CMPT 318 Assignment #2

Group 7

Question 1:

(I) What is feature scaling?

Feature scaling is a data preprocessing technique used to standardize the independent variables in a dataset (Gupta) . This technique results in shared common characteristics, such as having a mean of 0 and the same standard deviation. Feature scaling is crucial when dealing with features that have highly varying input values. Without these techniques, machine learning algorithms can become biased, producing skewed outcomes and affecting the model's performance (Feature Scaling - an Overview"). Two common feature scaling techniques are standardization and normalization.

(II) Why scaling features of a dataset is necessary?

In short, feature scaling improves the efficiency and accuracy of machine learning algorithms ("Feature Scaling," Dremio), more specifically for Distance and gradient-based algorithms such as K-Nearest Neighbors and principal component analysis ("Feature Scaling," Lyzr). Given a set of independent features, feature scaling is important due to the possible variation of ranges between those independent variables. Feature scaling assures that importance is not given to a feature simply due to its value range ("Feature Scaling - an Overview"). If one feature ranges from 0.002 to 0.005 while another ranges from -1,000,000 to 1,000,000, the latter could disproportionately influence the model, leading to incorrect and biased results. As noted by Mohit Gupta from Geeks for Geeks, machine learning algorithms tend to interpret larger values as more significant, regardless of their actual units (Gupta).

For example, if you were creating a machine learning algorithm to predict the total number of words in a textbook and the two features were an average number of words per page and the number of pages, without necessary scaling, the number of pages (on average the larger value) would dominate the calculations and could make the average number of words per page irrelevant. By scaling these two features, calculating the number of words becomes less dependent on the number of pages in comparison to the average number of words.

Another example from Aniruddha Bhandari at Analytics Vidhya is a machine learning algorithm based on the GPA of a student and their future income. In this example, the future income has a much larger range than the GPA and thus could create bias in the machine learning algorithm. By scaling the features of the dataset, it is ensured that neither feature dominates the other (Bhandari).

(III) What does normalization and standardization do to the data and the noise ?

Normalizing a dataset transforms the measurements to fall within a specific range, usually [0, 1] (Jalswal). This method preserves the relative distance between data points while making all of the measured features follow the same scale. This results in equal weighting of different features in a machine learning model as described in (II). There are several methods in which data can be normalized to a particular range. One method is the min-max scaling formula, which transforms data to fall in the range [0,1] ("Normalization"):

$$x_{normalized} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

where x is the original measurement, and x_{min} and x_{max} are the minimum and maximum values in the dataset for the feature that is to be normalized. Min-max scaling and most other forms of normalization have little effect on noise as outliers are usually scaled alongside normal data. Although this is true for most forms of normalization, Robust scaling scales data points based on how far they are from the median, which is much more resilient to outliers compared to the mean of a dataset or its minimum or maximum values, which are all used in other normalization schemes (Singh). In addition to the median, the interquartile range (IQR) is also used in robust standardization.

$$x_{robust} = \frac{x - median}{IQR}$$

Standardization is a form of normalization which results in a dataset with a mean of 0 and standard deviation of 1 (Jaadi). The formula for standardization involves using the mean and standard deviation of the unstandardized dataset: μ and σ respectively.

$$x_{standardized} = \frac{x - \mu}{\sigma}$$

Standardizing the data in this way does not guarantee a particular range in the same way that other forms of normalization do, however, it does guarantee that the standardized dataset has a mean of 0 and standard deviation of 1. This is valuable since scaling the data to fit these parameters causes it to follow the standard Gaussian distribution. Transforming the data in this way is useful since using the standard Gaussian is consistent across different data, and results in increased machine learning accuracy, as well as higher efficiency in databases (Jaadi). Standardization may result in some denoising for datasets with small outliers, however since it uses the mean and standard deviation in its calculation, if extreme outliers are present in the dataset, they will skew the mean and standard deviation and be scaled alongside normal values.

Question 2:

Most anomalous week: Week 52

Least anomalous week: Week 34

Scoring rational:

To identify the most and least anomalous weeks, we calculated the standard deviation of deviations for each week. Specifically, we computed the differences between the smoothened Global Intensity values and the average smoothened week. The standard deviations of these deviations were used as an anomaly score. A low score indicates that the week closely follows the average trend (least anomalous), while a high score indicates significant deviations from the average (most anomalous). This method provides a clear, interpretable measure of variability and is capable of being extended to other variables in the dataset. Additionally, the data aligns with real-world energy consumption patterns, as demonstrated by the last complete week as the most anomalous during the holiday season (Week 52), and the least anomalous week during a period of stable weather (Week 34).

