Mathematical Statistician and Engineering Applications
ISSN: 2094-0343
2326-9865

Real-time Computation of Features Based on Probabilistic Methods in Machine Learning

Rahimunisa Begum
Computer Science and Engineering
Institute of Aeronautical Engineering
Hyderabad, India unisarahim18@gmail.com

Dr. C. Madhusudhana Rao
Computer Science and Engineering
Institute of Aeronautical Engineering
Hyderabad, India
cmrao@iare.ac.in

Article Info
Page Number: 2169-2179
Publication Issue:
Vol. 72 No. 1 (2023)

Abstract—Traditional energy sources at a residential level for a sustainable environment. Energy consumption in Kilowatts(kW) in each household varies depending on the appliances used. To store energy in an emergency and maintain the continuity of supply-demand energy predictions have been developed. General Regression Neural Network(GRNN) uses probabilistic machine learning with uncertainty into consideration for prediction results. GRNN implemented on the energy consumption dataset containing humidity and temperature data columns considers errors and value uncertainty of the data. The GRNN is implemented against commonly used Tree-based algorithms like Random Forest Regressor, XGBoost Regressor, and Extra Tree Regressor performs well. The GRNN slightly outperforms the tree-based regressors with an R2(accuracy scale for neural networks)score of 0.61 and the best tree regressor of 0.58. The probabilistic approaches have better accuracy than the non-probabilistic approaches for prediction problems as they use distribution as inputs rather than valued data. Advanced data acquisition techniques can improve the R2 scores of GRNN by providing more data for the neural network model.

Article History
Article Received: *15 November 2022*
Revised: *24 December 2022*
Accepted: *18 January 2023*

Keywords—Prediction, Real-time, General Regression Neural Networks, Extra tree regressor, XGBoost regressor, Probabilistic Neural Network, Bayesian Networks, Energy Prediction, Uncertainty Prediction, GRNN

.

Introduction

The energy consumption prediction area has seen lots of advanced methodologies implemented lately. Renewable energy resources are replacing the current supplied by stations by using solar and wind sources for cleaner energy and environmental-friendly. Renewable energy sources are present in abundance but are not continuous due to factors like changes in seasons and weather. Prediction algorithms help to understand the analytics of the energy used and energy stored in batteries as backup for a certain period. Analytics helps in planning energy usage, saving, and tracking.

Commonly used methods for the prediction of energy consumption are direct tree-based regressions. Most regression algorithms are not probabilistic and do not consider uncertainty in the data. For such instances, a probabilistic approach is the best way to deal with such problems. Along with probability, dynamics between the attributes also improve the accuracy of predictions. Neural Networks are the best way to determine the linear and non-linear relationships among the features.

The dataset used in the prediction of the total energy consumption of a household contains temperature, humidity, wind speed, dewpoint, and visibility readings across various points. The training data is 75%, and the testing data is 25% of the dataset. Since the recordings are in different units it is used in a standardised format to calculate the exact values of both linear and non-linear correlations.

Probabilistic Neural Network models that perform regression are called General Regression Neural Networks. A GRNN is a regression performed using a probabilistic approach to a neural network. A GRNN represents a training sample to a mean radial basis neuron. The GRNN handles noises in data well, and learning happens in a single pass with no backpropagation. Depending on the size of the data passed it becomes computationally extensive.

Tree-based regressors values are compared to the GRNN model. Random Forest Regressor uses an ensemble learning method for regression by multiple model values than a single model. Extra Tree Regressor works similarly to random forest except that the split at nodes is randomly instead of the best optimal split (Greedy method). XGBoost regressor is the extended version of the gradient boosting algorithm. It combines the outputs of weaker models to predict the target value more accurately. Tree-based algorithms parameters perform differently, and only the best sets are used.

The software used to build the project is Google Colaboratory. Google Colab is a tool for data analysis and machine learning projects that provides built-in libraries and uses high-level languages such as R and Python in executable format. The files are stored in the drive folders and executed in the cloud. Colab is an easy tool that supports advanced libraries like Tensorflow and Keras without any high requirements of the desktop. It has also data visualization and exploration libraries like pandas, numpy, scikit-learn, seaborn, and matplot pre-installed.

