Predictive Modeling of Energy Consumption by Home Appliances.

Shilpu Srivastava
University of Ottawa
Ottawa, Ontario
ssriv071@uottawa.ca

***Abstract*—The demand for energy resources has been growing exponentially and an estimation of the energy consumption at the consumer level has become essential in terms of forecast and efficient planning of energy resources. Buildings contribute significantly to the consumption of energy and hence it is crucial to take appropriate measures to control the energy consumed by home appliances. The paper aims to perform a data-driven prediction of the energy usage of appliances in a low-energy house.The data used for the prediction comprises the energy use of the light fixtures and the temperature and humidity in all the rooms of the house. Some of the weather parameters of a nearby weather station, for instance, the wind speed, visibility, and dew-point temperature are also taken into consideration. The paper aims at building predictive models based on the machine learning algorithms, one from each of the (a) linear, (b) tree-based, (c) distance-based, (d) rule-based and (e) ensemble categories followed by an evaluation of the performance. Feature engineering has been deployed on the data set for filtering and eliminating the non-predictive parameters in order to exacerbate the accuracy and performance of the models. Based on the highest R-squared value and minimum Root Mean Square Error, the Random Forest regression model seems to yield the most optimal results. The the potential outlier entries in the data set are also identified using five different outlier detecting techniques and the entries that have been detected by a minimum of three or more methods are declared as outliers.**

***Index Terms*—Energy consumption, Machine learning, Regression, R-squared value, Root mean square error**

I. Introduction

With the dramatic increase in energy demand, management of energy resources has become crucial. The prediction of energy consumption is significant in this regard. This prediction can be useful in numerous applications that target efficient power consumption as with increasing industrialization, all countries are working towards reducing optimizing energy usage and also encourage using renewable sources of energy, be it farms, industries or the fuel consumption by the vehicles. Tracking energy consumption is challenging because of the fact that energy is derived from multiple sources [4]. However, the prediction of energy consumption can facilitate efficient planning and utilization of these resources in the most effective way. The predictive models have been effective in a large number of domains, for instance in detecting anomalous energy usage patterns [7], determining means of reducing power flow to the grid and facilitating load control [8].

Statistics show that approximately 40 percent of the total energy consumed globally has been discovered to be drawn from buildings and residential homes account for 75 percent of the energy consumed in buildings [3]. Also, with recent research in this domain, it has been predicted that energy consumption in the residences is going to observe a continuous surge until 2040 and hence active research is being put into practice to discover means of energy savings. With such a research coming into the limelight, the requirement of analyzing the physical, environmental and occupant behavioral factors [12] is gaining significance. It is interesting how the demographic changes in a nation have been affecting the energy resources on a very large scale. In countries like South Korea, the fashion of single-person households has seen to influence residential energy usage in a significant way [5]. Since situations like single-person households are under nobody's control, efficient means of optimizing energy usage in residences can be a potential solution.

The estimation of energy usage has emerged as an efficient mechanism as it is instrumental in helping the policy-makers to decide [9] how much energy might be required for accomplishing their tasks and what changes they could bring about in order to diminish energy usage. Since the amount of energy used is influenced by numerous internal as well as external factors,the prediction of energy usage becomes complicated. The energy consumption in domestic houses can be determined by two major internal factors that are the number of electrical appliances and their type and also the usage of these appliances by the occupants. Some of the environmental parameters can also influence energy usage of the appliances, for instance, the temperature, humidity [14] inside and outside the house. Sometimes the exterior environment factors like dew point temperature, windspeed, and visibility can also contribute towards determining the weather of a particular geographic location. The weather outside seems to have a correlation with the appliances used inside the house. For instance, if it has been snowing outside, people would want to turn their thermostats, geysers, and heaters on.

In this work, the appliance energy usage prediction has been carried out for a low energy house in Stambruges, which was constructed in December 2015 [2]. The house has been designed based on a passive house planning package tool that ensures that it has an annual cooling and heating load of no more than 15kW per meter square [2]. The house appliances contribute to approximately percent of the energy consumption because the light fixtures are LEDs that do not make much of

a difference. Zigbee wireless sensors are deployed in the house to monitor the temperature and humidity conditions [2]. The magnitude of temperature and humidity in 9 different rooms of the house that have a variety of appliances are considered as parameters for the prediction [2]. There is no weather station outside the house, but the weather data from the nearest airport weather station that is Chievres Airport in Belgium is merged considering the same timestamps of in-house data to ensure both internal and external factors of the house contribute to the prediction [2].

II. Case Study

A. Problem Domain Description

The objective of this work is to carry out predictive analysis of the energy consumed by appliances in a low-energy house situated in Stambruges, Belgium. The idea is to establish a relationship between the appliance energy consumption and the predictors. The predictors are the attributes that influence the appliance energy consumed and in this paper, we plan to see how the internal, as well as external attributes, contribute to the prediction. Also, we deploy different algorithms to test our prediction and also analyze the performance of the models in terms of high accuracy and low error rates. The data set [1] is huge and consists of a total of 29 attributes, hence data filtering is required to eliminate the attributes that do not add any value to the prediction. The idea is to try different approaches for filtering the data and selecting the most relevant features in order to increase the performance of the model and reduce the dimensionality of the data set. There is a high possibility of many data entries exhibiting anomalous behavior, we plan to detect these outliers that might degrade the performance of the model. The major aim of the paper is to perform appropriate feature engineering and deploy distinct data prediction algorithms on the data set and analyze the performance and accuracy of the results. This shall also facilitate a comparison between different approaches and identify which algorithm is the most efficient.

B. Description of the Algorithms

We deployed algorithms from five different families of supervised machine learning to train and test our data and optimize the results. Below, we provide a brief description of these algorithms.

1) ***Linear****:* We used the Linear and the Support Vector Regressors to train our data set.

Linear Regression model establishes a relationship between the target variable that is the number of appliances and the residual components that influence the energy consumption in the low-energy house.

Support Vector Regression model works analogous to that of Support Vector machines except for the fact that in case of regression problems predicting real values becomes challenging and hence the model minimizes errors and maximizes margins and tries to comply to a certain level of tolerance.

2) ***Tree-Based****:* The algorithm that has been deployed to implement the tree-based regression is the Decision-Tree Regressor.

Decision Tree Regression We used the decision tree to build the regression model in the form of a tree structure where the data set is broken into smaller and smaller subsets and the final leaf nodes yield the decision in terms of predicting values.

3) ***Distance-Based****:* The algorithm that we used to implement the Distance-based regression is the KNeighbors algorithm.

The *K Neighbors Regression* model uses the 'k' nearest neighbors technique to predict the energy usage of the building. The model selects the optimal value of 'k' by plotting a graph between the Root Mean Squared Error [RMSE] and 'k' values. We select the optimal 'k' value where the RMSE evaluates to a minimum when deployed on the test data. We run the k Neighbors Regressor at this 'k' value and build the model.

4) ***Rule-Based****:* To implement the rule-based regression model, we put into practice the RuleFit algorithm.

RuleFit algorithm compensates for those interactions between the features that a linear regression model does not consider. The model learns these hidden interactions in the form of decision rules that are basically extracted from decision trees [28]. Each path of the tree is converted to a decision rule by summing up the split decisions together and generating a rule.

5) ***Ensemble****:* The ensemble models are used to provide an improvement in the performance as they combine multiple models together. We used three different ensemble regression models to the data set to observe any significant optimization of results.

Gradient Boosting Regressor [GBR] works with the motive of boosting results by means of optimizing a loss function. The GBR deploys decision trees as weak learners and uses the additive model component is to add more weight to the weak learners so the loss function can get minimized and the model learns better.

Bagging Regressor trains each regression model on a random subset of the data and aggregate the predictions by means of a majority voting mechanism. We deploy the Bagging Regressor to our data set where underneath each random sample of the data set predictions are made and are aggregated to yield results.

Random Forest Regressor combines different decision trees that are built using subsets of features. This method exploits all hidden combinations of decision trees and builds up a Random Forest model. The results are combined using voting or averaging. We use the RandomForestRegressor to the data set and analyze the results obtained.

III. Experimental Setup

A. Data set Description

The data set describes the energy consumed by the appliances in a low energy house. The data set contains details of the energy consumed in Watt hour [Wh] per 10 minutes in a

house for approximately 4.5 months starting from January 11, 2016 until May 27, 2016. A Zigbee wireless sensor network [13] is utilized to monitor the humidity and temperatures across all the rooms of the house. The weather from the nearby airport weather station, Chievres Airport, Belgium is also considered while calculating the energy consumed by the appliances. This data has been extracted from a public data set Reliable Prognosis, and has been merged together with the experimental data based on time intervals. The data consists of a total of 29 attributes that we subdivide into five broad categories:

1) Timestamp (T): date and time: year-month-day hour:minute: second

2) Appliances (y): energy use in Wh

3) The parameters inside the house (X-inside): These are the parameters inside the low energy house that contribute to the energy consumption of the appliances. These comprise of 19 attributes that are the energy consumed by light fixtures, the magnitude of temperature and humidity recorded in 8 different rooms of the house and also right outside the building.