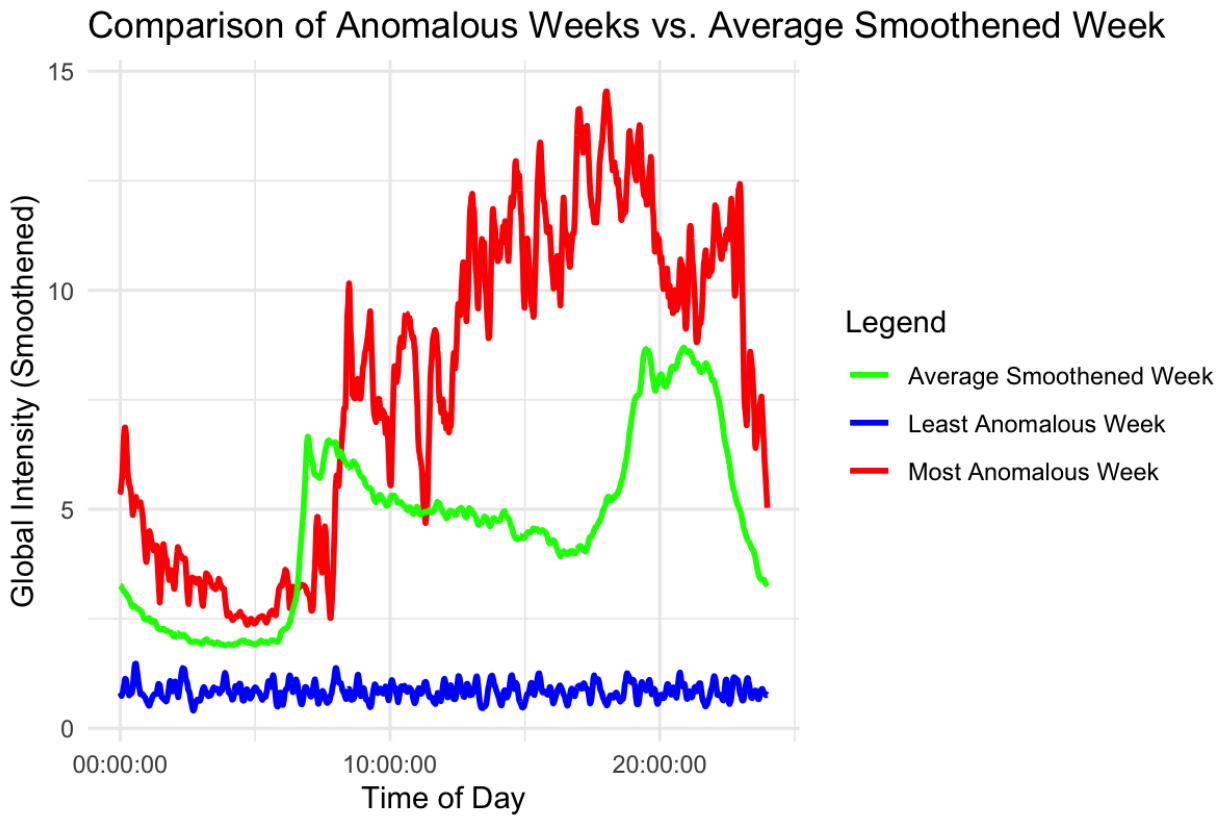
Anomaly table:

Week	Anomaly Score
1	5.087425051
2	4.9713799
3	4.792756942
4	4.113260136
5	4.998541472
6	4.257090293
7	4.769186082
8	5.175824999
9	3.634709238
10	4.802891693
11	3.879993699
12	4.537646968
13	4.524706983
14	4.222861326
15	2.516295169

16	3.432527735
17	2.731041063
18	3.741805405
19	3.649719118
20	3.676370238
21	3.465953579
22	3.206196506
23	3.405651998
24	3.525695932
25	3.575815959
26	3.103584399
27	3.271545407
28	3.201362271
29	3.007031209
30	2.519978875
31	2.866423432
32	2.100793371
33	2.041780299
34	2.020056873
35	3.208697047
36	3.272966
37	3.192653616
38	3.397696962
39	3.288678746
40	3.760196301
41	4.156169296
42	3.846388151
43	4.163873806
44	3.731594535
45	4.033665262
46	4.387490787
47	4.016808271
48	4.732928819

49	4.621598554
50	4.289481101
51	4.247114014
52	5.525765384

Plot of the smoothened versions of the most and the least anomalous weeks against the average smoothened week:



For graphing purposes, we decided to compute the average Global Intensity at each time point for both the most and least anomalous weeks. The resulting graph provides a more representative view of the deviations from the average smoothened week.