A variety of apps can be made using the programming language Python. For projects including artificial intelligence (AI), machine learning, and deep learning, developers think it's a wonderful option.

Pandas library is used for data exploration.An open-source framework called Pandas is specifically designed for working quickly and logically with relational or labeled data. It offers a range of data structures and methods for working with time series and numeric data. On top of the NumPy library, this one is constructed. Pandas is quick, and its users can get a lot done with it. It uses the data frame concept to handle data and perform operations on them. Matplot library pyplot function to plot the graphs and visualize the data. The scikit-

learn library is used for the preprocessing of data and splitting the data into test and training parts of data for the model. The tree-based regressor is implemented with the sklearn library. GRNN is implemented using the TensorFlow and Keras libraries. Additional libraries like tensorflow.probability are used.

This report will consist of tree-based regressors and a probabilistic approach to the prediction of total energy consumption in Kw and their performance against each other.

Literature Review

Energy consumption prediction models of household appliances have been investigated by several studies. For example, a model was proposed for predicting the next-hour and next 24-h total energy consumption of household appliances using various machine learning (ML) algorithms, namely, the decision tree (DT) algorithm, decision table classifier (DTC), and Bayesian network (BN). The proposed approach was to formalize expert knowledge on energy consumption and find a suitable data structure for the regressor. Moreover, they showed that the selection of the proper regression model for a given dataset is nontrivial.

Another reported data-driven prediction methods for home appliance energy use. Repeated cross-validation was used to train four statistical models, which were later assessed on test data. A support vector machine (SVM) with a radial kernel, a gradient boosting machine (GBM), multiple linear regression (MLR), and random forest were used to create the proposed models (RF). The best outcomes, though, came from utilizing GBM.

Three algorithms were also utilized by others to estimate the on/off times of appliances with a 1-hour resolution. The histogram algorithm, pattern search method, and Bayesian inference algorithm were specifically utilized. The authors discovered that there was no clear correlation between the performances of the various methods on the same dataset (obtained from a specific appliance).

The energy consumption of buildings as a whole was predicted using several models built on artificial neural networks (ANNs) and support vector machines (SVMs). They came to the conclusion that it is challenging to choose the optimal model without comparing each model for the same set of circumstances.

Several models suggested case-based reasoning and ANN-based energy prediction methods. An institutional building's hourly electricity consumption was predicted using these methods. The case-based reasoning models were regularly outperformed by the ANN-based models.

According to the literature currently in use, choosing the optimum regression model for a given issue and dataset is not an easy task. A regressor that works well for one issue can frequently fail to solve it for another. Also, it is not simple to choose the appropriate set of features for a particular regressor. Consequently, in this study, we suggest a technique that automatically chooses a suitable regressor and the related feature set. The RF regressor (RFR), extra trees regressor (ETR), decision tree regressor (DTR), and K-nearest neighbors regressor were the four regression techniques we employed (KNNR). It should be noted that in our method, the number of regressors employed depends on the user's needs and is easily modifiable.

In most real-world situations, artificial intelligence methods, which have been extensively used to estimate building energy usage, can produce more accurate prediction results. For predicting building energy usage, clusterwise regression, a unique technique that combines clustering and regression simultaneously, has been developed.

The commonality of pattern sequences for electricity price and demand prediction was discovered using a clustering algorithm. The pattern of electricity use in buildings was examined using the k-means approach. In addition, time series forecasting relating to power was examined using data mining approaches. A decision tree was employed to forecast energy consumption levels and comprehend energy consumption patterns.

Building energy efficiency has been improved with the aid of random forest (RF), according to facility managers. The energy consumption of low-energy buildings was predicted using support vector machines (SVM) and a pertinent data selection technique. A variety of ANN types have been provided for this application since artificial neural networks (ANNs) are crucial for predicting building energy use. The bioclimatic building's electricity use was predicted in the short term using an ANN model. To predict the energy consumption of residential buildings, an ANN based on the Levenberg-Marquardt and Output-Weight-Optimization (OWO)-Newton algorithms were used. The prediction of building energy usage looked at an ANN along with a fuzzy inference system.

