Anomaly Detection in the U.S. Electricity Grid Data

Jiaqi Ding (jd2269) Lewis Tian (zt89) Prabhat Koutha (pk454)

1. Overview

This project investigates the application of an ensemble model for anomaly detection in Energy Information Administration (EIA) hourly electric grid data, focusing on the Northwest region of the United States. Utilizing a comprehensive blend of machine learning algorithms: K-Means, DBSCAN, One-Class SVM, Autoencoder, Random Forest, and Isolation Forest, the study aims to improve the identification of anomalous behavior in power demand data. By enhancing anomaly detection accuracy, the paper underscores its potential implications for electricity traders, particularly in providing reliable warnings about irregularities in the grid's data. The results from the algorithmic approach to anomaly screening have shown the potential to efficiently mitigate data quality issues. The paper also suggests future improvements to make the model more robust.

2. Problem Definition

Electricity trading is a complex process that involves buying and selling electricity on various markets. The process is facilitated by various financial instruments such as futures contracts, options, and swaps. These instruments help traders to manage risks and make profitable trades in dynamic electricity markets. Market participants include energy producers, retailers, traders, brokers, industrial consumers, government entities, banks, hedge funds, and proprietary trading firms. Informed decision-making is essential for successful electricity trading, and electricity grid data is a valuable resource that provides insights into market trends, price forecasting, risk management, portfolio optimization, and regulatory compliance. By analyzing this data, traders can identify potential opportunities and risks, allowing them to make informed trading decisions.

However, anomalies in electricity grid data can have significant consequences for traders. These anomalies can occur due to various reasons, such as data errors, measurement inaccuracies, or equipment malfunctions, and can lead to incorrect price forecasting, improper risk management, and inaccurate regulatory compliance. To address this problem, we aim to identify anomalies in electricity grid data. By detecting and replacing these anomalies, we can improve the accuracy of price forecasting, risk management, and regulatory compliance, helping traders to make informed decisions and maximize profits.

3. Data

The Energy Information Administration (EIA) [1] offers access to electricity grid data with a range of features such as demand, demand forecast, net generation, and

generation by various energy sources. It also separates the U.S. electricity grid into several regions. Each region, besides an aggregated data source, has a number of smaller data sources called balancing authorities. Due to the nature of electricity grid data, the raw data is relatively well-maintained. In this project, we chose to analyze data from the Northwest region because it consists of over twenty independent balancing authorities, making it more likely for an anomaly to occur. Our data spans from July 1, 2015 to February 10, 2023 and has over 66000 hourly observations.

To effectively detect anomalies in the aggregated demand in the Northwest region, it is necessary to preprocess the raw data acquired from the EIA. In this study, we identified that not all of the features included in the dataset are essential for anomaly detection. Therefore, we performed feature selection to reduce redundant features and avoid introducing noise to the data. [2] Specifically, we selected the aggregated demand time series for the entire Northwest region, as well as the demand for the seven largest balancing authorities. Additionally, we included the first- and second-degree rate of demand change as features, as these can also serve as indicators of anomalous data. Furthermore, our domain knowledge on electricity data led us to discover that the demand is unlikely to remain unchanged for two consecutive hours. Hence, we created a new binary feature that marks an entry as 1 when this event occurs and as 0 otherwise, to assist machine learning algorithms in identifying these anomalies.

Although the EIA maintains the data well, it still contains missing values, which may negatively impact the performance of machine learning algorithms. We utilized linear regression to impute missing values in time series from balancing authorities based on their high correlation with the Northwest aggregated demand. Additionally, we standardized each column to ensure that different features with varying magnitudes would be treated equally during the modeling process. Overall, these preprocessing steps enabled us to make the data for anomaly detection more robust.

4. Methodology

4.1. K-Means

K-Means is a popular unsupervised machine learning algorithm that can be effectively utilized for anomaly detection. [3] In this context, the algorithm functions by partitioning the input data into distinct clusters based on the similarity of data points, which is typically determined by their Euclidean distance. During the clustering process, K-Means iteratively refines the centroids of each cluster to minimize the within-cluster variation. Mathematically, it can be formulated as

$$J = \sum_{j=1}^{k} \sum_{i=1}^{n} |x_i^{(j)} - c_j|^2$$

where J is the objective function, k is the number of clusters, n is the number of cases, c_j is the centroid for cluster j.