4) The parameters outside the house (X-outside): These are the parameters outside the house, near the airport station that contribute to the energy consumption of the appliances. There are 6 different parameters that comprise the X-outside category. They are the temperature and humidity measures at the weather station, pressure, visibility, dew point temperature and the wind speed.

5) Random variables (X-rv): These are 2 non-dimensional random variables that are introduced to perform testing on the data set and help extract the parameters that do not significantly contribute to the prediction.

B. Data Pre-processing

From data set description provided, we notice that it consists of attributes that vary significantly in terms of their measurement units and functionalities. Before using the data to train the models and potentially utilize the same for predicting energy use, we require pre-processing of the data.

1) Data Normalization: The data utilizes multiple variables that contribute to the prediction of energy, but most of these values have different measurement units and hence they need to be scaled so that our results are optimal and not biased. The temperature, humidity, energy use, pressure, wind speed, visibility all are distinct parameters that contribute to the prediction and hence we would want to normalize the data set. The Z Score normalization seems to be the best choice for normalizing our data since it accurately transforms the data to maintain a mean value of 0 and a standard deviation of 1 [29]. Such kind of normalization is the most appropriate with data sets that have multiple measurement units. We import the StandardScalar() method from the sklearn.preprocessing library and effectively scale our data to make it ready for analysis. Post scaling, our data set gets transformed from Figure 1 to Figure 2 as below:

	date	Appliances	lights	T1	RH_1	T2	RH_2	T3	RH_3	T4	...	T9	RH_9	T_out	Press_mm_hg	RH_out	Windspee
0	2016-01-11 17:00:00	60	30	19.89	47.596667	19.2	44.790000	19.79	44.730000	19.000000	...	17.033333	45.53	6.60	733.5	92.0	7.00000
1	2016-01-11 17:10:00	60	30	19.89	46.693333	19.2	44.722500	19.79	44.790000	19.000000	...	17.066667	45.56	6.48	733.6	92.0	6.66666
2	2016-01-11 17:20:00	50	30	19.89	46.300000	19.2	44.626667	19.79	44.933333	18.926667	...	17.000000	45.50	6.37	733.7	92.0	6.33333
3	2016-01-11 17:30:00	50	40	19.89	46.066667	19.2	44.590000	19.79	45.000000	18.890000	...	17.000000	45.40	6.25	733.8	92.0	6.00000
4	2016-01-11 17:40:00	60	40	19.89	46.333333	19.2	44.530000	19.79	45.000000	18.890000	...	17.000000	45.40	6.13	733.9	92.0	5.66666

Fig. 1. Energy data set before normalization

	date	Day	Appliances	lights	T1	RH_1	T2	RH_2	T3	RH_3	...	RH_9	T_out	Press_mm_hg	RH_out
0	2016-01-11 17:00:00	Monday	-0.367676	3.301264	-1.118645	1.843821	-0.520411	1.073683	-1.235063	1.686130	...	0.958136	-0.152788	-2.976328	0.82208
1	2016-01-11 17:10:00	Monday	-0.367676	3.301264	-1.118645	1.616807	-0.520411	1.057097	-1.235063	1.704566	...	0.965363	-0.175352	-2.962813	0.82208
2	2016-01-11 17:20:00	Monday	-0.465215	3.301264	-1.118645	1.517959	-0.520411	1.033550	-1.235063	1.748608	...	0.950910	-0.196035	-2.949298	0.82208
3	2016-01-11 17:30:00	Monday	-0.465215	4.561378	-1.118645	1.459321	-0.520411	1.024540	-1.235063	1.769092	...	0.926821	-0.218599	-2.935783	0.82208
4	2016-01-11 17:40:00	Monday	-0.367676	4.561378	-1.118645	1.526336	-0.520411	1.009797	-1.235063	1.769092	...	0.926821	-0.241162	-2.922268	0.82208

Fig. 2. Energy data set after Z score normalization

The predictive models built with normalized data after applying the regression algorithms from five different families are shown in figure 3:

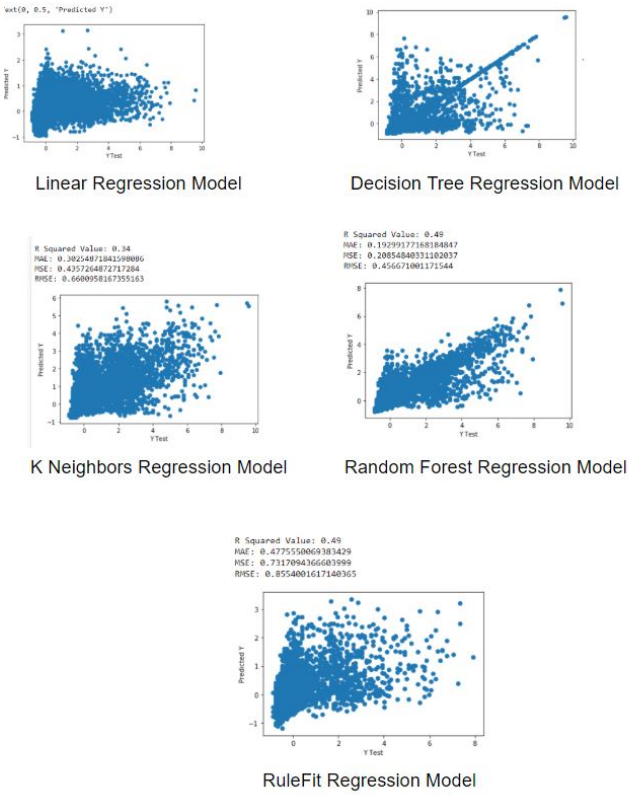

Fig. 3. Regression models built on normalized data

2) Feature Correlation: The data set comprises of quite a few attributes that represent the temperature and humidity of the rooms, there is a possibility of features being co-related. We evaluate the correlation between the features by using the .corr functionality of Pandas and represent the same via a Heat

map as shown in Figure 4. The intensity of the redness of the parameters as marked on the scale on the right shows how closely the features are correlated with each other. The variables "T6" and "T-out" seem to be highly correlated as per the heat-map. This interpretation makes sense because "T6" is the temperature outside the house and "T-out" is the temperature outside the nearby weather station. There is a high possibility of these two temperatures being correlated. We might want to consider only one of these features to eliminate any overfitting probabilities in our data set.

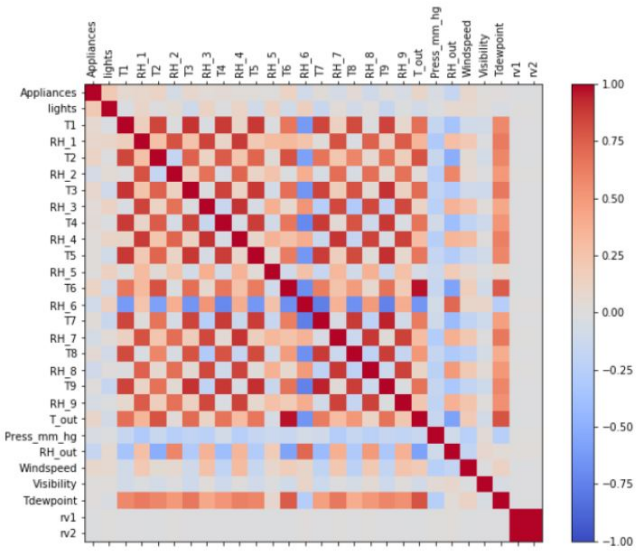

Fig. 4. Feature Correlation Heatmap

3) Feature Importance: In a data set like ours, it is extremely important to determine which features are most important in making predictions. We determine the feature importance using the XgBoost regressor that calculates the f scores and determines the most important features. We select the top 20 features from the Feature importance graph as shown in Figure 5.

C. Feature Selection

We realize that out of the 29 features listed in Section III (A), not all features significantly contribute to the prediction of energy usage in the house. Hence, the next step is to determine the significant features that actually make a difference while evaluating our data. Feature selection is an effective data pre-processing mechanism that is commonly used on high dimensional data. This technique of filtering out features helps in eliminating redundant features and improvises the predictive accuracy of the data set.

The three methods used for filtering out features are the Feature Importance Correlation, Lasso Regression technique, and the Boruta Recursive Feature Elimination [RFE] technique.

1) Feature Importance-Correlation: We combine the results of the results obtained from the Feature Correlation and Feature Importance analysis and select the top 20 features to train our model as shown in figure 6.