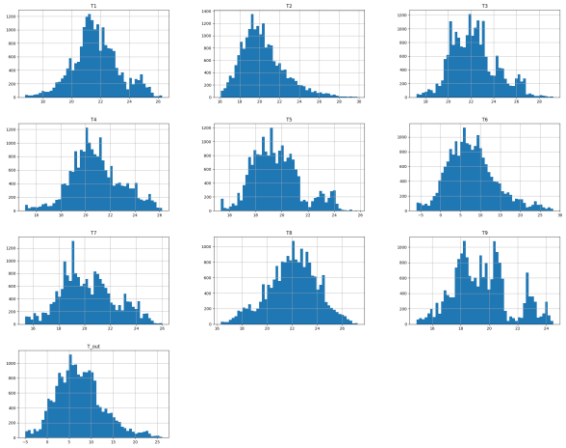
Real-time online building energy prediction was proposed using two adaptive ANNs with accumulative training and sliding window training. Also, a suggested ANN trained by an extreme learning machine (ELM) was contrasted with a genetic algorithm (GA)-based ANN for estimating building energy usage. The building's energy efficiency was also increased using a hybrid technique that combines the particle swarm optimization (PSO) algorithm with the radial basis function neural network (RBFNN). Although statistical approaches and current artificial intelligence techniques can produce satisfactory results, it is still difficult to make precise predictions due to random factors that can be impacted by the weather, working hours, population distribution, and construction equipment.

Contrarily, the deep learning methods that have surfaced in recent years give us a strong tool to improve modeling and prediction performance. The layer-wise pre-training method is used by the deep learning algorithm in combination with deep architectures or multiple-layer architectures to achieve excellent feature learning capabilities.

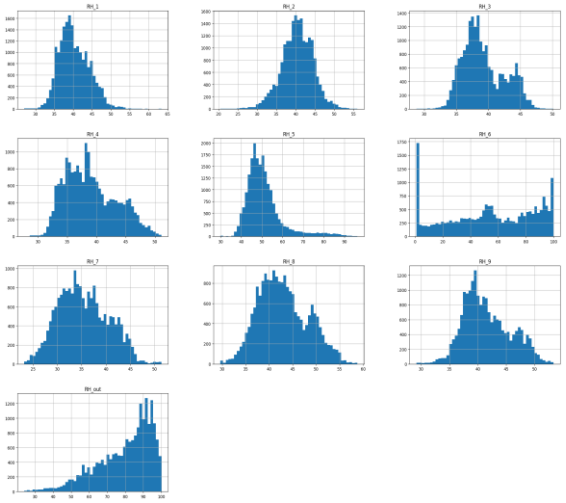
Methodology

Data acquisition is performed initially and the data is collected from Kaggle. The dataset consists of 19734 recordings against all 23 columns. Normalisation is applied to the dataset on the same scale for accurate model prediction. Visualization is performed using the Numpy library on all 23 features across all recorded data instances. d| is the normalized value for the data. d, dmin, and dmax are the data value, minimum value for the data column, and maximum value for the data column respectively.

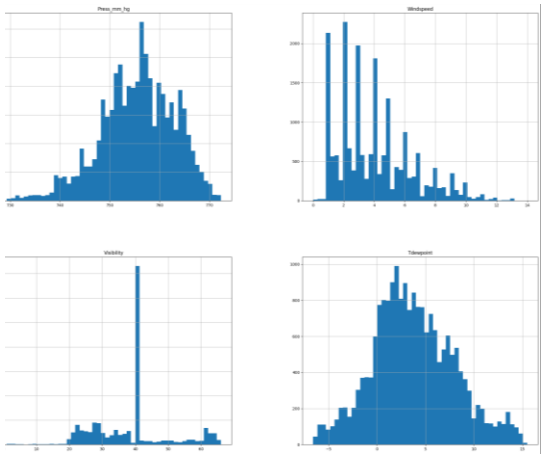
$$d| = (d - dmin) / (dmax - dmin)$$



Distributions of temperature columns



Distributions of pressure, windspeed, visibility, and dewpoint



The target variable is total energy consumption in Kw (Kilowatt). The scikit-learn library is used to split normalised datasets into training and testing samples in the ratio of 0.75:0.25. The function train_test_split is implemented from the scikit library with the parameter value of random_state=0. For every model Extra tree regressor, Random forest regressor, XGBoost regressor, and GRNN the same split ratio of the train to test data is applied.

