & \xi_i \geq 0 \quad \forall i = 1, \ldots, n \end{aligned} \tag{3.6}$$

Where $\mathbf{w}$ is the weight vector of the hyperplane, b is the bias term, $\phi(x_i)$ is a nonlinear mapping function, ξ_i are slack variables that allow for misclassification, and C is a hyperparameter that controls the tradeoff between maximizing the margin and minimizing the classification error.

The decision function of SVM is given by:

$$f(x) = \text{sign}(\mathbf{w}^T \phi(x) + b) \tag{3.7}$$

Where sign returns the sign of the input.

3.6 Artificial Neural Network (ANN)

ANNs mimic the structure and function of the human brain and are used for tasks like classification, regression, and unsupervised learning. ANNs are composed of

© PUNEET PAHADIA

interconnected neurons, which receive input signals, process them, and produce output signals. A network of neurons is created, where the output of one neuron can be used as input to others. ANNs typically have an input layer, one or more hidden layers, and an output layer, with each layer containing multiple neurons that perform calculations on the input signals and produce output signals.

The output equation of ANN is given as:

$$y = f(\sum_{i=1}^{n} w_i x_i + b) \tag{3.8}$$

where y is the output of the neuron, f is the activation function, w_i are the weights of the connections between the neuron and its inputs, x_i are the inputs to the neuron, b is the bias term, and n is the number of inputs.

There are several activation functions commonly used in ANNs, including the sigmoid function, the hyperbolic tangent function, and the rectified linear unit function. The choice of activation function depends on the specific task and the architecture of the network.

3.7 Ensembling

3.7.1 Single Prediction Model

Formerly, the practise of utilising machine learning to forecast building energy consumption has involved the utilisation of a solitary learning algorithm and the training of a singular model throughout the modelling procedure. This traditional method is referred to as the 'single' prediction method. During the past two decades, numerous ML algorithms, such as MLR [23], ANN [24], and decision tree [6], and SVR [25], were proposed for prediction of building energy consumption and have developed promising results. Nonetheless, one of the most significant drawbacks of the single prediction method involves the instability problem within each model. Models such as decision trees and ANN are unstable models that may introduce substantial shifts in the final value as a result of small alterations to the data being supplied [26]. This instability problem could prevent the application of these models for real scenarios, as a few energy consumption measurements heavily depend on the accuracy of the forecast; for instance, an unstable model could result in a high error ratio for building energy consumption. EL has been proposed by researchers as a technique to tackle regression

© PUNEET PAHADIA

and classification challenges [27].This approach aims to overcome the issue of instability and enhance the accuracy of predictions.

3.7.2 Ensembled Prediction Model

Ensemble Model (EM) for forecasting are comprised of a collection of separately trained models, such as decision trees and neural networks, whose outputs are merged to generate a single forecast [27]. By leveraging model mutual benefit, EMs for forecasting can provide more reliable and precise predictions compared to single forecasting models [28]. In accordance to foundational model development strategy, ensemble prediction methods can be further divided into two main groups: homogeneous and heterogeneous ensemble. Heterogeneous ensemble, generates its foundational models by applying identical training data to distinct models or the same models with various parameter configurations. Homogeneous EM builds its foundational models by re-sampling the training data and applying it to the same models with identical parameter settings. Heterogeneous EM boosts forecasting by leveraging the complementarity between a varieties of ML models, whereas homogeneous EM is the same as an optimisation process that boosts the performance of a particular model by training it repeatedly with diverse datasets along with integrating its forecasts. The ensemble forecast approach has drawn increased interest and has become a desired research topic in numerous disciplines, especially power system load forecasting [29], disease diagnosis [30] as well as data classification [31]. However, this new approach is not widely used in the field of building energy consumption prediction, and a handful of related studies did not begin until 2014, as will be discussed below.

A heterogeneous EMS was introduced in study, utilizing data mining techniques to forecast the peak demand and energy consumption of a commercial building [28]. The suggested EM consisted of eight separate foundation models trained with distinct forecast algorithms. Genetic Algorithm (GA) has been used for combining the predictions of every single base model and generate the EM's forecast. Preliminary results indicated that the suggested ensemble framework provided more accurate forecasts than the standard single model approach. The utilisation of ensemble learning was implemented through the amalgamation of diverse artificial intelligence models to ascertain the ventilation and energy usage of residential properties in the developmental stage [32]. Furthermore, the study incorporated a total of twelve distinct building types,

© PUNEET PAHADIA

which were subjected to simulation via energy modelling software. Also, six artificial intelligence models were employed as foundational predictors. A set of eight building characteristics were employed for the purpose of forecasting ventilation and heating requirements. According to the research findings, the EM was appropriate for predicting heating and cooling demands. Research demonstrated viability of employing EMS's to facilitate the initial design of energy-efficient structures

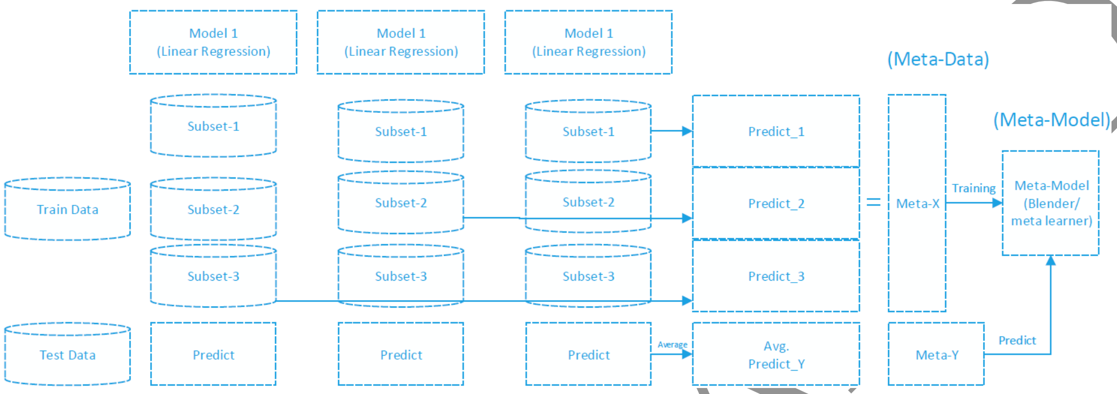


Figure.3 Model of Stacking Ensemble

Stacking is one approach to ensemble learning where multiple models as shown in Fig..3 are trained and their outputs are combined using a meta-model to make the final prediction. In stacking, the meta-model learns how to combine the predictions of the base models to make the best possible prediction. Stacking, also known as stacked generalisation, refers to the process of instructing a ML algorithm to integrate the forecasts of multiple other ML algorithms.

The remaining algorithms are trained utilising the obtainable data, followed by the training of a combiner algorithm to generate a conclusive prediction by incorporating every prediction of the remaining algorithms as supplementary inputs.

Where $\hat{y}_i$ is the predicted output of the i^{th} base model, and f is the function used to combine the outputs.

One common approach for combining the outputs of the base models is to use a linear regression model as the meta-model. In this case, the equation for the meta-model can be written as:

$$\hat{Y}_{stacked} = \beta_0 + \beta_1\hat{y}_1 + \beta_2\hat{y}_2 + \ldots\ldots + \beta_n\hat{y}_n \tag{3.9}$$

Where $\beta_0, \beta_1, \beta_2, \ldots, \beta_n$ are the parameters learned by the meta-model

© PUNEET PAHUJA

The utilisation of stacking in ensemble learning has been demonstrated in Figure 4 to enhance the efficacy of machine learning models across diverse domains such as image recognition, text classification, and fraud detection. However, it is important to note that stacking can also increase the risk of overfitting, and careful tuning of the models and meta-model is necessary to prevent this.

The algorithm for stacking can be described in Algorithm 1.

Algorithm 1 Stacking

Input: Training data $D = \{x_i, y_i\}_{i=1}^{m}$
Output: ensemble regressor H

Step 1: learn base-level regressors
  for $t = 1$ **to** T **do**
  | learn h_t based on D
end
Step 2: construct new data set of predictions
  for $i = 1$ **to** m **do**
  | $D_h = \{x'_i, y_i\}$, where $x'_i = \{h_1(x_i), ..., h_T(x_i)\}$
end
Step 3: learn a meta-regressor
  learn H based on D_h
  return H

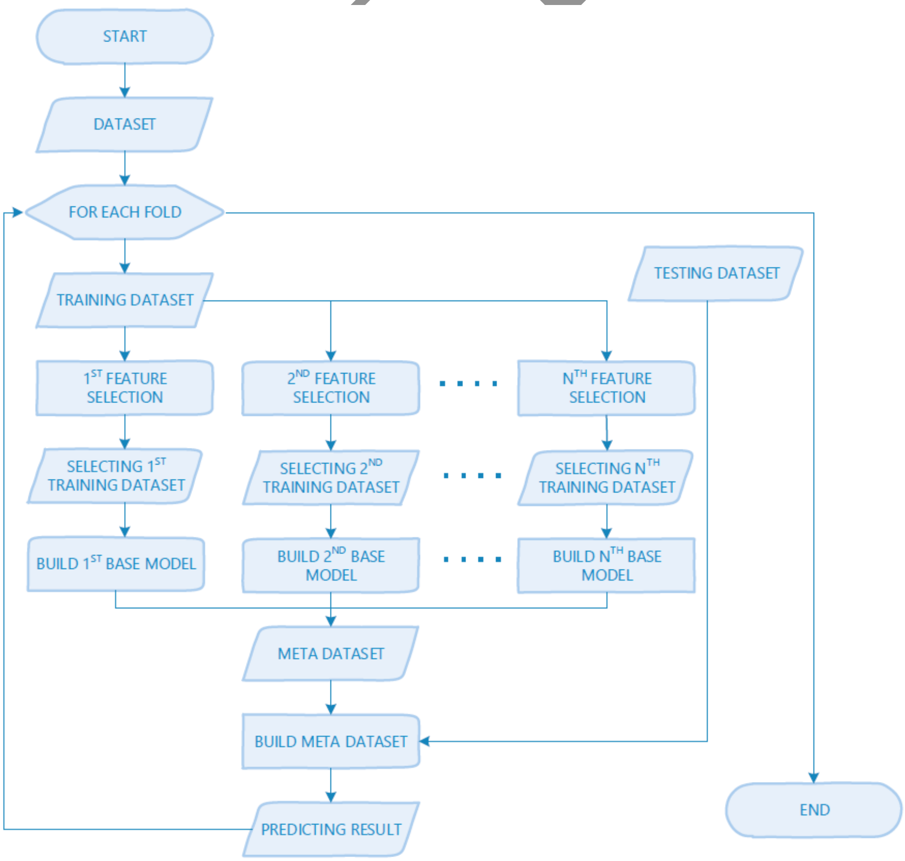


Figure. 4 Structure of Stacking Ensemble

Chapter: 4

Heap Based Optimization: A Novel Meta Heuristic Approach

4.1 Heap Based Optimization (HBO)

HBO is a technique used to optimize code by minimizing the number of memory allocations and deallocations. In this method, the programmer uses a heap data structure to allocate a large chunk of memory at once and then uses it to allocate smaller chunks of memory as needed. This approach reduces the overhead associated with memory management and can lead to significant performance improvements in certain applications.