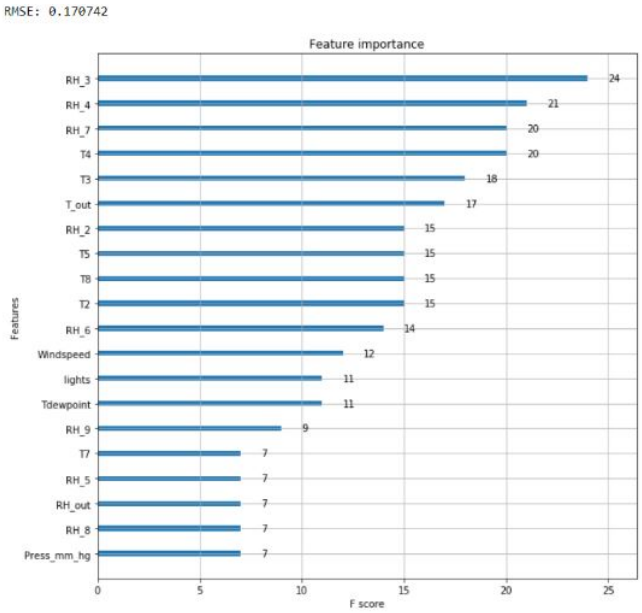

Fig. 5. Feature importance graph based on F-scores

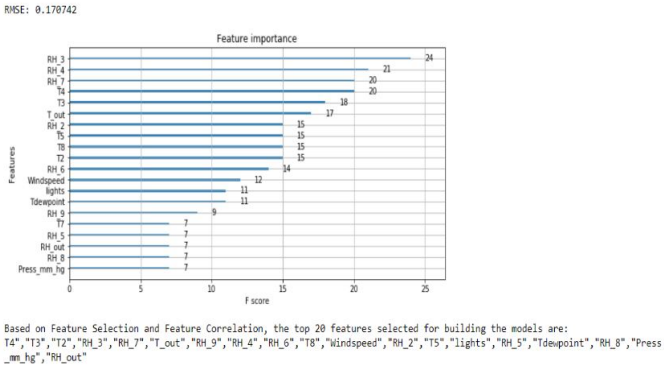

Fig. 6. Feature selection using Feature importance correlation

The regression models obtained after applying the Feature Selection based on feature importance and correlation is shown in figure 7.

2) Lasso Coefficients: The Lasso is an effective approach to deal with multicollinearity to improve the prediction performance. Multicollinearity is a property by virtue of which independent variables exhibit a very strong correlation [16]. The lasso regression technique tries to shrink the data values towards a central point, for instance, a mean. We implement the Lasso regressor to calculate the lasso coefficients with 10 fold cross-validation and use the features having non zero coefficients to train our model. The non-zero Lasso Coefficients obtained and the top 7 features extracted on the basis of these coefficients are shown in figure 8.

After performing Lasso feature selection, models appear as shown in figure 9.

3) Boruta Feature Selection: The Boruta Feature selection algorithm is a wrapper that works around the Random Forests

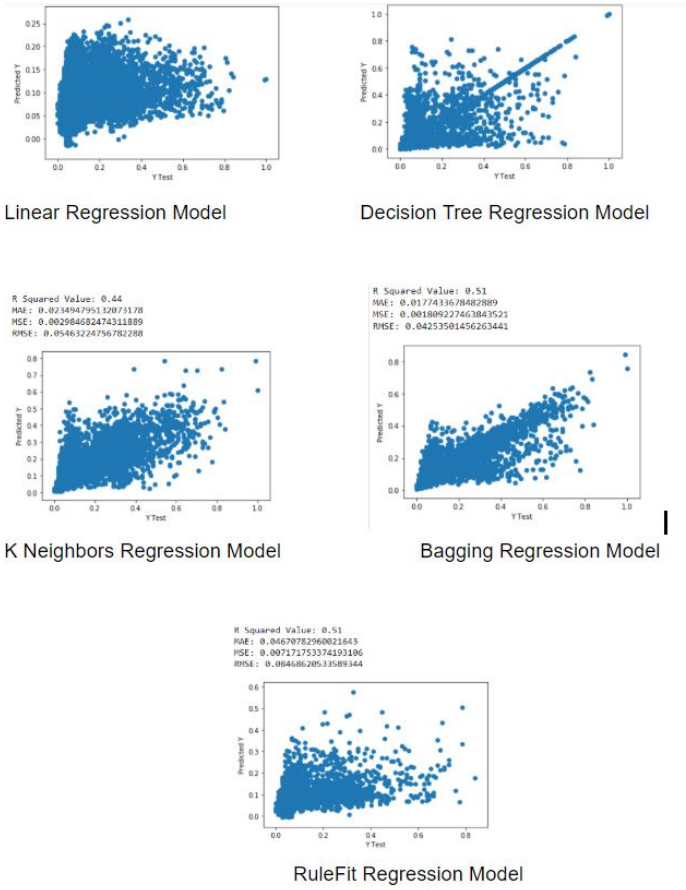

Fig. 7. Models after performing feature selection using Feature importance correlation

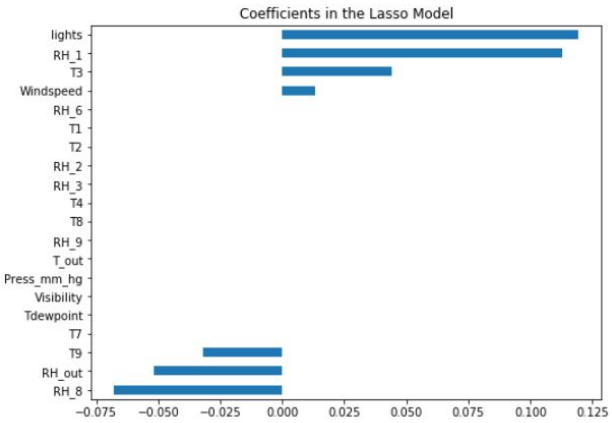

Fig. 8. Feature selection using Lasso Coefficients

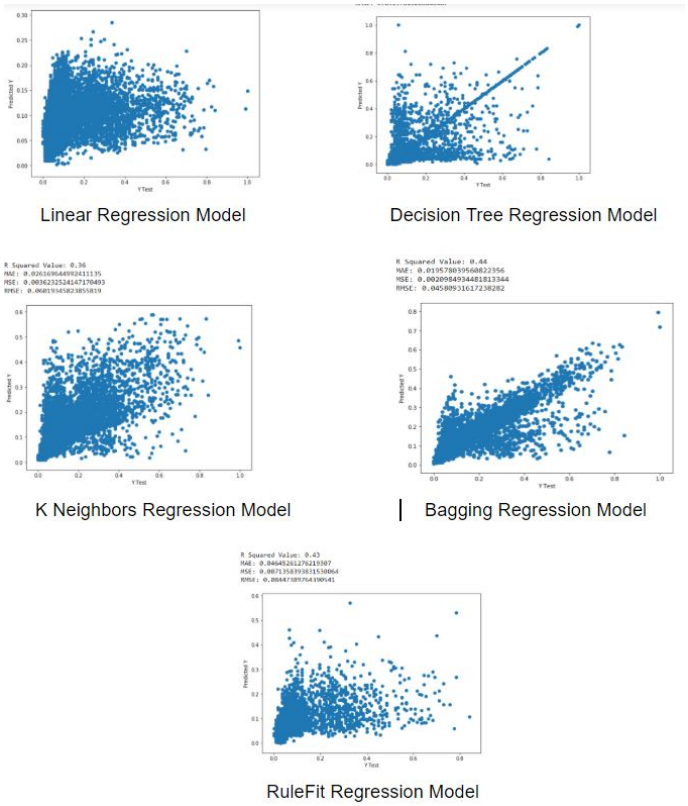

Fig. 9. Regression Models after Lasso Feature selection

and removes the attributes iteratively. This algorithm yields highly accurate results as it eliminates the attribute correlations by shuffling responses, unimportant features are recursively identified [15] and removed and all calculations are done using Maximum Z score among shadow attributes [MZSA] and assigns an edge to the attributes that score higher than the MZSA. After 13 iterations, the Boruta mechanism rejects 4 of the attributes as unimportant as shown in Figure 10.

After performing Boruta Feature selection on the data set, the regression models appear as shown in Figure 11.

D. Feature Extraction

We also wanted to perform feature extraction on our models to test how they perform when their dimensions are reduced [18]. To accomplish this, the *Principle Component Analysis* method was deployed. We try to determine the number of components that would help preserve the maximum variance of the data [19] as shown in Figure 12. The numbers depicted in the braces are a measure of the maximum variance preserved across the 27 attributes from the X-inside, X-outside and X-rv categories described in Section III(A). From this graph, we note that at n=17 the maximum variance seems to be preserved and hence we plan to reduce our 27 dimension model to 17 dimensions using PCA and then run the algorithms.