Once the optimal clustering is achieved, anomalies can be identified by analyzing the distance of each data point from its corresponding cluster centroid. Data points located far from their centroids are considered outliers, as they deviate significantly from the common patterns observed within the cluster. K-Means-based anomaly detection has proven particularly valuable due to its simplicity, scalability, and ability to uncover hidden patterns in large and complex datasets.

4.2. DBSCAN

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) is another popular unsupervised machine learning algorithm used for anomaly detection. [4] Unlike K-Means, DBSCAN does not require us to specify the number of clusters upfront and it can discover clusters of arbitrary shapes, making it particularly useful for datasets with complex spatial distributions. It works by defining clusters as high-density regions in the data space separated by areas of lower density. During its operation, DBSCAN classifies data points into three categories: core points, border points, and noise points. In the context of anomaly detection, the noise points, which lie in low-density regions and are far from any cluster, are considered anomalies. Comparatively, while K-Means is simpler and more scalable for large datasets, it may struggle with detecting anomalies when data is not spherical or evenly distributed, as it assumes isotropic clusters. DBSCAN, on the other hand, can handle complex spatial distributions better, and it inherently identifies anomalies as part of its clustering process, but it may struggle with datasets of varying density.

4.3. One Class SVM

The conventional SVM is used for classification and identifies a hyperplane with the maximum margin to distinguish positive and negative examples. However, one class SVM operates by minimizing the hypersphere of the single class of examples in the training data and treating all other samples outside of the hypersphere as outliers that fall outside of the training data distribution. [5] The figure below illustrates the hypersphere created by one class SVM to develop the capacity to classify data outside of the training distribution based on the hypersphere.

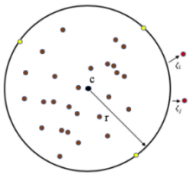

Figure 1 One-Class SVM visualization

The mathematical expression of this model is the following, where r is the radius, c is the center:

$$\min_{r,c,\zeta} r^2 + \frac{1}{vn}\sum_{i=1}^{n} \zeta_i$$

$$subject\ to\ |\phi(x_i) - c|^2 \leq r^2 + \zeta_i$$

Similar to the SVM model, we can use kernel function to map the data to a higher dimension. To learn the non-linear relationship among features, we used rbf function to transform the data onto higher dimension. Then we trained the model on the mapped data, set a margin around the hyperplane to label data as anomalies or not.

4.4. Autoencoder

Autoencoder is a type of neural network that can learn to replicate their inputs as outputs. They are commonly used for learning compressed representations of raw data. [6] Unlike PCA, which is a linear method for dimensionality reduction, autoencoders use non-linear transformations that allow them to learn more powerful generalizations. Autoencoder has two layers. The first layer is the encoder layer, it takes in input data with a high dimensionality and converts it into latent data with a lower dimensionality. The second layer is the decoder layer, it takes the output of the encoder as input and aims to reconstruct the original input data.

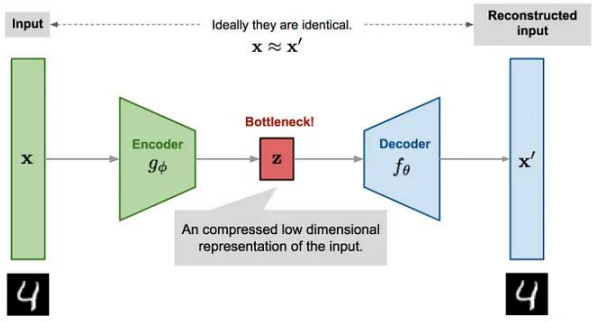

Figure 2 Autoencoder diagram

Autoencoders can detect anomalies because they are trained to recognize patterns in input data and properly rebuild it. By definition, an anomaly is an observation that deviates from the expected pattern or behavior. When an autoencoder is fed an abnormal input, the reconstruction error is likely to be significant, suggesting a deviation from the normal pattern. Autoencoders can thus be used to detect abnormalities based on the size of the reconstruction error. The input is considered abnormal if the reconstruction error exceeds a specific threshold.