In the context of a corporation, achieving a common objective often requires organizing individuals according to a corporate rank hierarchy (CRH). This idea inspires the proposal of a novel optimization algorithm that hierarchically organizes search agents based on their fitness [33]. HBO applies the heap structure to represent the CRH paradigm. Its mathematical framework is built on three foundations: relationships between employees and their immediate managers, interaction among co-worker's, and self-contribution of employees.

HBO is a unique meta-heuristic that draws inspiration from human social behaviour, which has advanced the study of swarm and evolutionary intelligence. Human behaviour-based techniques can be classified into sport-based techniques, algorithms based on human social interaction, algorithms based on colonization, and algorithms based on politics.

One form of human social interaction is observed in organizations where employees are organized in an organizational chart or CRH. Organizations structure their personnel according to the CRH because a group working towards a common objective cannot achieve it without specific organization. This results in structured arrangements within the organization, allowing employees to be effective in achieving organizational objectives. The organizational hierarchy takes the form of a tree-like framework. The apex of the organisational hierarchy is held by the manager, who assumes a role analogous to the root of a tree. The remaining personnel are arranged in a hierarchical structure resembling parent-child nodes. In this context, it can be observed that every superior node denotes a manager or head, while its descendants are regarded as

© PUNEET PAHADIA

subordinates. Each subordinate interacts with and follows the directives of their direct supervisor (the primary node). The ultimate objective of this hierarchy is to effectively carry out business-related activities through interpersonal connection [34]. Fig. 5 provides an example of CRH for illustration purposes. In the diagram, the CEO has been designated as the primary node and serves as the leader. COO and CFO report directly to the CEO, but are in charge of various departments. Colleagues are those who are on the same level in the tree.

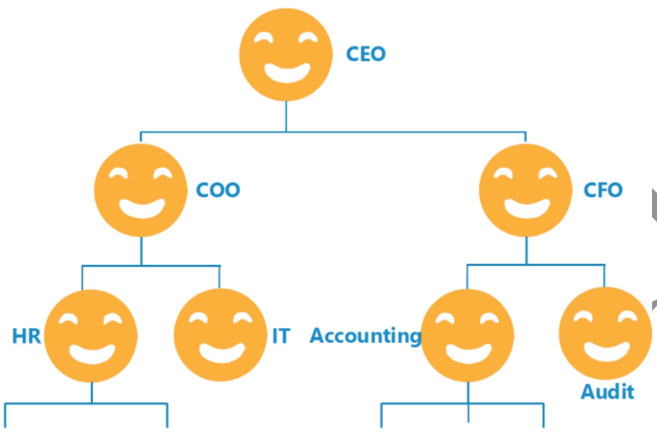


Figure.5 Structure of Corporate Ranking Hierarchy

4.1.1 Introduction to Heap-Based Optimization

The utilisation of heap-based optimisation is a viable approach to enhance the efficiency of software applications through the mitigation of the associated costs of memory allocation and deallocation. In this method, the programmer allocates a large chunk of memory using the heap data structure and then uses it to allocate smaller chunks of memory as needed. This approach is more efficient than traditional memory allocation techniques that allocate and deallocate memory on a per-request basis.

Heap-based optimization is particularly useful in applications that require frequent memory allocation and deallocation, such as real-time systems, multimedia applications, and scientific simulations [33]. By reducing the number of memory allocation and deallocation operations, heap-based optimization can improve the overall performance of these applications.

4.1.2 Advantages of Heap-Based Optimization

© PUNEET KALIA

Heap-based optimization offers advantages over traditional memory allocation techniques. It reduces the overhead associated with memory allocation and deallocation. The programmer allocates a large chunk of memory at once, eliminating the need for many small memory operations.

Heap-based optimization also reduces the likelihood of memory fragmentation. Small chunks of memory are allocated and deallocated repeatedly in traditional techniques, leading to fragmented memory. With heap-based optimization, large chunks of memory are allocated, reducing this risk.

Furthermore, heap-based optimization improves the performance of applications requiring frequent memory operations. By minimizing memory operations, it enhances the speed and responsiveness of these applications.

4.1.3 Limitations of Heap-Based Optimization

Despite its advantages, heap-based optimization may not be appropriate for all application types, particularly those with specific memory usage patterns or operating within limited memory environments.

Implementing heap-based optimization correctly can be challenging, as it requires meticulous management of memory allocation and deallocation to prevent issues such as memory leaks. Furthermore, the introduction of heap-based optimization can increase code complexity and potentially create difficulties in code comprehension, potentially leading to maintenance challenges in the future.

4.1.4 Modelling of the Corporate Rank Hierarchy

A heap is a type of non-linear data structure that exhibits a tree-like structure and is defined by a set of specific properties. In a min-heap, each parent node's key is either below or equal to the keys of its children, while in a max-heap, each parent node's key is either higher or equal to the keys of its children. This ensures that the minimum or maximum value is located at the root of the heap. Additionally, a heap is considered a fully developed tree, meeting the condition of being complete. This means that all levels, except the last one, are filled with nodes, and in the lowest level, the nodes are positioned as far to the left as possible. These characteristics make heaps efficient and organized structures for storing and retrieving data in various applications.

© PUNEET PAHADIA

The entire CRH is regarded as a community. During the implementation phase, it can be observed that a search agent is equivalent to a heap node. The predominant determinant of a node's position within the heap is considered to be the fitness of the search agent, whereas the indexing of the search agent within the population is deemed to represent the node's value within the heap. Fig. 6 depicts the CRH model utilising a heap structure, wherein the variable x_i represents the i^{th} search agent within the population. The curvature observed in the objective space is indicative of the topography of a hypothesised objective function, while the search agents are positioned on the landscape of fitness based on their respective levels of fitness. A heap is built based on the fitness of every agent in the search as a primary factor. As illustrated in Fig.. 6, the positioning of nodes within the heap is determined by their respective levels of fitness. As an illustration, it can be observed that x_4 represents the optimal solution within the given population, and as such, it also serves as the root node of the heap. It is noteworthy that the utilisation of a min-heap is necessary for the purpose of minimising, while a max-heap is required for the purpose of maximising.

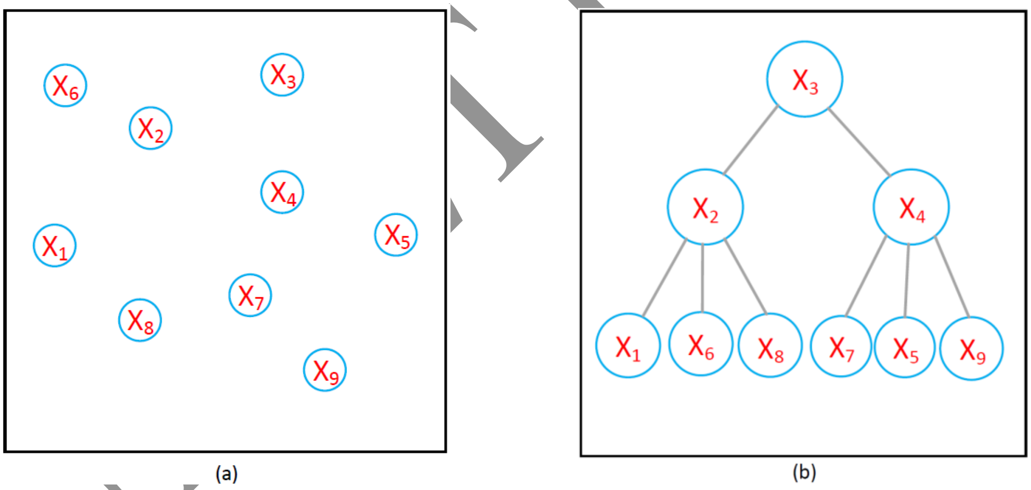


Figure.6 Structure of Corporate Ranking Hierarchy

4.1.5 Mathematical Modelling of the Interaction with Immediate Boss

In a centralised organisational structure, laws and standards are implemented from the highest level to lowest level, and subordinates obey their immediate boss. This can be numerically depicted by modifying the search for the agent's location as given in (2.10)

The symbol "where" denotes a stochastic variable, while the symbol "represents" signifies the most favourable outcome. The symbols "and" and "component vector

© PUNIT PAHADIA

number" denote the contemporaneous cohort and the numerical value of a vector component, respectively. The summation of the quantity denoting the number of generations and a randomly generated integer is equivalent to the quotient of a given integer and 25.

$$X_i^k(t+1) = B^k + \left| \frac{\left(t \bmod \frac{T}{C} \right)}{\frac{T}{4C}} \right| (2r-1) \left| B^k - X_i^k(t) \right| \tag{4.1}$$

4.1.6 Mathematical Modelling of the Interaction Between Colleagues

At this juncture, it should be expected that their co-workers are discussing. Each X_i updates its position using an arbitrarily generated solution S_r , as shown in Eq. (10).

$$X_i^k(t+1) = \begin{cases} S_r^k + \gamma(2r-1)^k \left| S_r^k - X_i^k(t) \right|, f(S_r) < f(X_i) \\ X_i^k(t) + \gamma(2r-1)^k \left| S_r^k - X_i^k(t) \right|, f(S_r) \geq f(X_i) \end{cases} \tag{4.2}$$

4.1.7 Mathematical Modelling of the Employee Self-Contribution

Using this formula, the prior value of solution is maintained.

$$X_i^k(t+1) = X_i^k(t) \tag{4.3}$$

4.1.8 Overall Position Update

Various mathematical equations can be employed to modify the current solution. Additionally, these formulas can be combined using probability calculations, as elucidated [34].

The optimisation of the balance in exploration and exploitation has been improved. The task is executed through the utilisation of the roulette wheel, and as a result, the updating technique is formulated in the way that follows:

$$X_i^k(t+1) = \begin{cases} X_i^k, p \leq p_1 \\ B^k + y\lambda^k \left| B^k - X_i^k(t) \right|, p > p_1 \,\&\, p \leq p_2 \\ S_r^k + \gamma\lambda^k \left| S_r^k - X_i^k(t) \right|, p > p_1 \,\&\, p \leq p_3 \,\&\, f(S_r) < f(X_i) \\ X_i^k(t) + \gamma\lambda^k \left| S_r^k - X_i^k(t) \right|, p > p_1 \,\&\, p \leq p_3 \,\&\, f(S_r) < f(X_i) \end{cases} \tag{4.4}$$

© PUNEET PAHADIA

In Eq. (12), p is a random number between 0 and 1, and p_l , l is the probability, which is denoted by:

$$p_1 = 1 - \frac{t}{T},\ p_2 = p_1 + \frac{1 - p_1}{2},\ p_3 = p_1 + \frac{1 - p_1}{2} \tag{4.5}$$

In the end, the HBO implementation starts by randomly creating a collection of N solutions, the process involves the computation of the fitness value of each element, followed by the representation of the heap as a d-array tree, that stands for CRH as provided in Eq. (14), represents the heap.

$$Parent_i = \left\lceil \frac{i+1}{d} \right\rceil \tag{4.6}$$

The aforementioned equation denotes the position of the progenitor node i, which possesses a total of three offspring. The mathematical formula for denoting the j^{th} child at i^{th} node is as follows:

$$child\left(i, j\right) = d * i - d + j + 1 \tag{4.7}$$

Furthermore, depiction of depth of i^{th} node is provided as:

$$depth_i = \left\lceil log_d\left(d * i - i + 1\right)\right\rceil - 1 \tag{4.8}$$

Subsequently, the subsequent procedure involves the generation of an integer number from $d-1$ for every colleague i. Subsequently, the $Heapify\_Up(i)$ algorithm endeavours to locate the most advantageous position for node i within the heap. The heap is initialised, where its key and value are utilised to maintain the indices of agents and their respective fitness values [35].

$$\left[\frac{d * d^{depth_i - 1}}{d-1} + 1, \frac{d * d^{depth_i - 1}}{d-1}\right] \tag{4.9}$$

4.2 Development of HBO Along with Assessment

In this section, every essential HBO stages and its implementation-related aspects are presented. In addition, it proves that integrating ensemble into the structure of HBO has no cumulative effect on the method's time.