The regression models after performing feature extraction using Principle Component Analysis are shown in figure 13.

```
Iteration:      1 / 100
Confirmed:      0
Tentative:      27
Rejected:       0
Iteration:      2 / 100
Confirmed:      0
Tentative:      27
Rejected:       0
Iteration:      3 / 100
Confirmed:      0
Tentative:      27
Rejected:       0
Iteration:      4 / 100
Confirmed:      0
Tentative:      27
Rejected:       0
Iteration:      5 / 100
Confirmed:      0
Tentative:      27
Rejected:       0
Iteration:      6 / 100
Confirmed:      0
Tentative:      27
Rejected:       0
Iteration:      7 / 100
Confirmed:      0
Tentative:      27
Rejected:       0

Iteration:      7 / 100
Confirmed:      0
Tentative:      27
Rejected:       0
Iteration:      8 / 100
Confirmed:      22
Tentative:      1
Rejected:       4
Iteration:      9 / 100
Confirmed:      22
Tentative:      1
Rejected:       4
Iteration:      10 / 100
Confirmed:      22
Tentative:      1
Rejected:       4
Iteration:      11 / 100
Confirmed:      22
Tentative:      1
Rejected:       4
Iteration:      12 / 100
Confirmed:      23
Tentative:      0
Rejected:       4

BorutaPy finished running.

Iteration:      13 / 100
Confirmed:      23
Tentative:      0
Rejected:       4
[ True  True  True  True  True  True  True  True  True  True  True  True
  True  True  True  True  True  True  True False  True  True  True False
  True False False]

['lights' 'T1' 'RH_1' 'T2' 'RH_2' 'T3' 'RH_3' 'T4' 'RH_4' 'T5' 'RH_5' 'T6'
 'RH_6' 'T7' 'RH_7' 'T8' 'RH_8' 'T9' 'RH_9' 'Press_mm_hg' 'RH_out'
 'Windspeed' 'Tdewpoint']
```

Fig. 10. Boruta Feature Selection

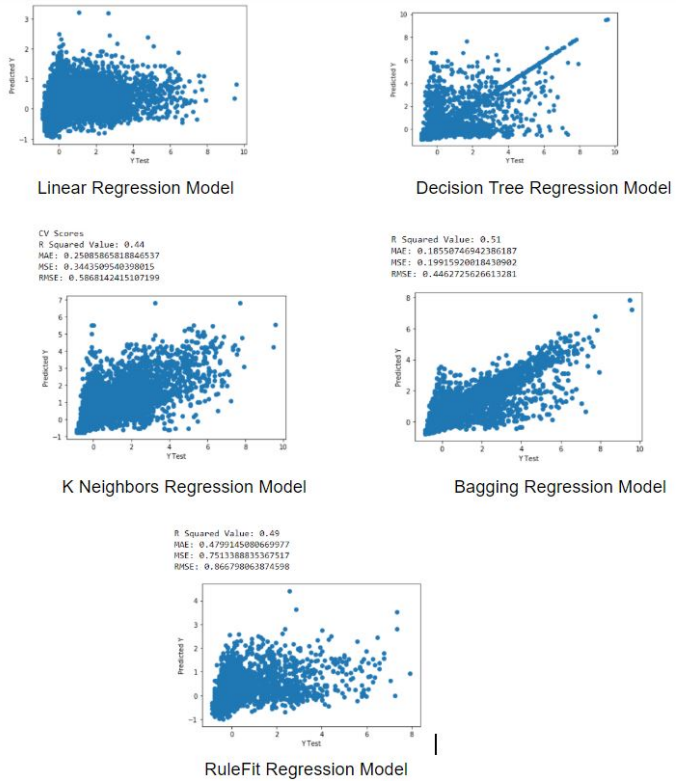

Fig. 11. Regression Models after Boruta Feature selection

```
[36.43 57.53 74.93 81.14 84.56 87.74 90.03 91.73 93.18 94.58 95.63 96.39
 97.03 97.58 98.02 98.44 98.8  99.1  99.33 99.5  99.64 99.76 99.86 99.94
 99.97 99.98 99.98]
```

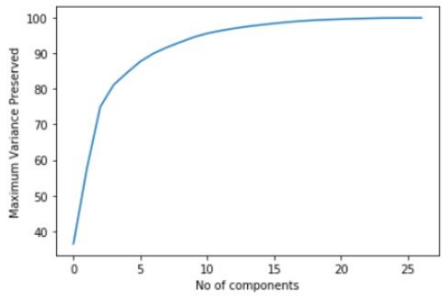

Fig. 12. Feature Extraction using PCA

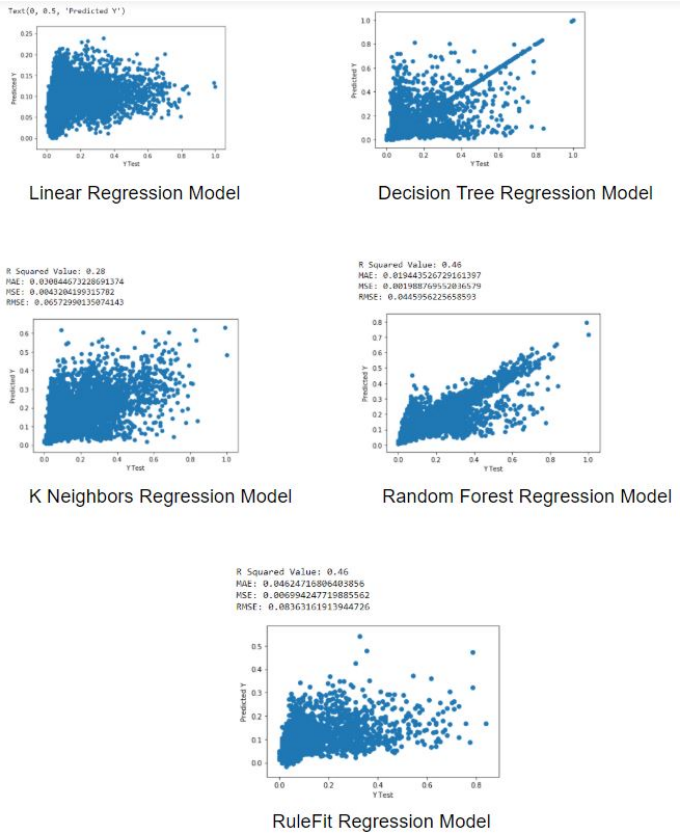

Fig. 13. Regression Models after PCA Feature Extraction

E. Outlier Detection in the Data set

We treat the data set as a potential time-series problem and try to predict the timestamps that exhibit anomalous behaviour. [25] The data is considered on a per-month basis for the months January 2016 to May 2016 and we determine those timestamps from the dataset that turn out to be outliers. As a part of the initial analysis we draw the boxplot of the data set for individual months to visualize the attributes that are potential candidates to be anomalies [21]. The figure 14 shows the boxplot of data set for January. From the figure, it is evident that some entries of the attributes like energy

consumed by lights, humidity in the kitchen RH1, temperature and pressure in the bathroom T5 and RH5 differ from the inter-quartile range and could be an outlier.

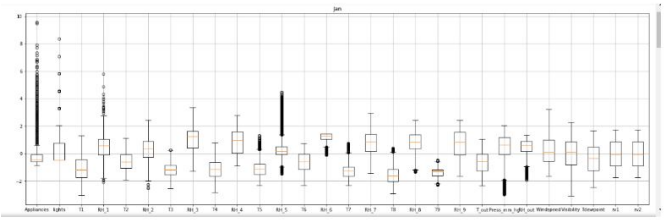

Fig. 14. Box plot of the data set for January

We used five different algorithms for detecting the outliers and selected the timestamps that are reported by more than three algorithms as potential candidates for being an outlier in the dataset.

1) Elliptic Envelope: We used the elliptic envelope algorithm which internally assumes the data to be normally distributed and considers the data around an ellipse and classifies all the observations that lie outside the ellipse as an outlier [24]. The statistics that were observed after using this technique is shown in Figure 15.

ellipticEnvelope

	MONTH	DATASIZE	NORMALS	OUTLIERS	OUTLIER_RATE
0	Jan	2922	2892	30	1.03
1	Feb	4176	4134	42	1.01
2	Mar	4464	4419	45	1.01
3	Apr	4320	4276	44	1.02
4	May	3853	3814	39	1.01

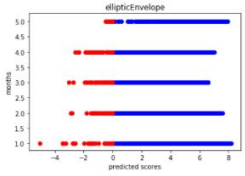

Fig. 15. Outliers observed using Elliptic Envelope

2) Gaussian Mixture: The Gaussian Mixture algorithm predicts the data points that do not lie in the area of one of the k Gaussian distributions as outliers [6]. Clusters do not take responsibilities of these outlying data points and hence get identified as outliers. The statistics after deploying the Gaussian mixture algorithm on our data set is shown in Figure 16.

gaussianMixture

	MONTH	DATASIZE	NORMALS	OUTLIERS	OUTLIER_RATE
0	Jan	2922	1813	1109	37.95
1	Feb	4176	2665	1511	36.18
2	Mar	4464	2695	1769	39.63
3	Apr	4320	2396	1924	44.54
4	May	3853	2435	1418	36.8