For our project, we used the ReLU function as the encoder function, because this is simple, computationally efficient, and has been shown to be effective in many deep learning models. Mathematically, it can be written as $f(x) = \text{ReLU(x)} = \max(x, 0)$. We then chose the tanh function, $f(x) = \tanh(x) = (e^{2x} - 1)/(e^{2x} + 1)$, as the decoder function because we have standardized the data, and the tanh function outputs

values between -1 and 1. It is also symmetric around zero, which can be helpful in reducing the bias in the reconstructed data.

4.5. Random Forest

Random Forest is an ensemble supervised learning algorithm that utilizes multiple decision trees to perform classification and regression tasks. The algorithm leverages the Bootstrap Aggregation (Bagging) technique to generate an ensemble of decision trees by randomly drawing bootstrap sampled datasets from the training data and train a model on each sample data before aggregating the results together [7]. Each decision tree is constructed by recursively partitioning the data based on the most informative feature at each node. The split is made based on maximizing a split criterion, such as information gain and Gini impurity.

Due to the major limitation that the true anomaly labels are unknown, we had to first create labels to train random forest. The labels were created by setting thresholds to identify data points that were outside of the expected range. In order to ensure that the proportion of identified anomalies was consistent across all of our algorithms, we set the threshold values based on a target percentage of 0.5% identified anomalies. In addition, since there is seasonality in the demand data, we created half-year windows to have a more uniform distribution of anomalies as shown in figure X. After labeling anomalies using thresholding, we trained random forest with the labeled data.

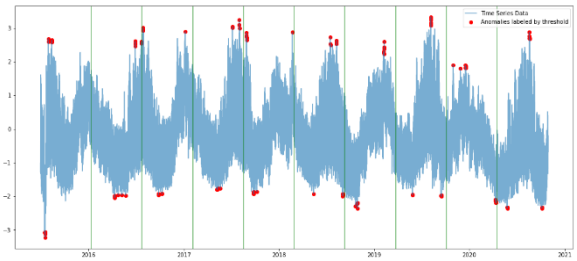

Figure 3 How we used windows and thresholds to label anomalies

4.6. Isolation Forest

Another tree-based algorithm we used for anomaly detection is the isolation forest, an ensemble method to separate anomalous data points from normal ones in a highly efficient manner. The algorithm works by randomly selecting a feature and then randomly selecting a split value between the maximum and minimum values of that feature. The data is then partitioned based on the randomly selected split value, and the process is repeated recursively until all data points are isolated in their own leaf nodes [8]. The average depth of the tree needed to isolate a data point is used to calculate the anomaly score for that point. The intuition behind this approach is that anomalous data points can be more easily separated and isolated from the rest of the data using fewer splits compared to normal data points. As a result, anomalous data points will have a lower average depth value and hence a higher anomaly score.

Isolation forest is an unsupervised learning algorithm, which makes it suitable for our task without true anomaly labels. Its *contamination* hyperparameter also allowed us to set the percentage of anomalies in the data.

4.7. Ensemble Method - Majority Voting

After running the above six algorithms and generating six sets of anomaly labels, we used majority voting to get the final results. Since the six anomaly detection algorithms work in different ways and produce relatively complementary errors, we treated them as an ensemble and used majority voting to make the final results more robust. To avoid biases toward a single algorithm because of the number of anomalies it produces, we tweaked their hyperparameters to keep the percentage of anomalies in the data around 0.5% for all algorithms.