© PUNEET PAHADIA

4.2.1 Algorithm for Heap-Based Optimization

Heap-based optimization involves several algorithms to efficiently manage and optimize memory allocation and deallocation. Here are three key algorithms used in heap-based optimization:

Algorithm 1: *Heapify_Up*(i)

This algorithm is responsible for maintaining the heap property by adjusting the position of a node with index i upwards in the heap. It compares the key of the node with its parent's key and swaps them if necessary. The process continues recursively until the node satisfies the heap property.

Algorithm 2 Heapify_Up(i)

Input: i(the index of the node to be heapify)
Assuming that the rest of the nodes fulfill the heap property
while $i \neq$ *root and heap[i].key* $<$ *heap[parent(i)].key* **do**
  swap(heap[i],heap[parent(i)])
  i ← parent(i)
end

Algorithm 2: *Build Heap*(P, N)

The Build_Heap algorithm constructs a heap from a given array of elements P of size N. It starts from the last non-leaf node in the array and applies Heapify_Up on each node to ensure that the entire array satisfies the heap property. This algorithm efficiently builds a heap in linear time complexity.

Algorithm 3 Build_Heap(P,N)

Input: P (population of search agents), N (population size)
for *each index i from 1 to N* **do**
  Create a new heap node and assign it to heap[i]
  Set heap[i].value to i
  Set heap[i].key to $f(x_i)$
  Call Heapify_Up(i) to restore the heap property
end

Algorithm 3: HBO *Main_Body*()

The HBO_Main_Body algorithm represents the main body of the heap-based optimization process. It utilizes the heap data structure and the previously mentioned algorithms to optimize memory allocation and deallocation. It manages the allocation of a large chunk of memory using Build_Heap, and subsequently allocates smaller

© PUNEET PATADIA

chunks from within that memory space based on the heap property. The algorithm also handles deallocation by efficiently releasing memory and maintaining the heap structure.

Algorithm 4 HBO_Main_Body()

for $i \leftarrow 1$ *to* T **do**
- Compute γ by using Eq.(3)
- Compute p_1 by using Eq.(7)
- Compute p_2 by using Eq.(8)
- heap[i].key $\leftarrow f(x_i)$
- **for** *index* $\leftarrow N$ *down to 2* **do**
  - valueI $\leftarrow$ heap[index].value
  - valueBI $\leftarrow$ heap[parent(index)].value
  - valueCI $\leftarrow$ heap[colleague(index)].value
  - $\vec{B} \leftarrow \vec{x_{bi}}$
  - $\vec{S} \leftarrow \vec{x_{ci}}$
  - **for** $k \leftarrow 1$ *to* D **do**
    - p $\leftarrow$ rand()
    - $x^k_{temp} \leftarrow$ update x^k_i by using Eq.(10)
  - **end**
  - **if** $f(\vec{x}_{temp}) < f(\vec{x}_i(t))$ **then**
    - $\vec{x}_i(t+1) \leftarrow \vec{x}_i(t)$
  - **end**
  - Heapify_Up(index)
- **end**

end

return $x_{heap[index].value}$

These algorithms work together to facilitate efficient memory management and optimization in heap-based optimization techniques. By leveraging the heap data structure and these algorithms, heap-based optimization minimizes memory overhead and fragmentation, leading to improved performance in relevant applications.

© PUNEET KHADIA

Chapter: 5

Proposed Model and Case Study

5.1 Proposed Optimized Stacked EM

This section outlines the procedural steps involved in the implementation of the HBO-Ensemble model. Fig. 7 illustrates the primary workflow. The primary concept underlying the HBO-Ensemble involves the optimisation of Ensemble parameters through the utilisation of the HBO algorithm. This procedure results in an improvement in the predictive precision of the conventional Ensemble. The initial step involves partitioning the input data into two distinct sets, namely the training and testing sets, with a ratio of 70% and 30%, correspondingly. Subsequently, during the training phase, the suggested approach initiates a collection of solutions, wherein each solution corresponds to the parameter of the EM. Subsequently, the fitness function may be utilised. The fitness function chosen for this study is the Root Mean Squared Error (RMSE), which can be mathematically expressed as:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{N}(a_i - p_i)^2} \tag{5.1}$$

Here, a_i represents the actual or real value, whereas p_i signifies the predicted value. Moreover, it is feasible to revise the solutions to align with the search algorithms utilised by HBO, as previously delineated. The determination of the optimal solution is contingent upon the attainment of the minimum root mean square error (RMSE) value. The process of iteratively revising solutions is persisted until the designated termination criterion is met. Additionally, the effectiveness of the HBO-Ensemble model is evaluated by allocating 30% of the dataset as the testing set and assessing the prediction output using suitable metrics.

5.2 Model Training

Due to the usage of high-level libraries, the model training procedure is rather straightforward, Keras, numpy, mlxtend, seaborn, matplotlib and Scikit Learn, and the complete dataset may be stored in the computer's main memory. The ensemble technique chosen for this study is Stacked Generalisation, which is characterised by its simplicity of implementation, high performance, and clear elucidation of each algorithm's contribution to the ultimate model. RR constructs a linear model for level 0

© PUNEET PAHADIA

algorithms, allowing for straightforward interpretation but poorly fit. While RF, which makes use of all features during training and has strong generalizability, ERT may be trained quickly and with less variation. In contrast to RF, where overfitting is less likely, Gradient Tree Boosting may often result in more accurate predictions. SVM have a strong generalisation power and are able to handle enormous datasets effectively due to its capacity to seek for maximum margin separation. Although SVM is less susceptible to overfitting, ANNs are capable of modelling continuous functions. Each of the selected algorithms has its own set of advantages and disadvantages.

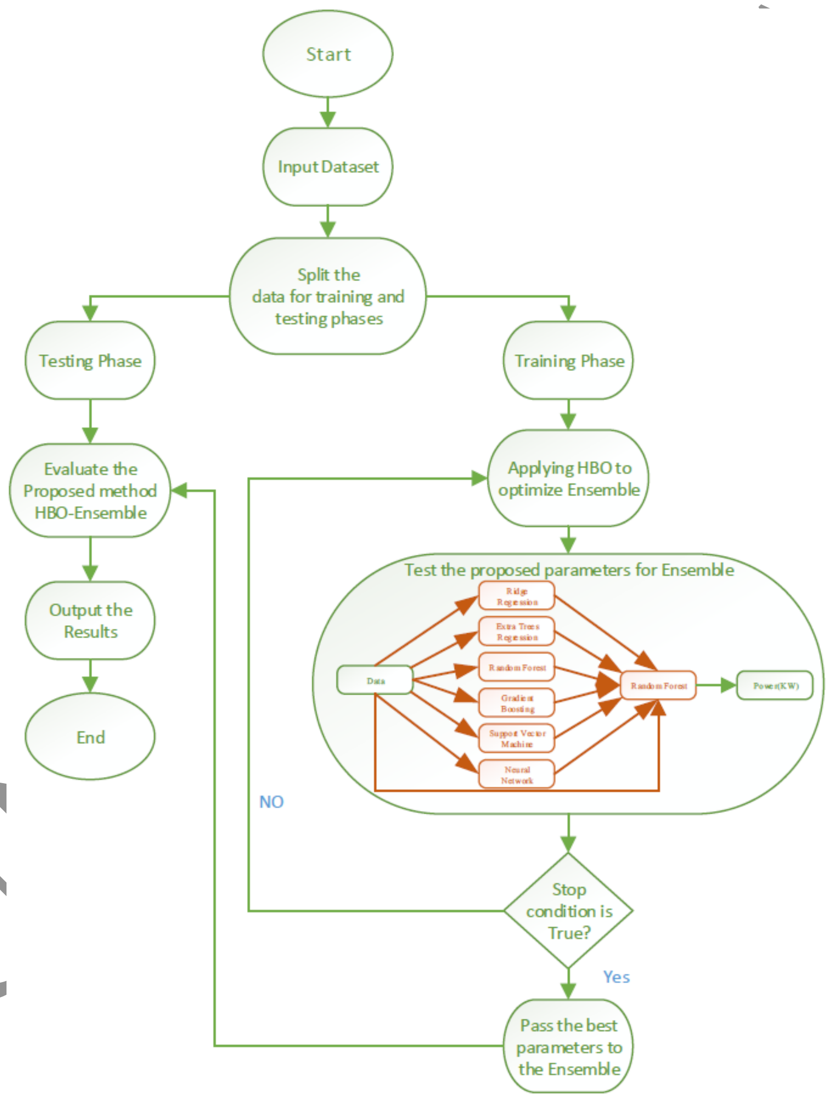


Figure.7 Structure of Corporate Ranking Hierarchy

The algorithms used in this study are discussed in detail in [35]–[37]. The level 1 algorithm was selected based on the precision of the level 0 algorithms, with RF achieving the highest precision (as discussed in section results) and having good generalization capability. Experiments showed that feeding the level 1 algorithm with

© PUTRA MALAYSIA

information other than the level 0's output improved its performance, yielded better results for this particular problem, as depicted in Fig.8.