Question 3:

Hidden Markov Models (HMMs) are essential for the detection of anomalous patterns in cyber intrusion detection systems. These models allow continuously operating cyber security systems such as supervisory control systems to analyze inbound and outbound data streams in real time, alerting of any suspicious activity that could be the result of cyber intrusions. The following section will discuss how these anomalous patterns are detected by Hidden Markov Models and the importance of having efficient real time anomalous pattern detection.

Hidden Markov Models function by learning from historical data, defining states, and then calculating transition and observation probabilities. By analyzing past data stream sequences, a HMM can determine typical behavioral patterns and detect deviations which could indicate potential security threats. Once it is in use, the model takes in the current state (data stream) as input and makes a prediction on what the next state/data stream should look like allowing it to detect any anomalous data patterns that arise in the system.

Although this is great in theory, Hidden Markov Models come with complexities, three main problems of which problem 1 interests us. Rabiner defines problem 1 as the task of evaluating the probability that a sequence of observations occurred given a particular HMM. In other words, how well was our model able to predict the sequence of observations. In order to efficiently calculate this probability, Rabiner advises the use of the Forward-Backward Algorithm: A procedure that inductively calculates the desired value in $O(n^2)$ time, which is very fast for this particular problem. Efficient computation of an observation's probability is crucial, as it enables real time monitoring and threat detection, minimizing delays in cyber intrusion detection that could allow cyber threats to persist undetected. At the same time, slow computation can discourage organizations from implementing Hidden Markov Model based cyber intrusion detection systems, reducing their infrastructur's security effectiveness.

In conclusion, Hidden Markov Models provide a structured approach to detecting anomalies by evaluating a system's standard operation and predicting upcoming behavioral patterns. Ensuring that the calculation of probabilities is efficient is crucial to a Hidden Markov Model's success since slow computation would hinder the model's ability to stop potential threats from propagating in the systems the model was intended to protect.

References:

Gupta, Mohit. "Feature Engineering: Scaling, Normalization, and Standardization." *GeeksforGeeks*, 17 Jan. 2025, www.geeksforgeeks.org/ml-feature-scaling-part-2/.

"Feature Scaling." *Dremio*, 6 Jan. 2025, www.dremio.com/wiki/feature-scaling/#:~:text=Why%20is%20Feature%20Scaling%20important,model%20due%20to%20its%20range.

"Feature Scaling: Enhance Model Performance with Data Normalization." *Lyzr*, 2 Jan. 2025, www.lyzr.ai/glossaries/feature-scaling/.

"Feature Scaling." *Feature Scaling - an Overview | ScienceDirect Topics*, www.sciencedirect.com/topics/computer-science/feature-scaling#:~:text=Feature%20Scaling%3A%20Feature%20scaling%20is,the%20data%20pre%2Dprocessing%20step. Accessed 9 Feb. 2025.

Bhandari, Aniruddha. "Feature Scaling: Engineering, Normalization, and Standardization (Updated 2025)." *Analytics Vidhya*, 20 Dec. 2024, www.analyticsvidhya.com/blog/2020/04/feature-scaling-machine-learning-normalization-standardization/.

Jaadi, Zakaria. "Data Standardization: Why to Do It and How It Matters" *builtIn*, Whitfield, Brennan. 9 Jan. 2025, https://builtin.com/data-science/when-and-why-standardize-your-data

Jalswal, Sejal. "What is Normalization in Machine Learning? A Comprehensive Guide to Data Rescaling" *datacamp,* 4 Jan 2024, https://www.datacamp.com/tutorial/normalization-in-machine-learning

Singh, Yashmeet. "Robust Scaling: Why and How to Use It to Handle Outliers" *Proclus Academy,* 22 Mar 2022, https://proclusacademy.com/blog/robust-scaler-outliers/