GRNN is used to generate continuous values. The by-product of the GRNN is Bayesian posterior probabilities. The parameter for optimal learning of the GRNN is a single radial basis kernel function bandwidth ($\boldsymbol{\sigma}$).

The conditional expectation of J on I=i is the regression of GRNN. The output for input vector x is the probable scalar Y. The joint continuous probability density function for an input vector I and output scalar J be f(i, J). Then the regression of GRNN, or the conditional probability of (J on i) is given as

$$E[y|\boldsymbol{X}] = \frac{\int_{-\infty}^{\infty} yf(\boldsymbol{X}, y)\, dy}{\int_{-\infty}^{\infty} f(\boldsymbol{X}, y)\, dy}.$$

If the dependent J and independent I relationship is expressed in the form of functional parameters then the regression is parametric. If the relationship between J and I is not known the non-parametric estimation is used. Gaussian function estimators are used to estimate non-parametric f(i, j). The partial first derivative of f(i,j) for random variables I and J is the best probability estimator f^(i,j) given by:

- P: is the dimension of the vector variable.
- N: is the number of training pairs $(ix \rightarrow jx)$.
- σ: smoothing parameter is chosen during network training.
- Jx: is desired scalar output given the observed input ix

GRNN consists of four layers.The first layer is the input layer for the initial recordings of the dataset fed into the neural network. The first hidden layer is the second layer of the GRNN to each input a vector is assigned and the difference in the vector to the training value is subtracted and squared in this layer to pass onto the third layer.

The third layer in GRNN is known as the summation layer. This layer has two nodes one stores the summation of all the training values and the second stores the summation of observed values over the training vectors.

The fourth layer is the last layer in GRNN. It gives the regression value of J on i.

The only parameter that can be changed in the GRNN structure is the smoothing parameter (σ). The best value for smoothing is found using the holdout method.

For a random forest regressor, a subset of data points is used to make a decision tree for a selected number of records and each sample. Every decision tree will generate an output. Since the total energy consumption is a continuous value, the Averaging method is used for regression.

$$\text{GINI INDEX} = 1 - \sum_{i=1}^{n} p(i)2$$

p: is the probability of class

To implement Extra tree regressor it can be directly used in the format of a built-in function from the library sklearn. Extra tree regressor works in the same way as the Random Forest classifier. An extra tree regressor is also known as Extreme Randomised Tree In the first step, decision trees are made for each sample. The output of each tree is taken and an averaging method is applied to get the final output.

In the Extra tree regressor, the node is not split using GINI Index instead it is split very randomly.

The last method to predict the total consumption of energy is the XGBoost regressor. XGBoost is also known as the Gradient Boosting method. In XGBoost, weak learner regression trees map a data point input to a leaf node of its own tree that stores a continuous value. L1 and L2 regularizations are minimized in XGBoost by combining a convex loss function (difference in predicted and target output) and model complexity penalty term (functions of regression tree).

XGBoost is one of the recently widely used algorithms for the ease of scalability by parallel fast learning and efficient usage of memory by distributed computing. XGboost uses ensemble learning to maximize the output by averaging the multiple outputs by trees.

It works in three steps. To predict j an initial model M0 is defined. Model M0 will be associated with a residual M0-j.

A new model n1 is fit to the residuals from the previous step. The boosted version of M0, M1 is a combined form of both M0 and n1. The MSE of M1 will be less than M0.

$$\mathbf{M1(x) \leftarrow M0(x) + n1(x)}$$

After ‘k’ iterations:

$$\mathbf{Mk(x) \leftarrow Mk\text{-}1(x) + nk(x)}$$

Usually the loss function for XGBoost is MSE, Mean Squared Error, the change will be exponential slightly. Instead of nk(x) fitting on the residuals, fitting on the gradient loss function, at a step when loss occurs. This results in generalization and applicability across all various loss functions.Differential functions can be minimized by gradient descent. Prediction of mean residual at each terminal node of regression tree for nk(x). In XGBoosting, the average gradient component is computed.