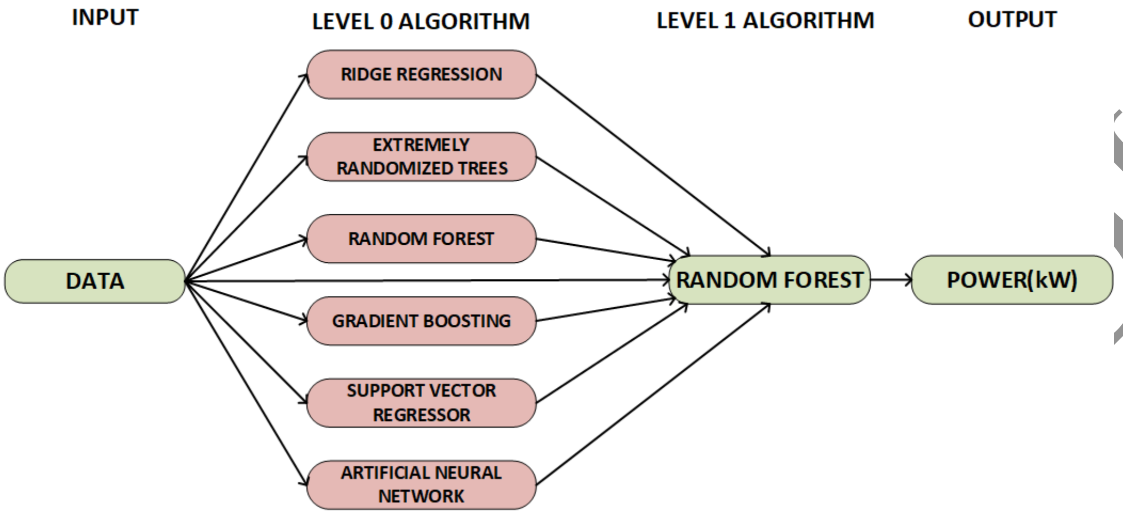


Figure.8 Structure of Ensembled Model

Each ML model has multiple hyperparameters that need to be optimized to obtain better results. Some of these hyperparameters were selected by testing different values and choosing the one that resulted in the best performance, while others were set to their default values as they had minimal impact on the final accuracy. Occam's Razor was used to choose the simplest model that could account for the observed data.

The code is designed to predict the energy consumption of a building, and the fit function is used to calculate the mean absolute percentage error (MAPE) of the predicted values.

The HBO algorithm is implemented using the OriginalHBO class from the mealpy library, with the fitness function defined as the fit function. The parameters to be optimized are the maximum depth of the random forest regressor, the number of estimators, the maximum number of samples, and the maximum number of features. The lower and upper bounds for the parameters are defined as [10, 100, 0.1, 0.1] and [50, 400, 1, 1], respectively, and the optimization is performed for one epoch with a population size of 50.

The best position and best fitness are then printed to the console, and a new random forest regressor is trained using the optimized parameters. The base models used in the ensemble are model1, model2, model3, model4, model5, and Ridge (α =1.0), while the meta model is a random forest regressor with the optimized parameters.

© PUNEET PAL SINGH

The stacked model is then fit to the training data, and the predicted values are calculated using the predict function on the test data. Subsequently, the anticipated values undergo a reversal transformation to revert to their initial scale by means of an inverse transform technique. The RMSE and MAPE are then computed utilising the mean_absolute_percentage_error and mean_square_error functions, respectively, from sklearn library.

Finally, the percentual error is calculated using the mean_absolute_error function from the sklearn library, and the results are printed to the console. The heap-based optimization technique used in this study helps to optimize the parameters of the random forest regressor, resulting in improved accuracy of the predicted energy consumption values.

5.3 Hyperparameters Optimization Tables

The Table 1 provides the hyperparameters for different models. Each row represents a specific model, and the corresponding hyperparameters are listed in the "Parameter" column. For each model, the respective hyperparameters are mentioned, such as maximum depth, number of estimators, bootstrap value, loss function, kernel type, regularization parameter, batch size, validation split, and callback functions.
These hyperparameters were used to configure and fine-tune the models for specific tasks or datasets.

Table 1-Hyperparameters

Model	Parameter
ERT	max_depth = 25, n_estimators = 400, bootstrap = True, max_samples = 0.7
RF	max_depth = 20, n_estimators = 400, bootstrap = 0.9
GBoosting	max_depth = 8, n_estimators = 400, Loss = squared_error
SVM	kernel = rbf, C= 900, epsilon = 1, gamma = scale, cache_size = 100
ANN	Epoch = 200, Verbose = 1, batch_size = 32, validation_split = 0.1, callbacks = callback_cp & callback_es
Stacked Ensembling	max_depth = 25, n_estimators = 400, max_samples = 0.6, max_features = 0.6
HBO+Ensemble	max_depth = 15, n_estimators = 290, max_samples = 0.1232, max_features = 0.7579

5.4 Datasets Description

The model's data is sourced from PGEN project database, which specifically selects

© PREET PAHADIA

data from commercial buildings from February 1, 2018, to December 13, 2019. As stated in reference [38], database is accountable for holding records to maintain database reliability, accessibility and truthfulness. The PGEN initiative employs a relational database management system that organises data in a structured manner, necessitating the use of SQL. The segregation of climatic and electrical data necessitates their individual retrieval and pre-processing prior to amalgamation.

Cross-Validation was used to combine the two ML approaches. This method randomly divides the data into two i.e. training group and test group. The training set is utilised to construct a functional model, while the test set is employed to evaluate the precision of the model [35]. It allows for an evaluation of the model's prediction and generalization capacity, or its ability to forecast future power values. 70% of the dataset was kept for training, while 30% was put aside for testing. This meant that 23582 data-points were utilized for training the model, rest were used for testing the model. Keras and Scikit libraries along with Python were utilized for managing the dataset and implementation of ML techniques

5.5 Data Pre-processing and integration

Climate database was used to extract wind speed (m/s), precipitation (mm), atmospheric pressure (hPa), and ambient temperature (ºC) with hourly interval. Fig. 9 depicts the probability density estimation of said variables, which were computed utilising the Density Estimation technique.

In electrical database total active power (kW), voltage (KV) and power factor (dimensionless) were extracted at an interval of 1 minute. Fig.10 shows the probability density estimate of the above variables which is calculated using the density estimation method.

To combine both the climatic and electrical data, the climate-dependent features were interpolated at a 10-minute frequency using linear interpolation, during the process of reducing the sampling rate of electrical variables, they are adjusted to a uniform frequency. For the 10-minute timeframe, standard deviations for power factor and active power were computed, and the performance indicator was defined as load factor presented by,

$$Load\ Factor = \frac{Maximum\ Active\ Power(P_{max})}{Mean\ of\ all\ Powers(P_{mean})} \tag{5.2}$$

Following the integration of the two datasets, three more features were extracted from

© PUNEET PAHADIA

timestamps which are month_of_the_year, hour and day_of_week. Noisy data which were measured due to equipment failure were removed. The performance of the technique was improved by normalizing the dataset, the scaled features exhibit a mean of zero and a standard deviation of one. However, for simplicity of understanding, only the output variable i.e. active power wasn't standardised.

Following the processes of data cleansing and integration, a total of 33,689 data points were retained. These data points consist of climatic and electrical measurements described across 12 variables, with a uniform interval of 10 minutes.

5.6 Outlier Detection and Removal

Outlier detection and removal is a crucial step in data analysis and it helps to identify and eliminate extreme values that may skew the analysis and results. Median-based outlier detection and removal is a popular method that involves replacing outliers with the median value of the remaining data. In this approach, the median value is used as a measure of central tendency instead of the mean, which can be affected by extreme values.

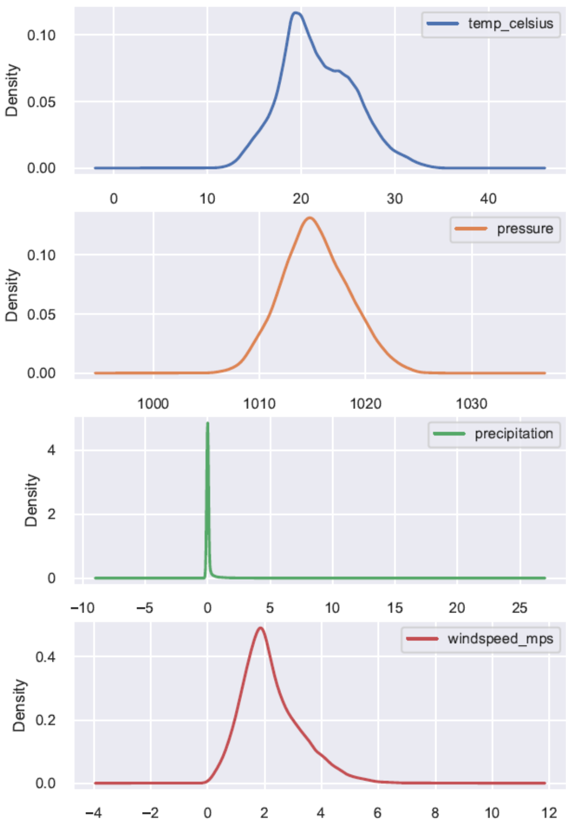


Figure 9 KDE of Climatic Variable

© PUNYA PAHADIA

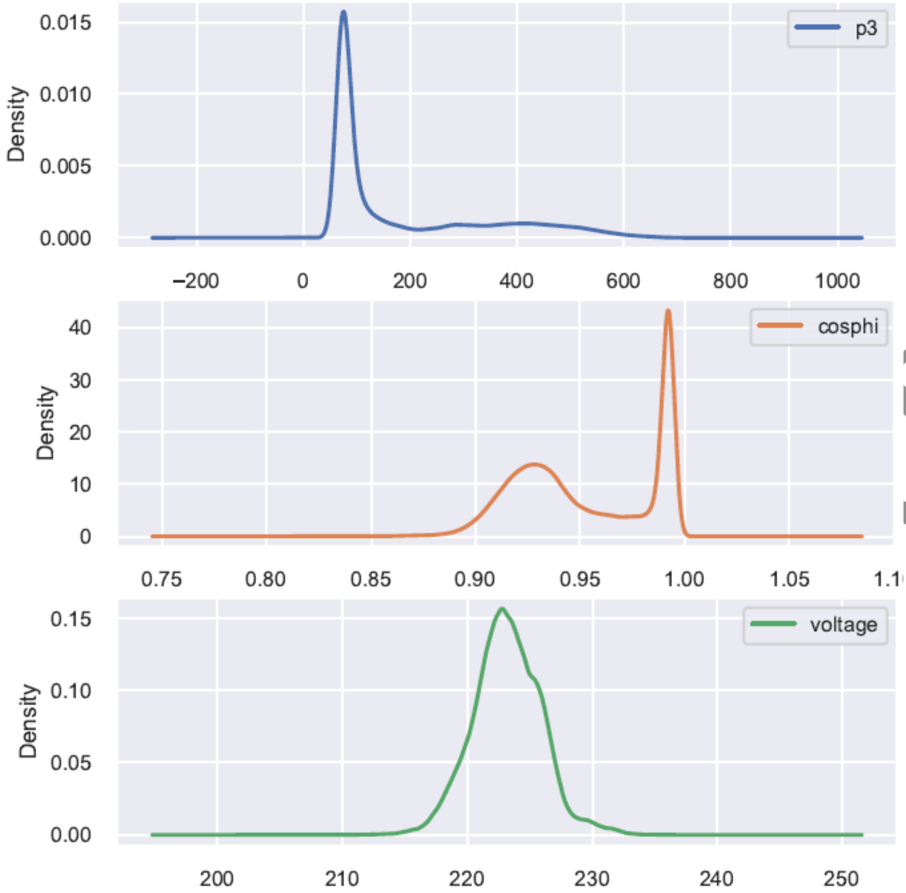


Figure 10 KDE of Electric Variable

Using median-based outlier detection and removal involves the following steps:

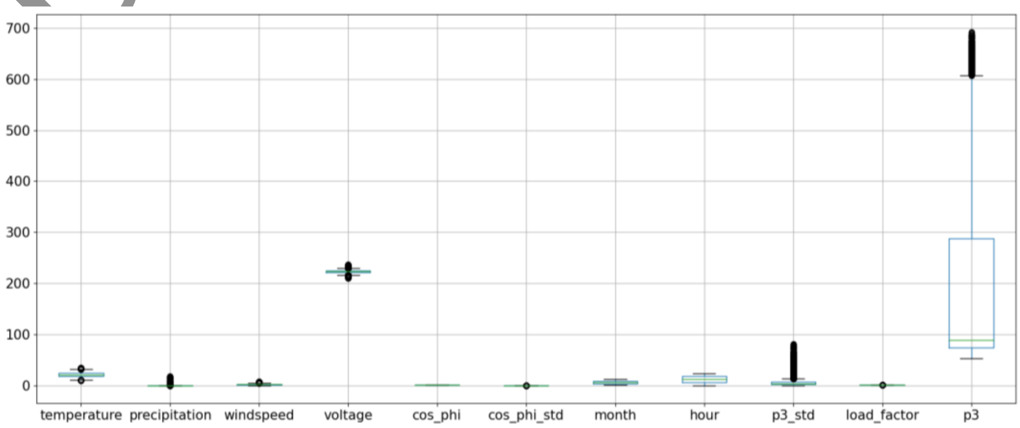
Step 1. Calculate the median of the dataset.