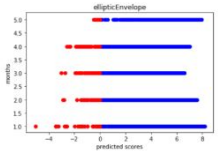

Fig. 16. Outliers observed using Gaussian Mixture

3) Local Outlier Factor: We used the local outlier factor mechanism to test for outliers that computes the local density deviation of the data point with respect to its neighbors [26]. The samples having lower density when compared to its neighbors are selected as outliers. The statistics for outlier detection using Local Outlier Factor is shown in Figure 17.

localOutlierFactor

	MONTH	DATASIZE	NORMALS	OUTLIERS	OUTLIER_RATE
0	Jan	2922	2901	21	0.72
1	Feb	4176	4151	25	0.6
2	Mar	4464	4431	33	0.74
3	Apr	4320	4293	27	0.62
4	May	3853	3825	28	0.73

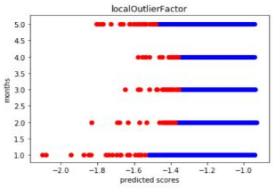

Fig. 17. Outliers observed using Local Outlier Factor

4) Isolation Forest: The isolation forest isolates outliers by randomly selecting features and identifying a potential split value that lies between the maximum and minimum values that the selected feature can have [22]. The statistics for the Isolation Forest outlier detection is shown in Figure 18.

isolationForest

	MONTH	DATASIZE	NORMALS	OUTLIERS	OUTLIER_RATE
0	Jan	2922	2892	30	1.03
1	Feb	4176	4134	42	1.01
2	Mar	4464	4419	45	1.01
3	Apr	4320	4276	44	1.02
4	May	3853	3814	39	1.01

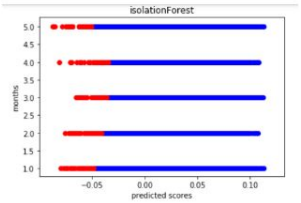

Fig. 18. Outliers observed using Isolation Forests

5) One Class SVM: One Class SVM technique learns the test data boundaries using a decision function that is built based on the data set and classifies the data points outside these boundaries as outliers [23]. The statistics for the same are depicted in Figure 19.

oneClassSVM

	MONTH	DATASIZE	NORMALS	OUTLIERS	OUTLIER_RATE
0	Jan	2922	2889	33	1.13
1	Feb	4176	4133	43	1.03
2	Mar	4464	4419	45	1.01
3	Apr	4320	4275	45	1.04
4	May	3853	3809	44	1.14

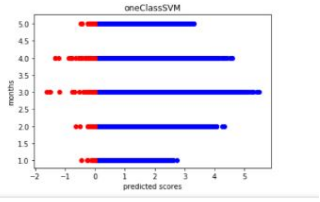

Fig. 19. Outliers observed using One Class SVM

IV. Evaluation

A. Evaluation Criteria

The evaluation of the results are based on the three parameters that are the R-squared value, MAE, and RMSE [17].

1) R-Squared [R2] value: The comparison of the model to the baseline and takes values between -infinity and 1. We basically take an average of all points imagining a horizontal line passing through these points. The higher the R2 value, the more accurate is our model's resemblance to the baseline.

2) Mean Absolute Error [MAE]: Mean Squared Error calculates error as an average of the absolute differences that are obtained between the actual values and predicted values. MAE is also less sensitive to outliers, hence we use it for the evaluation purpose of our data set as it would make the results more robust. The lesser the MAE value, the higher the accuracy of the model.

3) Root Mean Square Error [RMSE]: Root Mean Squared Error is the square root of Mean Squared Error and measures the average squared errors between the predicted and target values. The fact that errors are squared before taking their average gives a potentially high weight to the larger errors and penalizes them more [11]. Looking at our models, there is a possibility that some of the values are just unexpected and may not be outliers, hence we would want to use RMSE as an evaluation criterion. A low value of RMSE is desired for the model to exhibit optimal performance [30].

B. Evaluation of Results

We use three methods to evaluate our results: the train-test split, 10 fold cross validation, 10 fold shuffle split cross validation and the statistical paired-t test [20] to evaluate whether the algorithms have a statistically significant difference between them.

1) Train-Test Split: Using the train test split validation raw data, the Bagging and the Random Forest regression models showed the best accuracy and least error rates as shown in figure 20.

Model	Linear	Tree-Based	Distance-Based	Rule-Based	Ensemble
Without feature selection	***Linear Regressor:*** R2: **0.16** MAE: **0.524** RMSE:**0.932** ***Support Vector Regressor*** R2: **1.00** MAE: **0.384** RMSE:**0.908**	***DecisionTree Regressor*** R2: **0.13** MAE: **0.404** RMSE: **0.948**	***KNeighbors Regressor:*** R2: **0.30** MAE: **0.406** RMSE: **0.849**	***RuleFit:*** R2: **0.49** MAE: **0.485** RMSE: **0.876**	***GradientBoosting Regressor*** R2: **0.25** MAE: **0.475** RMSE: **0.882** ***Bagging Regressor:*** R2: **0.49** MAE: **0.347** RMSE: **0.723** ***RandomForest Regressor:*** R2: **0.49** MAE: **0.351** RMSE: **0.728**

Fig. 20. Train test split results on raw data

With Feature Importance-Correlation feature selection technique, the RuleFit algorithm performs the best it exhibits an R2 value of 0.51 and has very low error scores as shown in Figure 21.

The Lasso feature selection technique exhibited the best results for the Bagging and Random Forest algorithm as shown in figure 22. It showed the least error rates as compared to any other feature selection techniques.

Feature Importance - Correlation Feature S election	***Linear Regressor:*** R2: **0.12** MAE: **0.051** RMSE:**0.091** ***Support Vector Regressor*** R2: **1.00** MAE: **0.084** RMSE:**0.102**	***DecisionTree Regressor*** R2: **0.16** MAE: **0.038** RMSE: **0.089**	***KNeighbors Regressor:*** R2: **0.46** MAE: **0.032** RMSE: **0.0714**	***RuleFit:*** R2: **0.51** MAE:**0.0001** RMSE: **0.0007**	***GradientBoosting Regressor*** R2: **0.23** MAE: **0.046** RMSE: **0.085** ***Bagging Regressor:*** R2: **0.51** MAE: **0.032** RMSE: **0.068** ***RandomForest Regressor:*** R2: **0.51** MAE: **0.032** RMSE: **0.068**

Fig. 21. Train test split results with Feature Importance-Correlation

Lasso Feature Selection	***Linear Regressor:*** R2: **0.13** MAE:**0.051** RMSE:**0.090** ***Support Vector Regressor*** R2: **1.00** MAE: **0.083** RMSE:**0.102**	***DecisionTree Regressor*** R2: **0.05** MAE: **0.040** RMSE:**0.0948**	***KNeighbors Regressor:*** R2:**0.36** MAE: **0.036** RMSE: **0.078**	***RuleFit:*** R2:**0.43** MAE: **0.046** RMSE: **0.084**	***GradientBoosting Regressor*** R2: **0.21** MAE: **0.0471** RMSE: **0.086** ***Bagging Regressor*** R2: **0.43** MAE: **0.035** RMSE: **0.073** ***RandomForest Regressor:*** R2: **0.43** MAE: **0.035** RMSE: **0.073**

Fig. 22. Train test split results with Lasso Feature selection

The Boruta algorithm identified the Bagging algorithm that exhibited high R2 and low error values as shown in figure 23.

Boruta Feature Selection	***Linear Regressor:*** R2: **0.16** MAE: **0.524** RMSE:**0.933** ***Support Vector Regressor*** R2: **1.00** MAE: **0.378** RMSE:**0.898**	***DecisionTree Regressor*** R2: **0.13** MAE: **0.406** RMSE: **0.949**	***KNeighbors Regressor:*** R2: **0.43** MAE: **0.344** RMSE: **0.765**	***RuleFit:*** R2: **0.50** MAE: **0.479** RMSE: **0.859**	***GradientBoosting Regressor*** R2: **0.25** MAE: **0.475** RMSE: **0.881** ***Bagging Regressor*** R2: **0.51** MAE: **0.338** RMSE: **0.715** ***RandomForest Regressor:*** R2: **0.50** MAE: **0.341** RMSE: **0.717**

Fig. 23. Train test split results with Boruta feature selection

On performing Feature extraction using PCA, the train test split cross validation yielded best results for the Random Forest and Bagging regression models as shown in Figure 24.

2) 10 fold cross validation: Using 10 fold cross validation on raw data, the Rule Based Regression, Support Vector Regression and the Gradient Boosting Regression models showed a positive R2 and low RMSE and MAE. The Rule Based model shows the highest R2 and low RMSE as shown in figure 25.