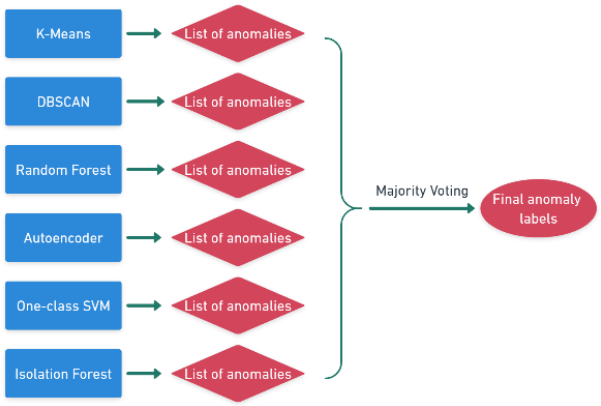

Figure 4 Workflow of ensemble method

5. Results

5.1. Anomaly Labels

The results produced by the six anomaly detection algorithms and their majority voted final results are shown below. Notably, the K-Means and DBSCAN algorithms generated similar anomaly labels that were rather uniformly distributed across the entire time frame. Conversely, the tree-based algorithms, namely random forest and isolation forest, identified anomalies that were more concentrated around extreme values, particularly towards the end of the time frame. In contrast, the One-class SVM and autoencoder algorithms produced results that were intermediate to the two aforementioned types of algorithms, as the anomalies exhibited a moderately dispersed distribution with some extreme values. These differences in the anomaly detection results highlight the potential benefits of using majority voting to generate final results. By aggregating the outputs of different algorithms, the overall errors are reduced, and the generalizability of the approach is enhanced.

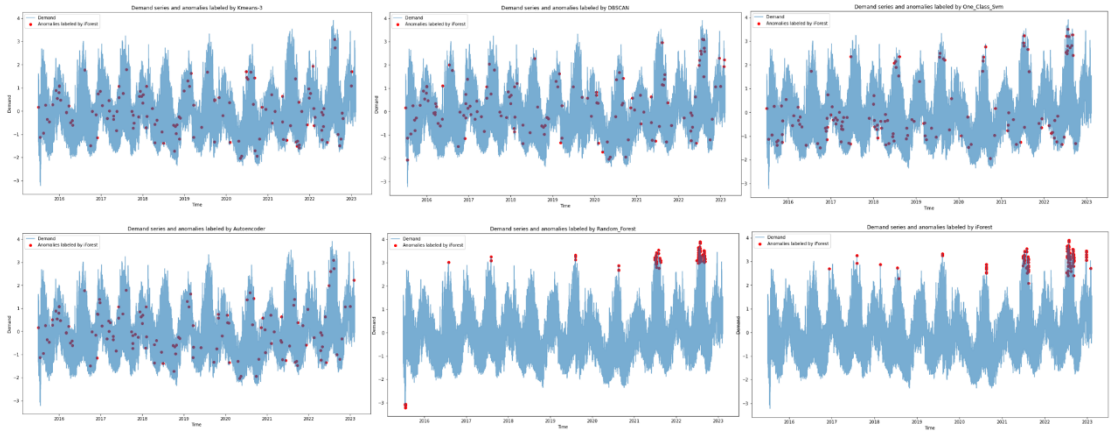

Figure 5 Anomaly labels from each algorithm

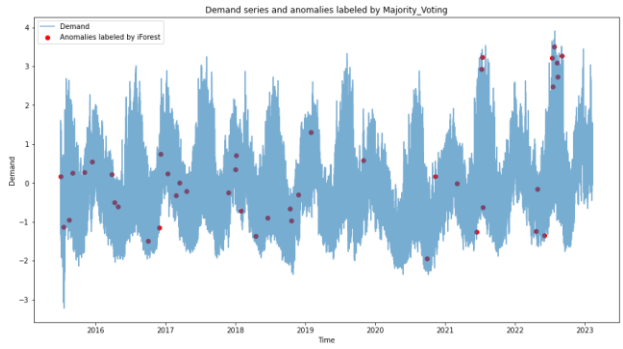

Figure 6 Anomaly labels from majority voting

Finally, the majority-voted results are shown above. We can see that it gives a more realistic and reasonable set of labels than each individual result, both in terms of the number of anomalies and the distribution of them. Upon closer examination, we checked some of the labeled entries and confirmed that they are actual anomalies. In addition, we used the final results from the ensemble method as the "true" anomaly label to calculate metrics such as precision, recall, and AUC. See the Appendix for the full table. Our results show that one-class SVM outperformed others while tree-based algorithms underperformed. But it is important to note that this is not conclusive as we do not have the actual anomaly labels. Computationally, clustering-based algorithms such as DBSCAN were slower whereas the tree-based algorithms were relatively fast.

5.2. Weapon of Math Destruction

"Weapons of Math Destruction" refers to mathematical models or algorithms that, despite their seemingly objective nature, can lead to harmful or destructive outcomes, particularly for the most vulnerable segments of society [9]. Such models are typically characterized by being opaque, scalable, and damaging. For this project, following potential issues could arise:

1. **Opacity**: Machine learning algorithms such as Autoencoder are often black boxes. This could result in unpredictable behavior, such as false positive or false negative detections, which could cause unnecessary panic or missed warnings.