Step 2. Calculate the median absolute deviation (MAD) of the dataset. This is the median of the absolute differences between each data point and the median.

Step 3. Define a threshold value as a multiple of the MAD. (Typically, a threshold value of 2 or 3 is used)

Step 4. Identify data points that are outside the threshold value as outliers.

Step 5. Replace outliers with the median value of the remaining data as shown in Fig..12



© PUNEET INDIA

Figure 11 Outliers initial

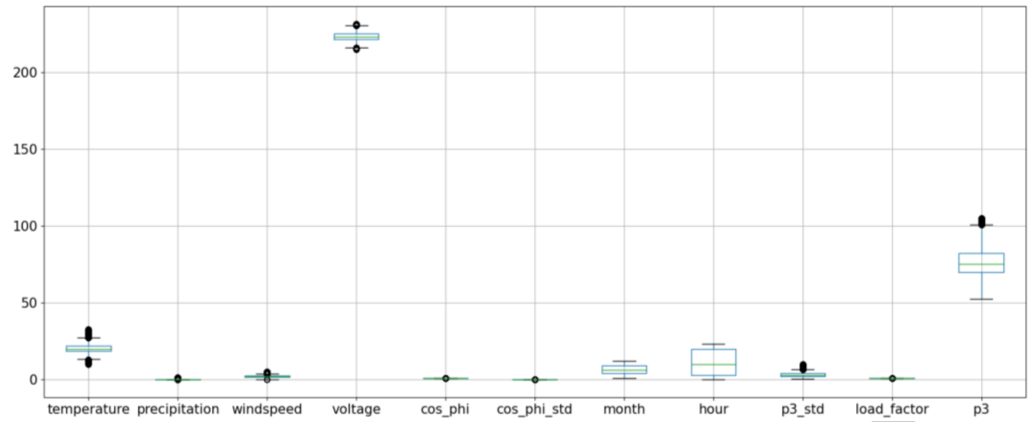


Figure 12 Outliers Final

This method is effective in removing outliers and improving the accuracy of the analysis. However, it is important to note that this method can also potentially remove genuine data points that may be important to the analysis. Therefore, it is important to carefully consider the threshold value used and examine the impact of outlier removal on the analysis.

5.7 Correlation Analysis

Correlation analysis is a statistical method that measures the strength and direction of the relationship between two variables. The correlation coefficient is a numerical measure that ranges from -1 to 1. A value of -1 denotes a complete negative correlation, while a value of 0 indicates no correlation, and a value of 1 denotes a complete positive correlation. In this analysis, the correlation between various variables with respect to $p3$ will be examined.

The Fig. 13 shows that $p3$ has a perfect positive correlation with itself (correlation coefficient of 1.0). This is expected since $p3$ is being compared to itself.

The variable with the highest positive correlation to $p3$ is cos_phi, with a correlation coefficient of 0.748. This indicates a strong positive relationship between $p3$ and cos_phi. A high cos_phi value indicates good power factor, which is indicative of efficient power utilization. Therefore, a high $p3$ value is likely to be associated with a high cos_phi value.

The next variable with a moderate positive correlation to $p3$ is *temperature*, with a correlation coefficient of 0.413. This suggests that $p3$ tends to increase with increasing

© PUNEET PAHADIA

temperature. The variable hour has a weak positive correlation with $p3$, with a correlation coefficient of 0.327, which suggests that $p3$ tends to increase during certain hours of the day.

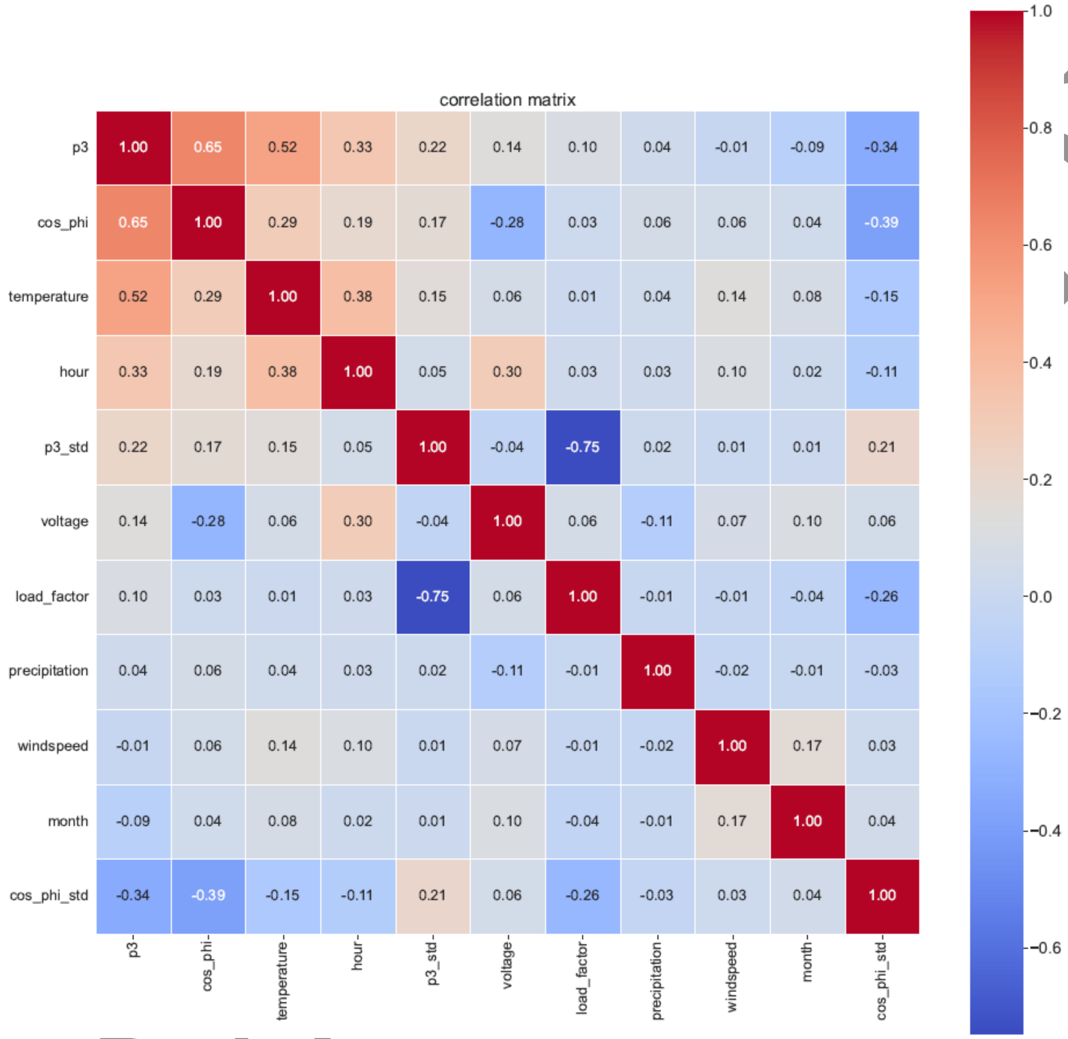


Figure 13 Correlation Plot

The Fig. 13 shows that $p3$ has a perfect positive correlation with itself (correlation coefficient of 1.0). This is expected since $p3$ is being compared to itself.

The variable with the highest positive correlation to $p3$ is cos_phi, with a correlation coefficient of 0.748. This indicates a strong positive relationship between $p3$ and cos_phi. A high cos_phi value indicates good power factor, which is indicative of efficient power utilization. Therefore, a high $p3$ value is likely to be associated with a high cos_phi value.

The next variable with a moderate positive correlation to $p3$ is *temperature*, with a correlation coefficient of 0.413. This suggests that $p3$ tends to increase with increasing *temperature*. The variable hour has a weak positive correlation with $p3$, with a correlation coefficient of 0.327, which suggests that $p3$ tends to increase during certain hours of the day.

On the other hand, the variable with the highest negative correlation to $p3$ is *pressure*, with a correlation coefficient of -0.320. This indicates that $p3$ tends to decrease as pressure increases. The variable cos_phi_std also has a moderate negative correlation to $p3$, with a correlation coefficient of -0.421, suggesting that $p3$ tends to decrease with increasing standard deviation of cos_phi.

The remaining variables (voltage, load_factor, day_of_ week, windspeed, month, and precipitation) have weak or no correlation with $p3$, with correlation coefficients ranging from -0.03 to 0.10.

The correlation analysis suggests that the most important variables associated with $p3$ are cos_phi, temperature, pressure, and cos_phi_std. These findings could be useful in predicting $p3$ values in a real-world setting, and could also inform decisions related to power utilization and energy efficiency.

5.8 Feature Selection

This method utilizes SelectKBest's chi-squared approach to find and eliminate outliers. Outliers were removed from a set of variables that included temperature, pressure, precipitation, windspeed, voltage, cos_phi_std, p3_std, load_factor, and $p3$. By using this method, extreme values in the variables that could have had a significant impact on the analysis were identified and removed.

The chi-squared method of SelectKBest is a feature selection technique that scores each variable based on its correlation with the target variable. The technique was used to score the variables and identify the most important ones for predicting the target variable (p3 in this case). Variables with the lowest scores were considered the least important and were removed as outliers.

The importance of performing outlier removal carefully and transparently is acknowledged. The criteria used to identify outliers were documented, and the

© PUNEET PAHADIA

reasoning behind these criteria was explained. Any changes in the distribution or variance of the variables after outlier removal were reported to ensure the validity of subsequent analyses.

Utilizing the chi-squared method of SelectKBest for outlier detection and removal proved to be a valuable step in the study. This technique improved the accuracy and reliability of the analysis and can be applied to various research studies to enhance the quality of the results.

© PUNEET PAHADIA

Chapter: 6

Results

6.1 Employed Evaluation Metrics

The accuracy of the model is measured using in four metrics: percentual error, root mean square error, mean absolute percentage error (MAPE) and mean absolute error (MAE).

