Large-scale Energy Anomaly Detection (LEAD)

Machine Learning Project

**Team 6**

|  |  |  |  |
| --- | --- | --- | --- |
| Full Name | ID | SEC | BN |
| اسامة صالح فرج السيد | 9210195 | 1 | 12 |
| عبدالرحمن محمد عبدالفتاح محمود | 9210587 | 1 | 23 |
| عبدالرحمن محمد حفني | 9210584 | 1 | 22 |
| عمرو صلاح الدين فؤاد | 9210774 | 1 | 30 |

Submitted to:

Eng. Mohamed Shawky

2025

# Problem Definition

Create a machine learning model to identify anomalies in hourly smart electricity meter readings, distinguishing between normal consumption patterns and unusual energy usage.

# Problem Motivation

Detecting energy consumption anomalies is crucial for energy providers to prevent fraud, identify faulty meters, and promote energy efficiency. Early detection can lead to significant cost savings and more reliable energy distribution.

# Evaluation Metrics

* **Accuracy**
* **Precision & Recall**
* **F1-score**
* **Area Under the ROC** (AUC-ROC)

# Dataset Link and information

[[Link]](https://www.kaggle.com/competitions/energy-anomaly-detection/data?select=train_features.csv)

Dataset has 1.75M rows and 57 features, missing values, and imbalanced classes with ‘0’ class spanning 97% of the dataset.

# Exploratory Data Analysis (EDA)

While exploring our dataset we did the following:

## Removed features that have only one value

Features ***gte\_meter***, ***year*** were removed.

## Exploring features’ relations

* Features like ***hour***, ***day***, ***week***, ***year***, ***hour\_x***, ***...*** are just an expanded form of ***timestamp*** feature, so we removed ***timestamp***.
* Also, features like ***building\_weekday***, ***building\_...*** are just combination of ***building\_id*** and features mentioned above, so we also removed them.

## Handling Null Values

Only ***meter\_reading*** feature had null values, and all rows that have ***meter\_reading*** as null are not anomalous (***anomaly*** = 0).

We dropped those rows as our dataset is large enough anyway.

## Handling non-numeric features

Only ***primary\_use*** feature is non-numeric and has 12 unique categorical values.

We used One-Hot Encoding to convert it to 11 Boolean features.

## Exploring features’ correlations

Highly correlated features were removed.

A graph of a graph with numbers

AI-generated content may be incorrect.

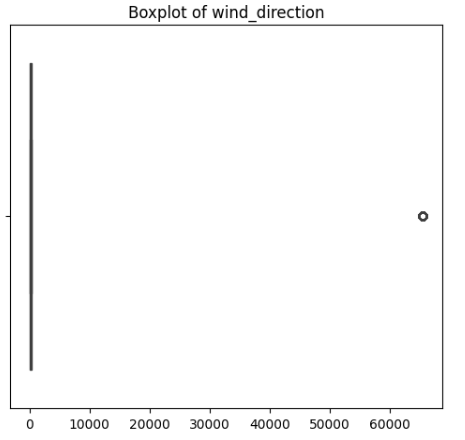
Figure : Correlation between features with threshold > 0.95

## Handling Outliers

Features ***meter\_reading***, ***year\_built***, ***cloud\_coverage***, ***percip\_depth\_1\_hr***, ***wind\_direction*** had outliers that were handled:

* ***wind\_direction*** outliers were handled by using log transformation.

A diagram of a wind direction

AI-generated content may be incorrect. 

* ***year\_built*** and ***cloud\_coverage*** outliers were just a replacement value for missing values that spanned a large amount of the dataset, we dropped them as they aren’t related to energy consumption anomalies anyway.

A diagram of anomalies

AI-generated content may be incorrect. A diagram of a cloud coverage

AI-generated content may be incorrect.

* ***meter\_reading*** and ***percip\_depth\_1\_hr*** outliers were handled by using min-max scaling.

A graph with a line

AI-generated content may be incorrect. A graph of a box plot

AI-generated content may be incorrect.

## Other Observations

* Some features like ***building\_id, site\_id, hour, weekday, month, hour\_x, hour\_y, month\_x, month\_y, weekday\_x, weekday\_y, is\_holiday, gte\_hour, gte\_weekday, gte\_month,*** ***floor\_count*** aren’t really related to if there is an anomaly in energy consumption or not. They may be useful, so we will try both using and not using them.
* The features’ distributions of ***square\_feet*** and ***wind\_speed*** were right-skewed, so we applied log transformation to them.

A graph of a number of feet

AI-generated content may be incorrect. A graph of a wind speed

AI-generated content may be incorrect.

# Experiments and Results

In all experiments, we will try to surpass **ZeroR** base model that gives 97% accuracy.