2. **Scale**: If a model with inherent biases or errors is used on a large scale, the negative impact can be widespread. While the project looks at the whole of Northwest region, the associated trading volume is relatively low.

3. **Damage**: Our model is an effective way to screen for anomalies but if the model is blindly trusted it could lead to significant trading losses.

It's important to note that the risk of becoming a weapon of math destruction doesn't inherently reside in the algorithms themselves, but in how they are used, the data they are trained on, and the interpretations made from their predictions. Our project is geared towards electricity traders primarily for algorithmic screening of anomalies hence the possibility of our model being used as a weapon of math destruction is very low.

5.3. Fairness

While our project primarily concerns electricity traders and firms, fairness still has potential relevance. This pertains to ensuring the algorithm doesn't generate biases that may unfairly affect specific groups of traders or participants. One main concern is data representation [10]. We're using open-source data from the EIA website, but if some market participants have access to superior non-public data, this could lead to imbalances and potentially breach trading regulations. Thus, it's crucial to ensure the model operates fairly even in this context.

6. Conclusions

Based on our research, there exists a trade-off between algorithm performance and calculation time. The One-class SVM algorithm achieved the highest score across all four metrics, but has a moderate running time. On the other hand, tree-based algorithms had the lowest scores but were the fastest. Our final labels were evaluated by actual traders and were deemed reasonable, thus our approach provides an effective preliminary screening method for traders and experts to identify anomalies with improved efficiency.

While we are confident that our results can have positive practical impacts on traders' workflow, we do think that it can be improved. One possible direction for future work would be to address the issue of diverse errors, where similar weaknesses in models may reduce ensemble benefits. One potential solution would be to use diverse and complementary algorithms to offset individual weaknesses. Additionally, there are issues such as interpretability, incompatibility of data assumptions, and sensitivity to noise that need to be considered.

References

[1]
“Real-time Operating Grid - U.S. Energy Information Administration (EIA),” *www.eia.gov*. https://www.eia.gov/electricity/gridmonitor/dashboard/electric_overview/US48/US48
[2]
P. Cramton, “Electricity market design,” *Oxford Review of Economic Policy*, vol. 33, no. 4, pp. 589–612, 2017, doi: https://doi.org/10.1093/oxrep/grx041.
[3]
I. Arroyo, “Unsupervised Anomaly detection on Spotify data: K-Means vs Local Outlier Factor,” *Medium*, Feb. 17, 2022. https://towardsdatascience.com/unsupervised-anomaly-detection-on-spotify-data-k-means-vs-local-outlier-factor-f96ae783d7a7 (accessed May 11, 2023).
[4]
N. S. Chauhan, “DBSCAN Clustering Algorithm in Machine Learning,” *KDnuggets*, Apr. 04, 2022. https://www.kdnuggets.com/2020/04/dbscan-clustering-algorithm-machine-learning.html
[5]
V. Kilaru, “One Class Classification Using Support Vector Machines,” *Analytics Vidhya*, Jun. 03, 2022. https://www.analyticsvidhya.com/blog/2022/06/one-class-classification-using-support-vector-machines/#:~:text=One%2DClass%20SVM%20is%20an (accessed May 11, 2023).
[6]
R. Khandelwal, “Anomaly Detection using Autoencoders,” *Medium*, Jan. 28, 2021. https://towardsdatascience.com/anomaly-detection-using-autoencoders-5b032178a1ea#:~:text=Anomaly%20Detection%3A%20Autoencoders%20tries%20to (accessed May 11, 2023).
[7]
L. Breiman, “Random Forests,” *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001, doi: https://doi.org/10.1023/a:1010933404324.
[8]
F. T. Liu, K. M. Ting, and Z.-H. Zhou, “Isolation Forest,” *2008 Eighth IEEE International Conference on Data Mining*, Dec. 2008, doi: https://doi.org/10.1109/icdm.2008.17.
[9]
M. Udell, “ORIE 4741: Learning with Big Messy Data Limitations and Dangers of Predictive Analytics,” 2021. Accessed: May 11, 2023. [Online]. Available: https://people.orie.cornell.edu/mru8/orie4741/lectures/limits.pdf
[10]
M. Udell, “ORIE 4741: Learning with Big Messy Data Fairness,” 2021. Accessed: May 11, 2023. [Online]. Available: https://people.orie.cornell.edu/mru8/orie4741/lectures/fairness.pdf
[11]
Y. Zhao and X. Zhou, “K-means Clustering Algorithm and Its Improvement Research,” *Journal of Physics: Conference Series*, vol. 1873, no. 1, p. 012074, Apr. 2021, doi: https://doi.org/10.1088/1742-6596/1873/1/012074.
[12]