Percentual error is a metric that measures the relative difference between the predicted and actual values [39]. The calculation involves determining the absolute difference between the predicted and actual values, followed by dividing the result by the actual value. The quotient is then multiplied by 100 to represent the outcome as a percentage. This metric is valuable in scenarios where the significance of the deviation between projected and observed values is crucial, irrespective of the orientation.

$$Percentual\ Error(\hat{y}) = \frac{MAE(\hat{y})}{Mean(\hat{y})} \tag{6.1}$$

RMSE is a metric that measures the root of the mean of the squared differences between the predicted and actual values. This metric is useful in situations where the direction of the difference between predicted and actual values is less important than the magnitude [40]. RMSE gives a higher weight to larger errors, and hence, it is more sensitive to outliers.

$$Root\ Mean\ Square\ Error(\hat{y}) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{6.2}$$

The MAPE is a quantitative measure utilised to assess the accuracy of a forecasting model by computing the percentage difference between the predicted and actual values. The metric is determined by computing the absolute discrepancy between the projected and factual values, which is subsequently divided by the factual value, and finally multiplied by 100 to represent it as a percentage. [41]. This metric is useful in situations where the direction and magnitude of the difference between predicted and actual values are both critical.

$$MAPE = \frac{1}{n} \times \sum \left| \frac{actual - forecast}{actual} \right| \tag{6.3}$$

© PUNEET PAHADIA

MAE is a metric that measures the mean of the absolute differences between the predicted and actual values [42]. It is a simpler metric than RMSE but gives equal weight to all errors, regardless of their magnitude.

$$Mean\ Absolute\ Error(\hat{y}) = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \qquad (6.4)$$

The choice of evaluation metrics depends on the specific problem and the goals of the analysis. Some metrics may be more appropriate than others in certain situations. For instance, RMSE is more sensitive to outliers and larger errors, while MAPE and MAE are more sensitive to small errors. Therefore, it is essential to choose the evaluation metrics that best suit the research problem and accurately reflect the accuracy of the model's predictions.

Here, y_i represents the i^{th} entry's actual output value, $\hat{y}_i$ is the algorithms forecasted value, and n represents the total number of cases in the testing set. MAE is straightforward to interpret and may be visually evaluated presented which can be calculated using (4.4). RMSE penalizes larger errors, hence, the error spread is measured denoted by (4.2). A measure of how far off from the mean a given variable is provided by its percentual error in (4.1).

In order to assess the model's capacity to generalise, the test group's data, which were not incorporated in the training set, were utilised. The resulting errors for each algorithm are presented in table 8. The ensemble model generated fewer errors than the most accurate individual model, demonstrating the effectiveness of this technique. By comparing the expected and actual power values, errors may be visually examined, where a perfect model would show a straight line.

Table 2-Error Comparison of Different Algorithms

Algorithm	MAPE	RMSE	Percentual
RR	0.058	5.76	5.82
ERT	0.029	3.20	3.02
RF	0.029	3.20	2.99
GTB	0.029	3.10	2.99
SVM	0.053	5.11	5.24
ANN	0.034	3.75	3.52
Ensemble	0.028	3.06	2.86
HBO-Ensemble	0.027	3.05	2.85

© PUNEET PAHADIA

The RR technique generates negative power values, and is not practical for this application. Additionally, its predictions become more inaccurate as the power values get lower or higher, indicating that the algorithm does not fit the data well.

6.2 Model Evaluation

One way to gain new insights from large datasets using ML models is through Data Mining. Analysing the linear coefficient in the Ridge Regression model is another method for determining whether the variables are inversely or directly connected with the output.

According to the findings, hour and the power factor are the two most important factors in the ensemble model, however this isn't the case for other individual models, showing that they all take slightly different approaches to prediction.

EMS can be used to generate alerts when there is abnormal energy consumption, which can help in detecting equipment breakdowns or abnormal circumstances. The system can emit an alert via various means, such as email, text message, or the web platform, when the gap between actual and forecasted consumption surpasses a given level. One more potential application is the automation of energy efficiency initiative evaluations, such as replacing less efficient equipment or new energy policies being adopted. By correlating actual consumption to that forecasted by a trained model on data, we may determine whether the initiative was successful. It is possible to calculate the amount of energy saved. This form of evaluation is often complex and calls for expert auditors. Yet, an automated technique can be fully incorporated into the EMS, allowing for continuous and dependable operation.

In addition to its other uses, the model is capable of predicting the energy bill by keeping certain variables constant, such as power factor or consumption standard deviation, which are difficult to predict. Furthermore, the model may be changed to use only known input variables, such as hour and temperature, needing retraining.

The study discusses the application of numerical modelling using ML. One potential application of these models is to mine data acquired by an EMS in order to gain new insight on the elements that contribute to energy usage. Another use is to generate warnings when usage exceeds the expected level given current conditions, possible use in identifying abnormal or failing machinery conditions. Additionally, Models can be

© PUNEET PAHADIA

used to evaluate the efficiency of energy-saving measures by comparing actual usage with forecast made using historical data. Finally, the models can be used to forecast energy consumption for estimation of energy bills. Overall, ML can be a valuable tool in helping organizations better manage their energy consumption and costs.

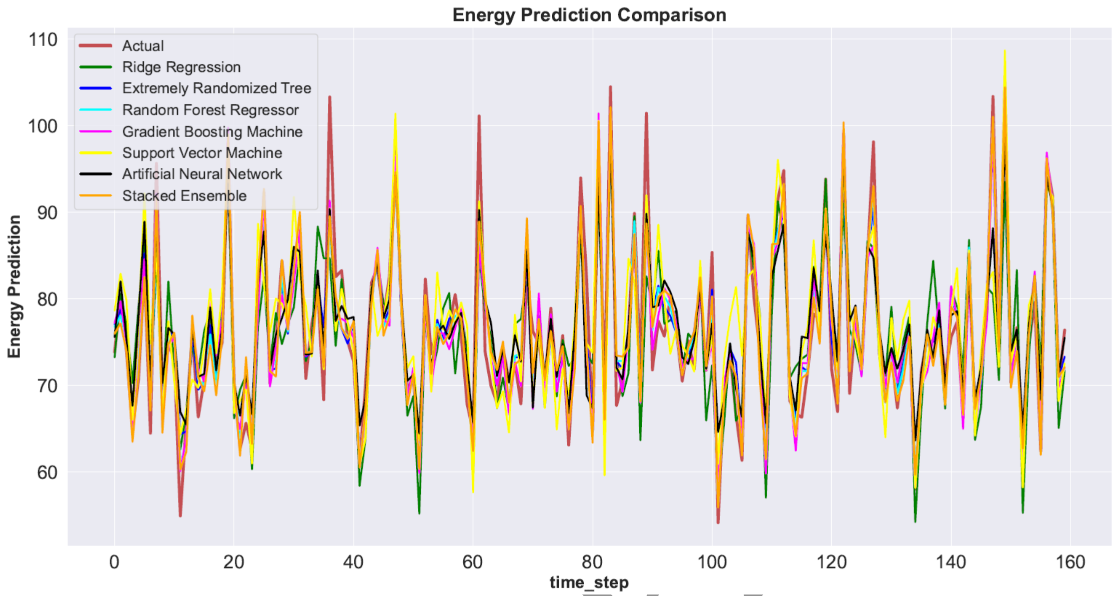


Figure 14 Comparison of predicted values of different models

The HBO-Ensemble model is subjected to a comparative analysis with other models such as Stacked Ensemble, RR, RF, SVM, SVR, ERT, GBM, and ANN. Table 1-7 displays the noteworthy hyper-parameters of the comparison models. In relation to the configuration and parameters of the models, certain heuristics are employed. The selection of an appropriate kernel function, such as linear, polynomial, or radial basis function (RBF) kernels, is a crucial aspect to consider when utilising SVMs for a given problem. For Random Forests, increasing the number of trees can improve performance up to a certain point, but beyond that point the benefits diminish and training time increases. When tuning hyperparameters, it's often better to use a coarse-to-fine approach, starting with a broad range of values and then narrowing down to a more specific range based on the results. When using ensemble methods like Stacked Ensemble, it's important to use diverse base models that have different strengths and weaknesses, in order to improve the overall performance of the ensemble. Strengths and weaknesses of different used models are as follows:

❖ Ridge Regression is a statistical technique that constructs a linear model, offering straightforward interpretability but suboptimal fitting.

© PUNEET KAUSHIK

- The training process of Extremely Randomised Trees is characterised by its speed and reduced variance in comparison to Random Forest, as it utilises all variables.
- Random Forest models possess strong generalisation ability due to their reduced sensitivity towards important variables. For instance, these models do not solely rely on the Power Factor to predict the Active Power, thereby minimising the risk of overfitting.
- Gradient Tree Boosting exhibits a greater susceptibility to overfitting in comparison to Random Forest. However, it may occasionally attain superior precision.
- Support Vector Machines exhibit strong resilience to high-dimensional data, demonstrate efficient handling of substantial datasets, and possess a notable capacity for generalisation, as they strive to identify the maximum margin separation.
- Artificial Neural Networks possess the capability to simulate continuous functions, which is not feasible with tree-based techniques. However, they are relatively more susceptible to overfitting as compared to Support Vector Machines.

Table 8 presents the error comparison of each model on the testing set, while the calculation method of errors is demonstrated in equations (4.1, 4.2, 4.3, 4.4). In order to present the prediction outcomes in a more comprehensible manner, the prediction curves for a single day within the test dataset were chosen and depicted in Figure 14.

Table 8 demonstrates that the HBO-Ensemble model surpasses other comparable techniques, such as Stacked Ensemble, Ridge Regression, Random Forest, SVM, SVR, Extremely Randomised Tree, Gradient Boosting Machine, and ANN. The HBO-Ensemble model exhibits a significant enhancement in predictive precision. The proposed model has resulted in a 0.001 improvement in the value of weighted MAPE for Ensemble 0.0288, which now stands at 0.027. In comparison to the MAPE, RMSE, and Percentual metrics of alternative models, the HBO-Ensemble model exhibited a 0.01 improvement in its Percentual metric.

The Root Mean Square Error (RMSE), Mean Absolute Percentage Error (MAPE), and Percentual distribution of the proposed model were evaluated and compared with other models in the test set. The results are presented in Figures 15-17.