A factor is multiplied at each node with nk(x). It represents the impact of the difference in each split of the tree. Prediction of the optimal gradient is done by XGBoost for the additive model. This differs from classical gradient descending which requires loss at each step of the algorithm. XGboost can be directly imported from the xgboost library in the Google colab.

The following way XGBoost works is applied:

1. M0(x): initial point in XGboost is defined as follows.

M0(x)= argmin$\boldsymbol{\gamma}$ L(Jx, $\boldsymbol{\gamma}$)

2. The gradient loss function is computed iteratively by

rxm=-$\boldsymbol{\alpha}$[$\boldsymbol{\partial}$(L(Jx, xi))/$\boldsymbol{\partial}$F(xi)]F(x)=Fm-1(x), $\boldsymbol{\alpha}$ is the learning rate

3. The gradient obtained fits each nk(x)

4. The multiplicative model $\boldsymbol{\gamma}$a for boosted model M0(x) is defined as

Mk(x) ← Mk-1(x) + $\boldsymbol{\gamma}$ank(x)

The overfitting of data in the XGBoost is prevented using L1 and L2 regularization and on penalized complex models.

Results

From the correlation plot, the temperature columns are highly correlated, and the humidity columns are least correlated. The Metric that decides the accuracy/performance of the is R2 score. R2 score is also known as the coefficient of determination. R2 score in terms of regression means the ability of the regression line to fit all the data points. An R2 score of 0.72 means for every 100 data points, the regression line fits 72 values correctly. The R2 score lies between 0 to 1. Values closer to 1 imply the model fits the regression line to original values.

The R2 is calculated as follows:

R2 score = (total variance explained by the model) / total variance

The general regression neural network has the highest R2 score of 0.61. Among tree-based regressors, the Extra tree regressor has an R2 score and is most closely related to GRNN is 0.58. The values for the R2 score of the Random forest and 0.48 and 0.22 respectively.

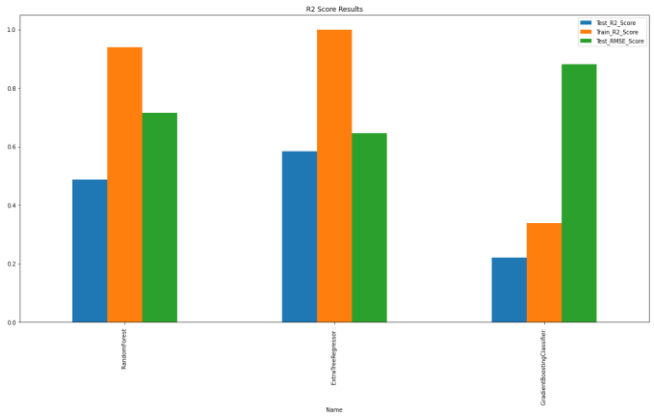
For GRNN, the value of the smoothing parameter() was 0.2. GRNNs outperformed tree-based regressors in predicting the total energy consumed by a house. The mid value for consumption of total energy was 230.0 kW.

The table below shows the algorithms used and their recorded R2 scores:

Algorithm Used to calculate the consumption of energy in Kilowatts	Recorded R2 Score
General Regression Neural Network(GRNN)	0.61
Random Forest Tree Regressor	0.48
Extra tree Regressor	0.58
Gradient Boosting (or) XGBoost	0.22

Table: comparison of recorded R2 scores by all models.

The following graph is a pictorial representation of tree-based regressors' performance for the regression of energy consumption. The histogram in green color represents the model's accuracy and the orange histogram represents the original regression with all samples as training.



Conclusion

Probabilistic-based approaches perform better than built-in tree-based regressors. Probability takes consideration of the underlying uncertainty in the data and performs well on noisy data. General regression neural network works similarly to a radial basis neural network. Like radial basis neural networks, the GRNN can be helpful in function approximation, time series prediction, classification, and for control systems. Unlike the simplicity and readability of tree-based regressors, GRNNs are complex neural networks. GRNNs perform well with the

noisy data in the dataset compared to the performance of tree-based regressors. Computational expenses are more in GRNNs than tree-based regressors.

The time required to solve is much lower in GRNN than in the tree-based methods. Among the tree-based methods, the Extra tree regressor performs well, and its accuracy is close to GRNN. Random Forest and XGBoost tree regressors perform very poorly compared to the Extra tree regressor.