With Feature Importance-Correlation feature selection technique, the RuleFit algorithm performs the best it exhibits an

PCA	*Linear Regressor:* R2: **0.11** MAE: **0.52** RMSE:**0.092** *Support Vector Regressor* R2: **1.00** MAE: **0.084** RMSE:**0.102**	*DecisionTree Regressor* R2: **0.02** MAE: **0.043** RMSE:**0.0968**	*KNeighbors Regressor:* R2: **0.26** MAE: **0.0411** RMSE:**0.084**	*RuleFit:* R2: **0.46** MAE: **0.046** RMSE: **0.083**	*GradientBoosting Regressor* R2: **0.23** MAE: **0.046** RMSE: **0.085** *Bagging Regressor:* R2: **0.46** MAE: **0.0355** RMSE: **0.071** *RandomForest Regressor:* R2: **0.46** MAE:**0.0357** RMSE: **0.0717**

Fig. 24. Train test split results with PCA

Model	Linear	Tree-Based	Distance-Based	Rule-Based	Ensemble
Without feature selection	*Linear Regressor:* R2: **0.11** MAE: **0.512** RMSE:**0.914** *Support Vector Regressor* R2: **0.09** MAE: **0.356** RMSE:**0.873**	*DecisionTree Regressor* R2: **-1.86** MAE: **0.1213** RMSE: **0.519**	*KNeighbors Regressor:* R2: **-0.36** MAE: **0.302** RMSE: **0.660**	*RuleFit:* R2: **0.49** MAE: **0.485** RMSE: **0.876**	*GradientBoosting Regressor* R2: **0.01** MAE: **0.445** RMSE: **0.817** *Bagging Regressor:* R2: **-0.30** MAE: **0.189** RMSE: **0.4513** *RandomForest Regressor:* R2: **-0.31** MAE: **0.192** RMSE: **0.456**

Fig. 25. 10 fold Cross validation results on raw data

R2 value of 0.51 and has very low error scores as shown in Figure 26.

Feature Importance-Correlation Feature Selection	*Linear Regressor:* R2: **0.07** MAE: **0.05** RMSE: **0.08** *Support Vector Regressor* R2: **-0.17** MAE: **0.083** RMSE: **0.1011**	*DecisionTree Regressor* R2: **-2.19** MAE: **0.015** RMSE: **0.04**	*KNeighbors Regressor:* R2: **-0.73** MAE: **0.023** RMSE: **0.054**	*RuleFit:* R2: **0.51** MAE:**0.0001** RMSE: 0.0007	*GradientBoosting Regressor* R2: **-0.04** MAE: **0.043** RMSE: **0.079** *Bagging Regressor:* R2: **-0.36** MAE: **0.017** RMSE: **0.042** *RandomForest Regressor:* R2: **-0.37** MAE: **0.018** RMSE: **0.043**

Fig. 26. 10 fold Cross validation results with Feature Importance-Correlation

The Lasso feature selection technique exhibited the best results for the RuleFit algorithm as shown in figure 27.

The Boruta algorithm identified the RuleFit algorithm that exhibited high R2 and low error values as shown in figure 28.

On performing Feature extraction using PCA, 10 fold cross validation yielded the best results for the RuleFit model as shown in Figure 29.

3) 10 fold shuffle split: Using 10 fold shuffle split cross validation on raw data, the Rule Based Regression, Support Vector Regression and the Gradient Boosting Regression models showed a positive R2 and low RMSE and MAE. The

Lasso Feature Selection	*Linear Regressor:* R2: **0.11** MAE: **0.0502** RMSE:**0.089** *Support Vector Regressor* R2: **-0.17** MAE: **0.083** RMSE:**0.101**	*DecisionTree Regressor* R2: **-2.23** MAE: **0.0122** RMSE: **0.051**	*KNeighbors Regressor:* R2:**-0.39** MAE: **0.0261** RMSE: **0.060**	*RuleFit:* R2:**0.43** MAE: **0.046** RMSE: **0.084**	*GradientBoosting Regressor* R2: **0.04** MAE: **0.044** RMSE: **0.081** *Bagging Regressor:* R2: **-0.25** MAE:**0.0195** RMSE: **0.045** *RandomForest Regressor:* R2: **-0.27** MAE: **0.0197** RMSE: **0.046**

Fig. 27. 10 fold Cross validation results with Lasso Feature selection

Boruta Feature Selection	*Linear Regressor:* R2: **0.11** MAE: **0.511** RMSE: **0.915** *Support Vector Regressor* R2: **0.10** MAE: **0.354** RMSE:**0.867**	*DecisionTree Regressor* R2: **-2.21** MAE: **0.122** RMSE: **0.520**	*KNeighbors Regressor:* R2: **-0.72** MAE: **0.250** RMSE: **0.586**	*RuleFit:* R2: **0.50** MAE: **0.479** RMSE: **0.859**	*GradientBoosting Regressor* R2: **0.01** MAE: **0.445** RMSE: **0.818** *Bagging Regressor:* R2: **-0.30** MAE: **0.1855** RMSE: **0.446** *RandomForest Regressor:* R2: **-0.32** MAE: **0.1878** RMSE: **0.4506**

Fig. 28. 10 fold Cross validation results with Boruta feature selection

Gradient Boosting Regression model shows the highest R2 and low RMSE as shown in Figure 30.

On performing feature selection based on the Feature Importance-Correlation, the RuleFit and the Random Forest Ensemble regressor exhibit the best performance as shown in Figure 31.

As per our analysis, the Bagging and the Random Forest Regression models showed optimal results with the Lasso feature selection as shown in Figure 32.

The Boruta algorithm identified the Random Forest algorithm that exhibited the best results as shown in Figure 33.

On performing Feature extraction using PCA, 10 fold shuffle split cross validation yielded the best results for the RuleFit

PCA	*Linear Regressor:* R2: **0.07** MAE: **0.512** RMSE:**0.902** *Support Vector Regressor* R2: **-0.18** MAE: **0.083** RMSE:**0.100**	*DecisionTree Regressor* R2: **-1.55** MAE: **0.013** RMSE: **0.053**	*KNeighbors Regressor:* R2: **-0.36** MAE: **0.0308** RMSE:**0.065**	*RuleFit:* R2: **0.46** MAE: **0.046** RMSE: **0.083**	*GradientBoosting Regressor* R2: **0.06** MAE: **0.043** RMSE: **0.079** *Bagging Regressor:* R2: **-0.12** MAE: **0.019** RMSE: **0.044** *RandomForest Regressor:* R2: **-0.13** MAE:**0.019** RMSE: **0.045**

Fig. 29. 10 fold Cross validation results with PCA

Model	Linear	Tree-Based	Distance-Based	Rule-Based	Ensemble
Without feature selection	***Linear Regressor:*** R2: **0.16** MAE: **0.512** RMSE:**0.914** ***Support Vector Regressor*** R2: **0.22** MAE: **0.356** RMSE:**0.873**	***DecisionTree Regressor*** R2: **0.33** MAE: **0.121** RMSE: **0.519**	***KNeighbors Regressor:*** R2: **0.26** MAE: **0.445** RMSE: **0.817**	***RuleFit:*** R2: **0.49** MAE: **0.485** RMSE: **0.876**	***GradientBoosting Regressor*** R2: **0.50** MAE: **0.189** RMSE: **0.451** ***Bagging Regressor:*** R2: **0.49** MAE: **0.192** RMSE: **0.456** ***RandomForest Regressor:*** R2: **0.49** MAE: **0.484** RMSE: **0.876**

Fig. 30. 10 fold shuffle split results on raw data

Feature Importance-Correlation Feature Selection	***Linear Regressor:*** R2: **0.12** MAE: **0.05** RMSE: **0.08** ***Support Vector Regressor*** R2: **-0.12** MAE: **0.083** RMSE:**0.101**	***DecisionTree Regressor*** R2: **0.10** MAE: **0.011** RMSE: **0.048**	***KNeighbors Regressor:*** R2: **0.43** MAE: **0.023** RMSE: **0.054**	***RuleFit:*** R2: **0.51** MAE:**0.0001** RMSE: **0.0007**	***GradientBoosting Regressor*** R2: **0.25** MAE: **0.043** RMSE: **0.079** ***Bagging Regressor:*** R2: **0.51** MAE: **0.017** RMSE: **0.042** ***RandomForest Regressor:*** R2: **0.51** MAE: **0.018** RMSE: **0.043**

Fig. 31. 10 fold shuffle split results with Feature Importance-Correlation

Lasso Feature Selection	***Linear Regressor:*** R2: **0.13** MAE:**0.050** RMSE:**0.089** ***Support Vector Regressor*** R2: **-0.12** MAE: **0.083** RMSE:**0.101**	***DecisionTree Regressor*** R2: **-0.02** MAE: **0.0122** RMSE: **0.051**	***KNeighbors Regressor:*** R2: **0.36** MAE: **0.0261** RMSE: **0.060**	***RuleFit:*** R2:**0.43** MAE: **0.0464** RMSE: **0.0844**	***GradientBoosting Regressor*** R2:**0.22** MAE: **0.044** RMSE: **0.081** ***Bagging Regressor:*** R2: **0.44** MAE: **0.0195** RMSE: **0.0458** ***RandomForest Regressor:*** R2: **0.44** MAE: **0.0197** RMSE: **0.0463**

Fig. 32. 10 fold shuffle split results with Lasso Feature selection

Boruta Feature Selection	***Linear Regressor:*** R2: **0.16** MAE: **0.511** RMSE:**0.915** ***Support Vector Regressor*** R2: **0.23** MAE: **0.354** RMSE:**0.867**	***DecisionTree Regressor*** R2: **0.11** MAE: **0.122** RMSE: **0.520**	***KNeighbors Regressor:*** R2: **0.42** MAE: **0.250** RMSE: **0.586**	***RuleFit:*** R2: **0.50** MAE: **0.479** RMSE: **0.859**	***GradientBoosting Regressor*** R2: **0.51** MAE: **0.1855** RMSE: **0.446** ***Bagging Regressor:*** R2: **0.51** MAE: **0.1855** RMSE: **0.446** ***RandomForest Regressor:*** R2: **0.50** MAE: **0.187** RMSE: **0.450**

Fig. 33. 10 fold shuffle split results with Boruta Feature selection

and the Random Forest models as shown in Figure 34.