Dataset was split into train and test sets using stratification on ***anomaly***, so each set has the same distribution of ***anomaly*** feature.

## Linear Model (Logistic Regression)

We tried applying Logistic Regression with stochastic gradient descent learning as a linear binary classifier.

The results were bad as the dataset is heavily imbalanced and the model couldn’t find a linear separator to separate data

A chart with a yellow and purple squares

AI-generated content may be incorrect.

So, we tried to handle the imbalance of the dataset using two approaches:

* **Down Sampling**

We used a balanced subset of the dataset with 50% coverage for both classes, trained the model with different numbers of epochs and different learning rates.

A screenshot of a computer

AI-generated content may be incorrect.

* **Over Sampling using SMOTE** **(Synthetic Minority Over-sampling Technique)**

By generating synthetic data from the minor class (***anomaly*** = 1) till the dataset is balanced, we also trained the model with different numbers of epochs and different learning rates.

A screenshot of a computer screen

AI-generated content may be incorrect.

We can see that the model still underfits with the dataset balanced, meaning our data can’t be linearly separated.

The model achieved similar results when using features removed in [EDA](#_Other_Observations).

## Non-Linear Model (SVM)

As Non-linear SVM takes a lot of time in training, we will first apply PCA to see if we can reduce the dimensionality of our feature space. PCA was applied on the continuous features only.

A graph with a line

AI-generated content may be incorrect.

We can see that almost all features contribute to the variance and dropping a lot of them will affect model's performance badly.

So, we tried two other approaches using SVM with a gaussian kernel (RBF):

* **Down Sampling** (same as linear model)

A black screen with white numbers

AI-generated content may be incorrect.A black screen with white numbers

AI-generated content may be incorrect.

Figure : Non-Linear SVM Figure : Non-Linear SVM with [features](#_Other_Observations) removed

* **RBF Kernel approximation**

By transforming our feature space into a linear one by approximating the gaussian kernel using these steps:

1. Sample D random vectors *ω*i from the Fourier transform of the gaussian kernel

A black background with white text

AI-generated content may be incorrect.

1. Sample random phases *b*i from Uniform [0, 2pi]
2. Using this feature mapping

A math formula on a black background

AI-generated content may be incorrect.

1. Use Dot Products in the New Space

A black background with white letters and numbers

AI-generated content may be incorrect.

Then we applied a linear SVM on this linear space and got the following results:

A black screen with white numbers

AI-generated content may be incorrect.A black screen with white numbers

AI-generated content may be incorrect.

Figure : On Test set Figure : On Train set

We can see that SVM performs better than logistic regression, but it still underfits and linear approximations made it perform worse.

## Ensemble Model (XGBoost)

We used XGBoost instead of AdaBoost to reduce training time as our dataset is large and XGBoost supports parallelization.

XGBoost achieved good results on the whole dataset:

A screen shot of a computer code

AI-generated content may be incorrect.

We tried to improve its performance by:

1. Balancing dataset

* **Down Sampling** (same as linear model)

A screen shot of a computer code

AI-generated content may be incorrect.

* **Over Sampling using SMOTE** (same as linear model)

A screen shot of a computer code

AI-generated content may be incorrect.

We can see that balancing the dataset didn't help as

* Taking a small subset of data didn't make the model generalize well
* Generating synthetic data made the model overfit and generalize poorly to the imbalanced test set
* Boosting models are somewhat robust to imbalanced datasets anyway

1. Hyperparameter Tuning

* **Number of Estimators**

A graph with a line and a number of estimating

AI-generated content may be incorrect.

We found that the best number of estimators is 1000.

* **Class weighting**

A screenshot of a black screen

AI-generated content may be incorrect.

We found that scaling positive class weights by 3 gave the best performance.

After trying the best parameters, we got the following results:

A screen shot of a computer code

AI-generated content may be incorrect.A screen shot of a computer

AI-generated content may be incorrect.

Figure : Score on test set Figure : Score on test set after removing [features](#_Other_Observations)

# Conclusion

Linear model (Logistic Regression) performed badly whether the dataset was balanced or not due to the dataset not being linearly separable.

Non-Linear model (SVM) performed better than linear one but still underfits.

Ensemble model (XGBoost) achieved the best results with accuracy better than our base model ZeroR.

Keeping the features mentioned in [EDA](#_Other_Observations) gave better results.

# Workload Division

|  |  |
| --- | --- |
| Full Name | Task |
| اسامة صالح فرج السيد | EDA |
| عبدالرحمن محمد عبدالفتاح محمود | Non-linear Model |
| عبدالرحمن محمد حفني | Linear Model |
| عمرو صلاح الدين فؤاد | Ensemble Model |