© PUNEET PAHADIA

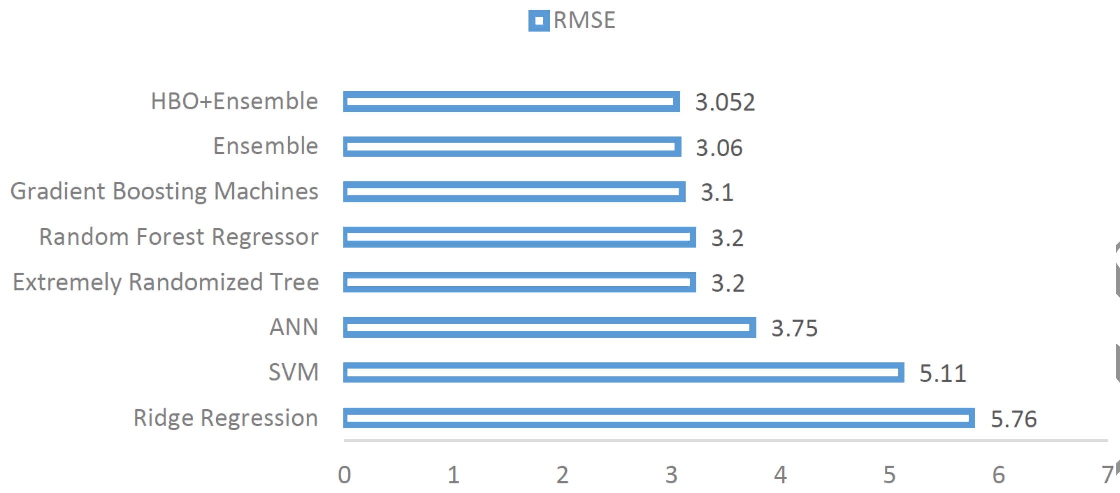


Figure 15 RMSE Comparison Plot

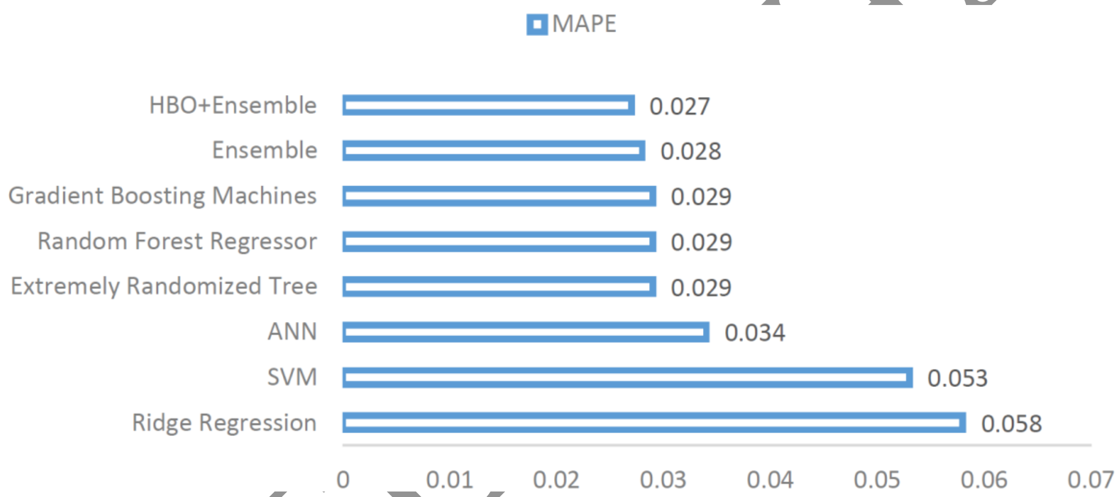


Figure 16 MAPE Comparison Plot

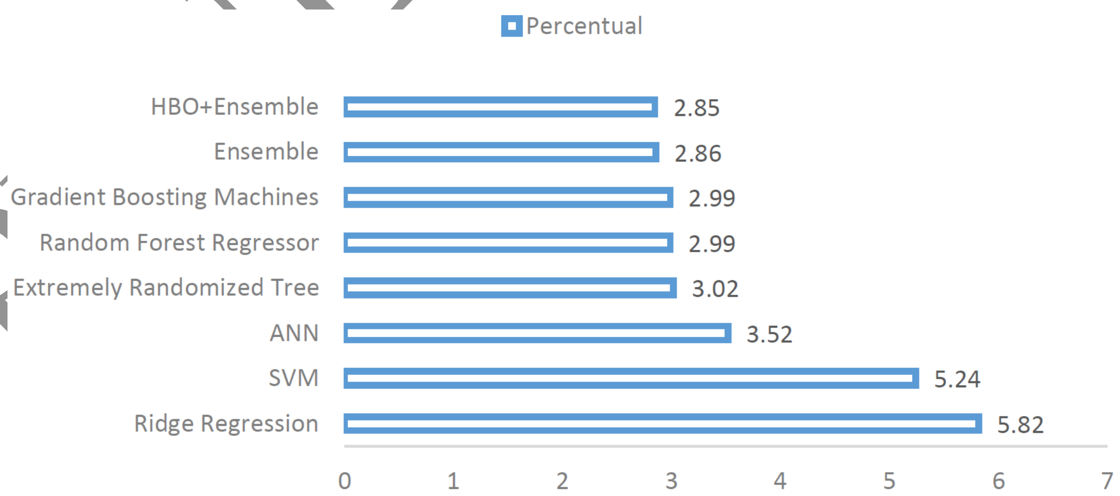


Figure 17 Percentual Comparison Plot

The stability of the proposed model's performance is apparent from the data presented in Figure 15-17. The HBO-Ensemble model exhibits the lowest Mean Absolute Percentage Error (MAPE) in comparison to alternative models. It is noteworthy that the Ensemble model exhibited favourable results in the conducted experiment, ranking second in overall performance after the proposed HBO-Ensemble model. This outcome serves as an indication of the exceptional performance of the encoder-decoder structure. Combination of multiple models with different strengths and weaknesses can help to reduce bias and variance in the final predictions, leading to better overall performance. This result supports the effectiveness of ensemble models in improving the performance of ML models

Regarding the duration of the training process, each model conducts its training independently. The independent forecasting models that operate on a single load execute parallel training, resulting in a time cost equivalent to the maximum time required for a single task. Table 3 displays the duration of mandatory training for recording.

Table 3- Training Time Comparison of Different Algorithms

Machine Learning Model	Training Time (In Seconds)
Ridge Regression	0.019493818283081055
Extremely Randomized Tree	15.06816577911377
Random Forest Regressor	72.37776017189026
Gradient Boosting Machines	64.70265936851501
Support Vector Machine	0.9426419734954834
Artificial Neural Network	185.60636830329895
Stacked Ensembling	870.4956455230713
HBO+Ensemble	392.3873143196106

Table 3 demonstrates that despite the intricate network architecture of the HBO-Ensemble model, the training duration remains within acceptable limits given the current hardware constraints. This indicates that the proposed HBO-Ensemble model can yield more precise prediction outcomes in a timely manner. In comparison to Support Vector Machines (SVM) and Artificial Neural Networks (ANN), HBO-Ensemble demonstrates a similar training time while possessing a greater number of trainable parameters, thereby indicating its superiority. In contrast to alternative models such as Random Forest and SVR, the HBO-Ensemble model exhibits a longer duration of training in order to achieve heightened levels of predictive accuracy.

© PUNEET PAHADIA

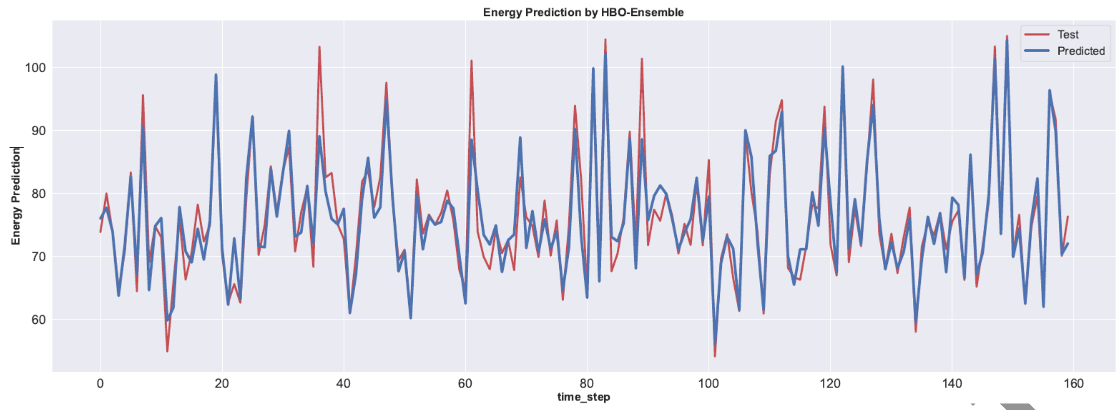


Figure 18 Comparison of predicted values Ensemble and HBO-Ensemble models

The results reveal variations in precision among different loads, indicating that the prediction difficulty varies across different loads. Intuitively, some loads are more challenging to predict than others. Fig. 18 illustrates that the proposed model performs better in predicting the electricity consumption of a building. This improvement can be attributed to the model's ability to capture the strong internal regularity of consumption. Forecasting the energy consumption patterns of individual users poses a greater level of difficulty in comparison to that of user groups. Moreover, the complexity of forecasting can be influenced by the existence of significant energy consumers, given that the load capacity is a contributing factor.

6.3 Comparison with Ensemble Model

The Ensemble model shares a similar architecture with the HBO-Ensemble model. However, the HBO-Ensemble model incorporates an HBO (Hyperparameter Optimization) architecture, which aids in fine-tuning the model's hyperparameters. In the evaluation, the performance of the proposed model, HBO-Ensemble, is compared with the Ensemble model, as depicted in Fig 18.

Figure 18 illustrates that the HBO-Ensemble exhibits a lower Mean Absolute Percentage Error (MAPE) compared to the Ensemble. This finding highlights the potential of the HBO-Ensemble in leveraging internal connections and knowledge transfer to enhance the precision of forecasting. The analysis reveals that the MAPE change range exhibited by HBO-Ensemble is comparatively narrower than that of Ensemble. This observation suggests that HBO-Ensemble is capable of identifying the correlation rule between the loads, thereby demonstrating superior generalisation ability. The need for a comprehensive consideration of the correlation between loads in

© PUNEET PAHADIA

building energy load prediction can better reflect the performance difference between two models.

6.4 Feasibility Analysis

This analysis discusses the feasibility of applying HBO to an ensemble model will be discussed.

1) **Enhanced Model Performance:** Ensemble models combine multiple base models to improve predictive performance by leveraging the strengths of individual models. By implementing HBO, the ensemble can be dynamically managed by prioritizing and updating the most relevant models based on their performance. This approach has the potential to enhance the overall model accuracy and predictive power.
2) **Efficient Resource Allocation:** HBO allows for efficient resource allocation by prioritizing models that contribute the most to the ensemble's performance. By leveraging a priority queue, the most relevant models can be selected for prediction or ensemble updates, while less relevant or underperforming models can be removed or deprioritized. This ensures that computational resources are focused on the most valuable models, resulting in improved efficiency.
3) **Adaptability to Changing Data:** In real-world scenarios, data distributions and patterns may change over time. HBO offers the advantage of adaptability to changing data. By continuously monitoring and updating the ensemble models based on their performance, the ensemble can quickly adapt to new data patterns or emerging trends. This flexibility enhances the model's ability to handle dynamic and evolving data, leading to more accurate predictions.
4) **Scalability and Parallelization:** Ensemble models, especially those with a large number of base models, can be computationally demanding. HBO enables efficient parallelization by assigning different models to separate processing units. This parallelization capability enhances scalability, making it feasible to handle large ensembles and process a high volume of data in a timely manner.
5) **Improved Generalization:** Ensemble models aim to improve generalization by reducing overfitting and capturing a broader range of patterns in the data. HBO enhances this aspect by dynamically managing the ensemble's composition. By continuously updating the ensemble with the most informative models, the model's ability to generalize to unseen data can be further improved.