PCA	***Linear Regressor:*** R2: **0.11** MAE: **0.512** RMSE:**0.902** ***Support Vector Regressor*** R2: **-0.12** MAE: **0.083** RMSE: **0.1009**	***DecisionTree Regressor*** R2: **-0.05** MAE: **0.013** RMSE: **0.053**	***KNeighbors Regressor:*** R2: **0.27** MAE: **0.0308** RMSE:**0.065**	***RuleFit:*** R2: **0.46** MAE: **0.046** RMSE: **0.083**	***GradientBoosting Regressor*** R2: **0.23** MAE: **0.043** RMSE: **0.079** ***Bagging Regressor:*** R2: **0.46** MAE: **0.019** RMSE: **0.044** ***RandomForest Regressor:*** R2: **0.46** MAE:**0.019** RMSE: **0.045**

Fig. 34. 10 fold shuffle split results with PCA

Based on the feature validations performed, it is evident that the best results obtained are from the Bagging, RuleFit and the Random Forest Regression models as shown in Figure 35.

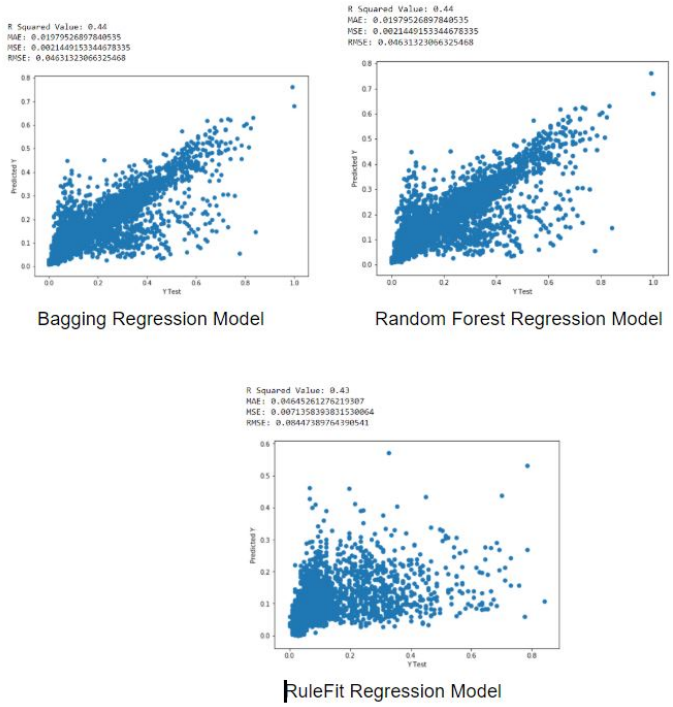

Fig. 35. Models showing optimal performance with Lasso Feature Selection

The Decision Tree Regression and Linear Regression models have exhibited poor performance compared to the other models. These models have been depicted in Figure 36.

4) Statistical paired-t test: We performed the statistical paired-t test [27] between the shuffle split cross validation scores obtained from the Random Forest(best performing), KNeighbors(average performing) and the Decision Tree(least performing) regression models. The CV scores of each of these models is shown in Table I.

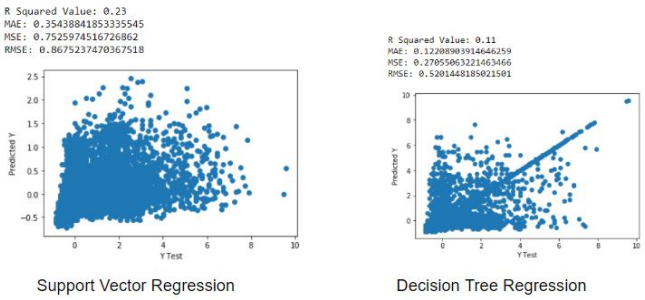

Fig. 36. Models showing poor performance with Lasso Feature Selection

TABLE I
CV SCORES FOR REGRESSION MODELS

Fold	Decision Tree	K Neighbors	Random Forest
1	0.156	0.425	0.503
2	0.115	0.379	0.497
3	0.107	0.442	0.521
4	0.136	4297	0.491
5	0.086	4297	0.478
6	0.061	4297	0.504
7	0.060	4297	0.533
8	0.048	4297	0.484
9	0.053	4297	0.515
10	0.211	3840	0.518

We calculate the difference between the CV scores and also the mean and standard deviations of the results from Table II.

TABLE II
DIFFERENCE BETWEEN CV SCORES FOR REGRESSION MODELS

Fold	RF-kNN	RF-DT	kNN-DT
1	0.269	0.347	0.078
2	0.264	0.382	0.118
3	0.335	0.414	0.079
4	0.285	0.355	0.07
5	0.324	0.392	0.068
6	0.359	0.443	0.084
7	0.402	0.473	0.071
8	0.384	0.436	0.052
9	0.386	0.462	0.076
10	0.21	0.307	0.097
Mean	0.329	0.401	0.079
SD	0.0632	0.0541	0.0178
t value	16.461	23.43	14.034
p value	0.001	0.001	0.001

Since the p-values of all the three algorithms are less than 0.05, we reject the hypothesis and say that there is a statistically significant difference between the three algorithms Decision Tree, K Neighbors and Random Forests.

C. Outlier Detection Results

We choose the timestamps that are selected by more than three of the outlier detection algorithms to be potential outliers. The detailed results are shown in table III.

TABLE III
OUTLIERS DETECTED FOR INDIVIDUAL MONTHS

Month	Data size	Normals	Outliers
January	2922	2903	19
February	4176	4151	25
March	4464	4435	29
April	4320	4297	23
May	3853	3840	13

With this data we identify the timestamps of each month that are outliers and list them out. A subset of the timestamps discovered as outliers for January is shown in Figure 15. It is observed that most of the timestamps that are outliers correspond to the evening timings between 1500 to 1800 hours. In the morning, the timings between 700 to 800 hours has the most outliers.

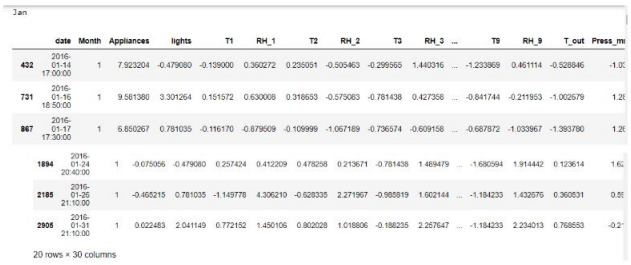

Fig. 37. Subset of timestamps identified as outliers for January

V. LIMITATIONS

The major limitation of this study is that it only takes into consideration a single house that is situated quite far from the weather station. Hence, the parameters near the weather station do not add much value to the energy usage prediction and this is a principal cause why most of the X-outside attributes were eliminated with feature selection techniques. Another challenge here is the fact that the data doesn't take into consideration details about the number of occupants, their age and lifestyle and the geometry of the house. We are also not sure how accurate is the Zigbee wireless sensor that captures the energy usage and whether it has been deployed in a way that yields the most accurate results. The data does not contain information about all the months and hence a seasonal evaluation of the energy consumption does not look feasible.

VI. CONCLUSION AND FUTURE WORK

With the statistical data analysis in section IV, we conclude that the Random Forest, Bagging Regression and the RuleFit algorithms have showcased the most favourable results. However, the Decision tree model seems to have a poor performance. Since Random Forests are built on random samples of training data of multiple single trees, they are considered more accurate than decision trees [31]. The K Neighbors Regression has also been decent as far as the results are concerned. The

Decision Tree and Linear Regression models exhibited the least performance while comparing to other models. Hence, the ensemble and rule based families of algorithms seem to have an upper edge in terms of performance and low error rates in this problem domain.