S. Singh, “All you need to know about the DBSCAN Algorithm,” *https://medium.com*, 2023. https://medium.com/analytics-vidhya/all-you-need-to-know-about-the-dbscan-algorithm-f1a35ed8e712 (accessed May 11, 2023).
[13]
Z. Hu and Z. Xue, “On the Complexity of One-class SVM for Multiple Instance Learning,” *arXiv:1603.04947 [cs]*, Mar. 2016, Accessed: May 11, 2023. [Online]. Available: https://arxiv.org/abs/1603.04947
[14]
M. Akhgary, “machine learning - What is the time complexity for training a neural network using back-propagation?,” *Artificial Intelligence Stack Exchange*, Mar. 18, 2018. https://ai.stackexchange.com/questions/5728/what-is-the-time-complexity-for-training-a-neural-network-using-back-propagation (accessed May 11, 2023).
[15]
S. SURANA, “Computational Complexity of Machine Learning Models - II | Data Science and Machine Learning,” *www.kaggle.com*. https://www.kaggle.com/general/263127
[16]
C. Stanton, Gilad Katz, and D. Song, “Isolation Forest for Anomaly Detection.” Available: https://e3s-center.berkeley.edu/wp-content/uploads/2017/08/RET_CStanton-2015.pdf
[17]
K. YÜKSEL, “Accuracy, Precision, Recall, F1 Score and ROC curve,” *emkademy.com*, Mar. 02, 2020. https://emkademy.com/research/toolbox/2020-03-02-accuracy-precision-recall (accessed May 11, 2023).

Appendix

A. Individual team member contributions:

Prabhat Koutha (pk454): Ideation, Problem definition, K-Means, DBSCAN, Score table, Weapon of Math Destruction and Fairness

Lewis Tian (zt89): Data, Random Forest, Isolation Forest, Ensemble, Results

Jiaqi Ding (jd2269): One Class SVM, Autoencoder, Conclusion, Data preprocessing code

B. Score table**

	Recall	**Precision**	**F-1 score**	**AUC-ROC**	**Space complexity**	**Time complexity**
K-Means [11]	0.9985	0.9999	0.9992	0.9295	**O(n * K * d)**	**O(n * K * I * d)**
DBSCAN [12]	0.9984	0.9999	0.9992	0.9411	**O(n)**	**O(n*log(n))**
One-Class SVM [13]	0.9986	1.0000	0.9993	0.9877	**O(n^2)**	**O(n * d)**
Autoencoder [14]	0.9985	0.9999	0.9992	0.9295	**O(n * d)**	**O(n^2)**
Random Forest [15]	0.9983	0.9994	0.9989	0.5573	**O(n * t)**	**O(n * t * log(n) * d)**
Isolation Forest [16]	0.9981	0.9995	0.9988	0.5921	**O(n)**	**O(t * n * log(n) * d)**

**In anomaly detection, the majority of instances are "normal" (negative class) and a small minority are "anomalies" (positive class). This is known as an imbalanced classification problem. Since we don't have access to the true labels, we are using the ensemble labels as the true labels. Since the ensemble strategy is designed to label anomalies if a majority of the models do so, then any instance that an individual

model labels as an anomaly is very likely to be labeled as an anomaly by the ensemble as well. This would lead to a high number of True Positives (TPs) compared to True Negatives (TN), False Positives (FP), and False Negatives (FN). As a result, we have a high recall, precision and F-1 score. While high precision, recall, and F-1 score are generally desirable, here our **goal is to show that individual models are contributing effectively to the ensemble model and that there is a high degree of agreement between the individual models and the ensemble**.