In the Extra tree regressor, the split at the node is done randomly, unlike the Random forest that splits based on the GINI index value. Both algorithms follow the next step to choose the next best random feature among the available subset of features. But the extra tree regressor adds randomness with optimization and outperforms the Random Forest regressor. Also, random forest uses bootstrapping and replaces the subsets again and again. Whereas in the Extra tree regressor, the original sample is used. The variance is higher in the random forest regressor due to bootstrapping. Hence, the extra tree regressor performs with better accuracy than the random forest tree regressor.

XGBoost regressor, or Gradient Boosting regressor, performs very poorly compared to all the methods employed. The Gradient Boosting algorithm is sensitive to the noise in the data. It can easily overfit the model if the depth of the tree increases with an increase in the feature. Therefore, the XGBoost tree regressor performs poorly. Regression tree-based approaches are not the first choice for achieving state-of-the-art accuracy on time series data. Instead, other characteristics are good interpretability and simplicity, which neural networks often lack in neural networks.

Tree-based regressors do not deal with the probability of values and the uncertainty in the data.

In conclusion, the best models with maximum accuracy for real-time systems are the probabilistic approaches. In this case, the General regression neural network is a probabilistic approach to the energy consumption of a house that outperforms all the tree-based regression methods like Random forest regressor, Extra tree regressor, and the XGBoost. Therefore, for real-time data prediction, probabilistic approach-based models are better.

Future work

The probabilistic methods help in considering the uncertainty of the data in the datasets. These findings might help people to use probabilistic methods like Bayes in the other domains for prediction and forecasting.

Acknowledgment

The satisfaction that accompanies the successful completion of any task would be incomplete without the mention of the people who made it possible and whose constant guidance and encouragement crown all the efforts with success. I thank our college management and respected Sri Marri Rajasekhar Reddy, Chairman, Institute of Aeronautical Engineering College, Dundigal for providing me with the necessary infrastructure to carry out the project work.

References

Dorin Moldovan, Adam Slowik (2021)

[1] 'Energy consumption prediction of appliances using machine learning and multi-objective binary grey wolf optimization for feature selection', *Applied Soft Computing*, Volume 111.

[2] Shwet Ketu, Pramod Kumar Mishra (2021) 'A Hybrid Deep Learning Model for COVID-19 Prediction and Current Status of Clinical Trials Worldwide', *Computer, Materials & Continua*, DOI: 10.32604/cmc.2020.012423.

[3] Alvarez, Sunil K V, Swaroop Kumar M L, Sajitha Banjan, Usha S Diggi (2021) 'MVAi: A Framework and Three-Stage Approach to Detect Detection', *IEEE Mysore Sub Section International Conference (MysuruCon)*, Hassan, India, 2021, pp. 523-529, doi: 10.1109/MysuruCon52639.2021.9641646.

[4] Jingya Ding, Mingxin Yu, Lianqing Zhu, Tao Zhang, Jiabin Xia, Guangkai Sun (2020) 'Diverse spectral band-based deep residual network for tongue squamous cell carcinoma classification using fiber optic Raman spectroscopy', *Photodiagnosis and Photodynamic Therapy*, Volume 32, p.102048.

[5] Guilherme Ribero, Camila Maione, Cristhiane Goncalves, Diego de Castro Rodrigues, Rommel Melgaco Barbosa (2021) 'Recent advances in detection and prediction of customers energy consumption patterns through the use of machine learning techniques', *International Conference on Engineering and Emerging Technologies (ICEET)*, Istanbul, Turkey, 2021, pp. 1-8, doi: 10.1109/ICEET53442.2021.9659738.

[6] Chengdong Li, Zixiang Ding, Dongbin Zhao, Jianqiang Yi, and Guiqing Zhang (2017) 'Building Energy Consumption Prediction: An Extreme Deep Learning Approach', *Energies*, *10*(10), p.1525.

[7] Chou, Jui-Sheng, and Ngoc-Tri Ngo (2019) 'Time series analytics using sliding window metaheuristic optimization-based machine learning system for identifying building energy consumption patterns, *Applied energy*, *177*, pp.751-770.