With Feature selection, Lasso builds highly accurate models with error rates being significantly reduced. With PCA as well, error rates have decreased when compared to the normalized data models without any feature selection. However, not much of a difference is observed in performance metrics with the Boruta feature elimination. The statistical paired-t test performed between the three algorithms - Random Forest, KNeighbors and Decision Tree shows that there is a statistically significant difference between the results obtained from these algorithms.Since k fold validation just divides the data into k folds while the k fold shuffle split iteratively samples the entire data set randomly to generate a training and testing set for each iteration. With our results, it is evident that the k-fold Shuffle split cross validation yields better results when compared to the k-fold cross validation. This can be correlated with the fact that the decision trees that consider the entire data set are also not yielding good results. This can mostly be because the data set seems to have quite a few outliers that have been detected in Section V and hence when the shuffling and random selection of features is being utilized to build or cross-validate the models, the results are better.

The parameters have been broadly classified into categories X-inside, X-outside and X-rv as described in Section III(A). In the X-inside category as per data provided by the wireless Zigbee sensor, the temperature and humidity in the kitchen, laundry, office, living room and bathroom were significant contributors, also the energy consumed by light fixtures had a vital role to play. Out of the attributes that constitute X-outside, the most contributing ones are the Pressure, windspeed, humidity and the dew point temperature, while the visibility and the temperature outside the weather station are not regarded as valuable by the feature selection algorithms. The X-rv category has random variables that do not contribute significantly building the models and are eliminated by most of the feature selection algorithms. The heat map shows the temperature outside the weather station and that outside the house to be highly correlated.

The data analysis has resulted in curious observations. Future work in this regard could be considering factors like precipitation and solar radiations as potential attributes to contribute to the prediction[62]. The information about the number of occupants in the house and the number of hours when the house is occupied would add valuable information for co-relating the energy usage with the home-occupancy. Treating the data set as a time series problem and predicting the energy usage in weekdays and weekends to observe any significant difference in the consumption could be an extension of the data modeling.

REFERENCES

[1] Gokagglers. (2017, September 16). Appliances Energy Prediction. Retrieved December 4, 2019, from https://www.kaggle.com/loveall/appliances-energy-prediction.

[2] Candanedo, L., Feldheim, V., Deramaix, D. (2017). Data driven prediction models of energy use of appliances in a low-energy house. Energy And Buildings, 140, 81-97. doi: 10.1016/j.enbuild.2017.01.083

[3] Kim, J.-Y., Cho, S.-B. (2019). Electric Energy Consumption Prediction by Deep Learning with State Explainable Autoencoder. Energies, 12(4), 739. doi: 10.3390/en12040739.

[4] Mosavi, A., Bahmani, A. (2019). Energy Consumption Prediction Using Machine Learning; A Review. doi:10.20944/preprints201903.0131.v1

[5] Kim, S., Jung, S., Baek, S.-M. (2019). A Model for Predicting Energy Usage Pattern Types with Energy Consumption Information According to the Behaviors of Single-Person Households in South Korea. Sustainability, 11(1), 245. doi: 10.3390/su11010245

[6] Reddy, A., Ordway-West, M., Lee, M., Dugan, M., Whitney, J., Kahana, R., Rao, M. (2017). Using Gaussian Mixture Models to Detect Outliers in Seasonal Univariate Network Traffic. 2017 IEEE Security and Privacy Workshops (SPW). doi:10.1109/spw.2017.9

[7] Seem, J. E. (2007). Using intelligent data analysis to detect abnormal energy consumption in buildings. Energy and Buildings, 39(1), 52-58. doi:10.1016/j.enbuild.2006.03.033

[8] Zhao, P., Suryanarayanan, S., Simoes, M. G. (2010). An Energy Management System for Building Structures Using a Multi-Agent Decision-Making Control Methodology. 2010 IEEE Industry Applications Society Annual Meeting. doi:10.1109/ias.2010.5615412

[9] Zhao, P., Suryanarayanan, S., Simoes, M. G. (2010). An Energy Management System for Building Structures Using a Multi-Agent Decision-Making Control Methodology. 2010 IEEE Industry Applications Society Annual Meeting. doi:10.1109/ias.2010.5615412

[10] World Energy Outlook 2016. (2016). World Energy Outlook. doi:10.1787/weo-2016-en

[11] Swalin, A. (2018, July 10). Choosing the Right Metric for Evaluating Machine Learning Models ? Part 1. Retrieved from https://medium.com/usf-msds/choosing-the-right-metric-for-machine-learning-models-part-1-a99d7d7414e4

[12] Fabi, V., Andersen, R. V., Corgnati, S., Olesen, B. W. (2012). Occupants' window opening behaviour: A literature review of factors influencing occupant behaviour and models. Building and Environment, 58, 188-198. doi:10.1016/j.buildenv.2012.07.009

[13] Liu, J., Hui, S. (2012). The study of ZigBee networking with wireless sensor. 2012 2nd International Conference on Consumer Electronics, Communications and Networks (CECNet). doi:10.1109/cecnet.2012.6202070

[14] Xie, J., Chen, Y., Hong, T., Laing, T. D. (2018). Relative Humidity for Load Forecasting Models. IEEE Transactions on Smart Grid, 9(1), 191-198. doi:10.1109/tsg.2016.2547964

[15] Zhang, C., Li, Y., Yu, Z., Tian, F. (2016). Feature selection of power system transient stability assessment based on random forest and recursive feature elimination. 2016 IEEE PES Asia-Pacific Power and Energy Engineering Conference (APPEEC). doi:10.1109/appeec.2016.7779696

[16] Dubey, A. (2019, February 4). Feature Selection Using Regularisation ? Retrieved from https://towardsdatascience.com/feature-selection-using-regularisation-a3678b71e499

[17] Assessing the Fit of Regression Models. (2018, May 9). Retrieved from https://www.theanalysisfactor.com/assessing-the-fit-of-regression-models/

[18] Roopa, H., Asha, T. (2019). A Linear Model Based on Principal Component Analysis for Disease Prediction. IEEE Access, 7, 105314-105318. doi:10.1109/access.2019.2931956

[19] VanderPlas, J. (n.d.). In Depth: Principal Component Analysis. Retrieved from https://jakevdp.github.io/PythonDataScienceHandbook/05.09-principal-component-analysis.html

[20] Yin, J. (2018, September 3). Understanding the data splitting functions in scikit-learn. Retrieved December 4, 2019, from https://medium.com/@julie.yin/understanding-the-data-splitting-functions-in-scikit-learn-9ae4046fbd26

[21] Nurunnabi, A., Nasser, M. (2009). Outlier Detection by Regression Diagnostics in Large Data. 2009 International Conference on Future Computer and Communication. doi:10.1109/icfcc.2009.60

[22] Lewinson, E. (2019, September 26). Outlier Detection with Isolation Forest. Retrieved from https://towardsdatascience.com/outlier-detection-with-isolation-forest-3d190448d45e
[23] Dawson, C. (2019, August 28). Outlier Detection with One-Class SVMs. Retrieved from https://towardsdatascience.com/outlier-detection-with-one-class-svms-5403a1a1878c
[24] Yeh, A. (2019, June 12). A Simple Way to Detect Anomaly. Retrieved from https://towardsdatascience.com/a-simple-way-to-detect-anomaly-3d5a48c0dae0
[25] Li, S. (2019, January 24). Time Series of Price Anomaly Detection. Retrieved from https://towardsdatascience.com/time-series-of-price-anomaly-detection-13586cd5ff46
[26] Anomaly/Outlier Detection using Local Outlier Factors. (n.d.). Retrieved from https://www.datasciencecentral.com/profiles/blogs/anamoly-outlier-detection-using-local-outlier-factors
[27] Ronaghan, S. (2019, March 14). Statistical Tests for Comparing Machine Learning and Baseline Performance. Retrieved from https://towardsdatascience.com/statistical-tests-for-comparing-machine-learning-and-baseline-performance-4dfc9402e46f
[28] Molnar, C. (2019, November 15). 4.6 RuleFit — Interpretable Machine Learning. Retrieved from https://christophm.github.io/interpretable-ml-book/rulefit.html
[29] J. Daniel Semrau. (2017, June 25). The Surprising Longevity Of The Z-Score. Retrieved from https://towardsdatascience.com/the-surprising-longevity-of-the-z-score-a8d4f65f64a0
[30] JJ. (2016, March 23). MAE and RMSE ? Which Metric is Better? Retrieved from https://medium.com/human-in-a-machine-world/mae-and-rmse-which-metric-is-better-e60ac3bde13d
[31] Deng, H. (2018, December 12). Why random forests outperform decision trees. Retrieved from https://towardsdatascience.com/why-random-forests-outperform-decision-trees-1b0f175a0b5.