© PUNEET PAHADIA

The HBO-Ensemble model has been found to be more effective in accurately predicting building energy consumption when compared to traditional consumption forecasting techniques along with other neural network methods.

© PUNEET PAHADIA

Chapter: 7

Conclusion

The key objective of this work was to create a framework for energy consumption forecast that is accurate and trustworthy. This model is essential for maximising energy use and encouraging sustainability in buildings.

Several well-known algorithms, including ERT, RR, GBT, RF, ANN, and SVM, were evaluated in terms of their MAPE, RMSE, and Percentual accuracy. Additionally, an ensemble approach and the proposed Heap-Based Optimization-Ensemble (HBO-Ensemble) were implemented to further enhance the predictive performance.

The outcomes of the experimentation phase show how well the various algorithms forecast the energy consumption of buildings. Among the individual algorithms, RR, ERT, and GBoosting achieved the lowest MAPE, RMSE, and Percentual values, indicating their superior predictive performance. Notably, the ensemble model demonstrated improved results compared to the individual algorithms, achieving a lower MAPE, RMSE, and Percentual.

Moreover, the proposed HBO-Ensemble algorithm outperformed all other approaches, including the ensemble model, by yielding the lowest MAPE, RMSE, and Percentual values. This highlights the significant contribution of the novel optimization technique in enhancing the accuracy of energy consumption predictions.

The results of this research offer significant insight into how ML algorithms might be used to provide predictions for the energy use of buildings. The results demonstrate the potential of ensemble methods and the effectiveness of the HBO-Ensemble algorithm in achieving improved prediction accuracy. These outcomes are essential for informing decision-making processes related to energy management and facilitating the development of energy-efficient strategies for buildings.

Future research in this field could focus on expanding the dataset, incorporating additional relevant features, and exploring other optimization techniques to further improve the prediction accuracy. Furthermore, investigating the generalizability of the proposed approach across different geographical regions and building types would provide a comprehensive understanding of its applicability.

© PUNEET PAHADIA

PUBLICATIONS

- P.Pahadia, P.Jain, A.K.Agarwal, A. Prajesh and A.Harit, **“Stacked Ensemble Approach for Building Energy Consumption Prediction”** *14th International Conference on Computing, Communication and Networking Technologies (ICCCNT)* [ACCEPTED]

- P.Pahadia, P.Jain, A.K.Agarwal, A. Prajesh and A.Harit. **“HBO-Ensemble: Based Building Energy Consumption Prediction”** *energy*. [SUBMITTED]

© PUNEET PAHADIA

REFERENCES

[1] IEA, “Energy efficiency 2019 – analysis,” *IEA*. [Online]. Available: https://www.iea.org/reports/energy-efficiency-2019

[2] L. Zhang *et al.*, “A review of machine learning in building load prediction,” *Appl Energy*, vol. 285, p. 116452, 2021.

[3] R. Olu-Ajayi, H. Alaka, I. Sulaimon, F. Sunmola, and S. Ajayi, “Building energy consumption prediction for res`idential buildings using deep learning and other machine learning techniques,” *Journal of Building Engineering*, vol. 45, p. 103406, 2022.

[4] P. Nie, M. Roccotelli, M. P. Fanti, Z. Ming, and Z. Li, “Prediction of home energy consumption based on gradient boosting regression tree,” *Energy Reports*, vol. 7, pp. 1246–1255, 2021.

[5] M. B. Afuosi and M. R. Zoghi, “Indoor positioning based on improved weighted KNN for energy management in smart buildings,” *Energy Build*, vol. 212, p. 109754, 2020.

[6] Z. Yu, F. Haghighat, B. C. M. Fung, and H. Yoshino, “A decision tree method for building energy demand modeling,” *Energy Build*, vol. 42, no. 10, pp. 1637–1646, 2010.

[7] K. Yan, W. Li, Z. Ji, M. Qi, and Y. Du, “A hybrid LSTM neural network for energy consumption forecasting of individual households,” *IEEE Access*, vol. 7, pp. 157633–157642, 2019.

[8] Y. Tian, S. Yu, M. Wen, K. Zhang, and Y. Chen, “Short-term load forecasting scheme based on improved deep residual network and LSTM,” in *CIRED 2020 Berlin Workshop (CIRED 2020)*, 2020, pp. 117–120.

[9] N. Somu, G. R. MR, and K. Ramamritham, “A hybrid model for building energy consumption forecasting using long short term memory networks,” *Appl Energy*, vol. 261, p. 114131, 2020.

[10] S. Seyedzadeh, F. P. Rahimian, I. Glesk, and M. Roper, “Machine learning for estimation of building energy consumption and performance: a review,” *Visualization in Engineering*, vol. 6, pp. 1–20, 2018.

© PUNEET PAHADIA

[11] X. J. Luo, L. O. Oyedele, A. O. Ajayi, and O. O. Akinade, "Comparative study of machine learning-based multi-objective prediction framework for multiple building energy loads," *Sustain Cities Soc*, vol. 61, p. 102283, 2020.

[12] H. Lu, F. Cheng, X. Ma, and G. Hu, "Short-term prediction of building energy consumption employing an improved extreme gradient boosting model: A case study of an intake tower," *Energy*, vol. 203, p. 117756, 2020.

[13] Y. Li, "Prediction of energy consumption: Variable regression or time series? A case in China," *Energy Sci Eng*, vol. 7, no. 6, pp. 2510–2518, 2019.

[14] T. C. Carneiro, P. A. C. Rocha, P. C. M. Carvalho, and L. M. Fernández-Ramirez, "Ridge regression ensemble of machine learning models applied to solar and wind forecasting in Brazil and Spain," *Appl Energy*, vol. 314, p. 118936, 2022.

[15] D. Liu and K. Sun, "Random forest solar power forecast based on classification optimization," *Energy*, vol. 187, p. 115940, 2019.

[16] L. Wehenkel, D. Ernst, and P. Geurts, "Ensembles of extremely randomized trees and some generic applications," in *Robust Methods for Power System State Estimation and Load Forecasting*, 2006.

[17] P. Geurts, D. Ernst, and L. Wehenkel, "Extremely randomized trees," *Mach Learn*, vol. 63, pp. 3–42, 2006.

[18] S. Touzani, J. Granderson, and S. Fernandes, "Gradient boosting machine for modeling the energy consumption of commercial buildings," *Energy Build*, vol. 158, pp. 1533–1543, 2018.

[19] V. K. Ayyadevara and V. K. Ayyadevara, "Gradient boosting machine," *Pro machine learning algorithms: A hands-on approach to implementing algorithms in python and R*, pp. 117–134, 2018.

[20] J. R. Lloyd, "GEFCom2012 hierarchical load forecasting: Gradient boosting machines and Gaussian processes," *Int J Forecast*, vol. 30, no. 2, pp. 369–374, 2014.

[21] Y. Lin, Y. Lee, and G. Wahba, "Support vector machines for classification in nonstandard situations," *Mach Learn*, vol. 46, pp. 191–202, 2002.

© PUNEET PAHADIA

[22] H.-C. Kim, S. Pang, H.-M. Je, D. Kim, and S. Y. Bang, "Constructing support vector machine ensemble," *Pattern Recognit*, vol. 36, no. 12, pp. 2757–2767, 2003.

[23] T. Catalina, J. Virgone, and E. Blanco, "Development and validation of regression models to predict monthly heating demand for residential buildings," *Energy Build*, vol. 40, no. 10, pp. 1825–1832, 2008.

[24] B. B. Ekici and U. T. Aksoy, "Prediction of building energy consumption by using artificial neural networks," *Advances in Engineering Software*, vol. 40, no. 5, pp. 356–362, 2009.

[25] Z. Ma, C. Ye, H. Li, and W. Ma, "Applying support vector machines to predict building energy consumption in China," *Energy Procedia*, vol. 152, pp. 780–786, 2018.

[26] L. Breiman, "Heuristic of instability in model selection," 1994.

[27] D. Opitz and R. Maclin, "Popular Ensemble Methods: An Empirical Study," *Journal of Artificial Intelligence Research*, vol. 11, pp. 169–198, 1999.

[28] C. Fan, F. Xiao, and S. Wang, "Development of prediction models for next-day building energy consumption and peak power demand using data mining techniques," *Appl Energy*, vol. 127, pp. 1–10, 2014.

[29] K. Siwek, S. Osowski, and R. Szupiluk, "Ensemble neural network approach for accurate load forecasting in a power system," *International Journal of Applied Mathematics and Computer Science*, vol. 19, no. 2, pp. 303–315, 2009.

[30] A. C. Tan and D. Gilbert, "Ensemble machine learning on gene expression data for cancer classification," 2003.

[31] G. Giacinto and F. Roli, "Design of effective neural network ensembles for image classification purposes," *Image Vis Comput*, vol. 19, no. 9–10, pp. 699–707, 2001.

[32] J.-S. Chou and D.-K. Bui, "Modeling heating and cooling loads by artificial intelligence for energy-efficient building design," *Energy Build*, vol. 82, pp. 437–446, 2014.

© PUNEET PAHADIA

[33] Q. Askari, M. Saeed, and I. Younas, "Heap-based optimizer inspired by corporate rank hierarchy for global optimization," *Expert Syst Appl*, vol. 161, p. 113702, 2020.

[34] A. A. Ewees, M. A. A. Al-qaness, L. Abualigah, and M. Abd Elaziz, "HBO-LSTM: Optimized long short term memory with heap-based optimizer for wind power forecasting," *Energy Convers Manag*, vol. 268, p. 116022, 2022.

[35] J. Gareth, W. Daniela, H. Trevor, and T. Robert, *An introduction to statistical learning: with applications in R*. Spinger, 2013.

[36] S. J. Russell, *Artificial intelligence a modern approach*. Pearson Education, Inc., 2010.

[37] K. Faceli, A. C. Lorena, J. Gama, and A. C. P. de L. F. de Carvalho, "Inteligência artificial: uma abordagem de aprendizado de máquina," 2011.

[38] A. Silberschatz, H. F. Korth, and S. Sudarshan, "System Concepts," 2008.

[39] R. Kumar, P. Kumar, and Y. Kumar, "Time series data prediction using IoT and machine learning technique," *Procedia Comput Sci*, vol. 167, pp. 373–381, 2020.

[40] T. Khatib, A. Mohamed, K. Sopian, and M. Mahmoud, "Assessment of artificial neural networks for hourly solar radiation prediction," *International Journal of Photoenergy*, vol. 2012, 2012.

[41] Y. Dai, M. Khandelwal, Y. Qiu, J. Zhou, M. Monjezi, and P. Yang, "A hybrid metaheuristic approach using random forest and particle swarm optimization to study and evaluate backbreak in open-pit blasting," *Neural Comput Appl*, pp. 1–16, 2022.

[42] C. J. Willmott and K. Matsuura, "Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance," *Clim Res*, vol. 30, no. 1, pp. 79–82, 2005.

© PUNEET PAHADIA