Anomaly-Based Intrusion Detection for Cyber-physical Systems

Abstract

This paper explores the use of anomaly-based intrusion detection to analyze household electricity consumption data. By training our Hidden Markov Model (HMM) on three years of normal behaviour, the intrusion detection system can identify abnormal behaviour. After scaling the data, we selected two of the variables, then used the timeframe of Wednesdays 4-8pm to train with. We then ran the model with 8 states and successfully identified the normalized training log-likelihood to be -0.222 with a maximum deviation of 1.339.

Contents

List of Figures

List of Tables

1 Introduction

The methods used for cyberattacks today are becoming more and more difficult to defend against and the increased attack surface of critical infrastructure has made it more and more vulnerable, requiring modification to our defense methods. With the limitation of traditional signature-based intrusion detection, we must shift to a more robust method of intrusion detection: behavioural anomaly detection. The aim of this project is use this method to accurately detect abnormal behaviour that may be cause for concern, specifically with the provided household electricity consumption data.

2 Feature Scaling

As a first step before training our model, feature scaling is necessary in preprocessing the dataset; the factors were measured on different scales and therefore difficult to compare. By rescaling the different magnitudes of the features to one scale, it ensured each feature could contribute equally in analysis carried out in other steps. This allows for more accurate interpretability of relationships between the different features and unbiased results. Two popular methods of feature scaling are standardization and normalization, of which we chose to use standardization.

In normalization, each feature is scaled to a range between 0 and 1. This is achieved by subtracting the minimum value and dividing by the range (maximum value - minimum value) of each feature:

$$Xscaled = \frac{X - Xmin}{Xmin - Xmin}$$

As the range is compressed to a fixed interval, any values which are particularly large or small compared to the rest of the dataset will greatly affect the scaling. In this way, this method of scaling is more susceptible to outliers than standardization. The rescaling to a fixed range can also reduce noise to a certain extent.

In standardization (z-score normalization), each feature is scaled to have a mean of 0 and a standard deviation of 1. This is achieved by subtracting the mean and dividing by the standard deviation of each feature.

$$Xscaled = \frac{X - \mu}{\sigma}$$

μ: the mean of feature X

σ: the standard deviation of feature X

The standardization of a dataset does not change either the range or the distribution, since the data is scaled using the mean and the standard deviation. As such, this method is robust to outliers and the result is largely unaffected by noise.

We chose to use (Z-score) standardization over normalization due to the higher susceptibility of normalization to outliers. Another factor is that normalization does not always preserve the original distribution shape, whereas standardization does. This is especially important for anomaly detection. Through this step, some balance was provided to the importance of the original features.

3 Feature Engineering

3.1 Introduction

Feature engineering is very critical within the data preprocessing of machine learning, particularly when such a volume and complexity of data can be pretty overwhelming to handle—like in the cybersecurity domain. This step transforms raw data into features that better represent what the underlying problem is for predictive models and, hence, contribute to better model accuracy on unseen data.

3.1.1 Overview of PCA

Principal Component Analysis (PCA) is a statistical procedure that converts sets of observed data, which are supposed to be correlated, into a smaller set of linearly uncorrelated data called principal components. It captures the variance of the data in all the different components and hence helps us reduce dimensionality without losing much of the predictive information. Thus, PCA helps in reducing the dimensionality and thus providing simplicity to our dataset, so that high-dimensionality is avoided and the computational efficiency increases.

3.1.2 Purpose of the Report and the Role of PCA

This report summarizes the process and findings made during the application of PCA within the broader task of unsupervised intrusion detection in a supervisory control system. The main target of this exercise is to identify and isolate significant features from a multivariate dataset. By doing so, we aim to refine the input for Hidden Markov Models (HMMs) that are to be trained subsequently. This is where PCA plays an instrumental role, aiding greatly in distilling the essence of the data while helping in the discernment of normal patterns from potential anomalies.

3.1.3 Background Context

This assignment is set in the context where the level of cyberattacks is growing—developing sophistication, meaning a much more advanced and proactive detection system. Given multivariate data streams in supervisory control systems, traditional methods of feature selection based on correlation do not cut the cake. The PCA provides the best alternative to ensure that, after considering the variables to be engineered and modeled, those to be finally modeled are most indicative of anomalous behavior. This will improve the fidelity of our anomaly detection system in a principled manner that adheres to the overall goal of cybersecurity to assure integrity and reliability in critical infrastructure systems.

3.2 Methodology

3.2.1 Data Description

The dataset on which PCA was performed is a collection of multivariate time series. The dataset was derived from a supervisory control system that is meant to oversee and regulate the consumption of energy in residential buildings. We have a comprehensive dataset that spans three years of event resolution, with about 1.5 million recorded entries. These entries include seven chronological variables with a measure of about seven distinct measurements related to electricity consumption: continuous variables like instantaneous electrical power in kilowatts as well as discrete variables like operational mode indices. We split the data into train and test data, the train data containing 80 percent of the total dataset and then the test data containing 20 percent of the total dataset.

3.2.2 PCA Process

For the decomposition of the standardized data into principal components, the `prcomp` function was used. This function allows one to do the Singular Value Decomposition (SVD) of the centered and scaled data matrix. This method transformed the correlated variables into a set of linearly uncorrelated principal components, utilizing the SVD algorithm to maximize variance and thereby capture the inherent structure of the data.

As to the selection of the number of principal components to be kept, we were guided by the loading scores and the scree plot. In particular, the number of components was chosen based on eigenvalues: only the components with an eigenvalue of above one were retained since they account for the most variance. Additionally, we used the scree plot method to confirm the adequacy of our choice based on the percentage score of each PC.

3.3 Implementation

This had been done in a series of steps: data cleaning and preparation, standardization, then implemented in the decomposition, as described previously. The issues that were to be tackled involved the missing values, and this was solved through the omission of missing data points. This preparatory work was such that PCA could follow smoothly and provide a set of Principal Components that we subsequently used for Feature Engineering in the HMM training and testing phase.

3.4 Results

In our Principal Component Analysis, shown in Table 1 displays the standard deviation of PC1 being 1.7957 which is considered to be high. This standard deviation shows us the square root of the corresponding eigenvalue, showing the degree of variance captured by PC1 when multidimensional data is projected onto this new axis. This value represents the PC1 accounts for the spread of data, with the SVD at 1.7957 in the transformed feature space.

Table 1: Importance of components.

	PC1	PC2	PC3	PC4	PC5	PC6	PC7
Standard deviation	1.7957	0.9947	0.9657	0.9211	0.8868	0.46672	0.02714
Proportion of Variance	0.4607	0.1413	0.1332	0.1212	0.1123	0.03112	0.00011
Cumulative Proportion	0.4607	0.6020	0.7352	0.8564	0.9688	0.99989	1.00000

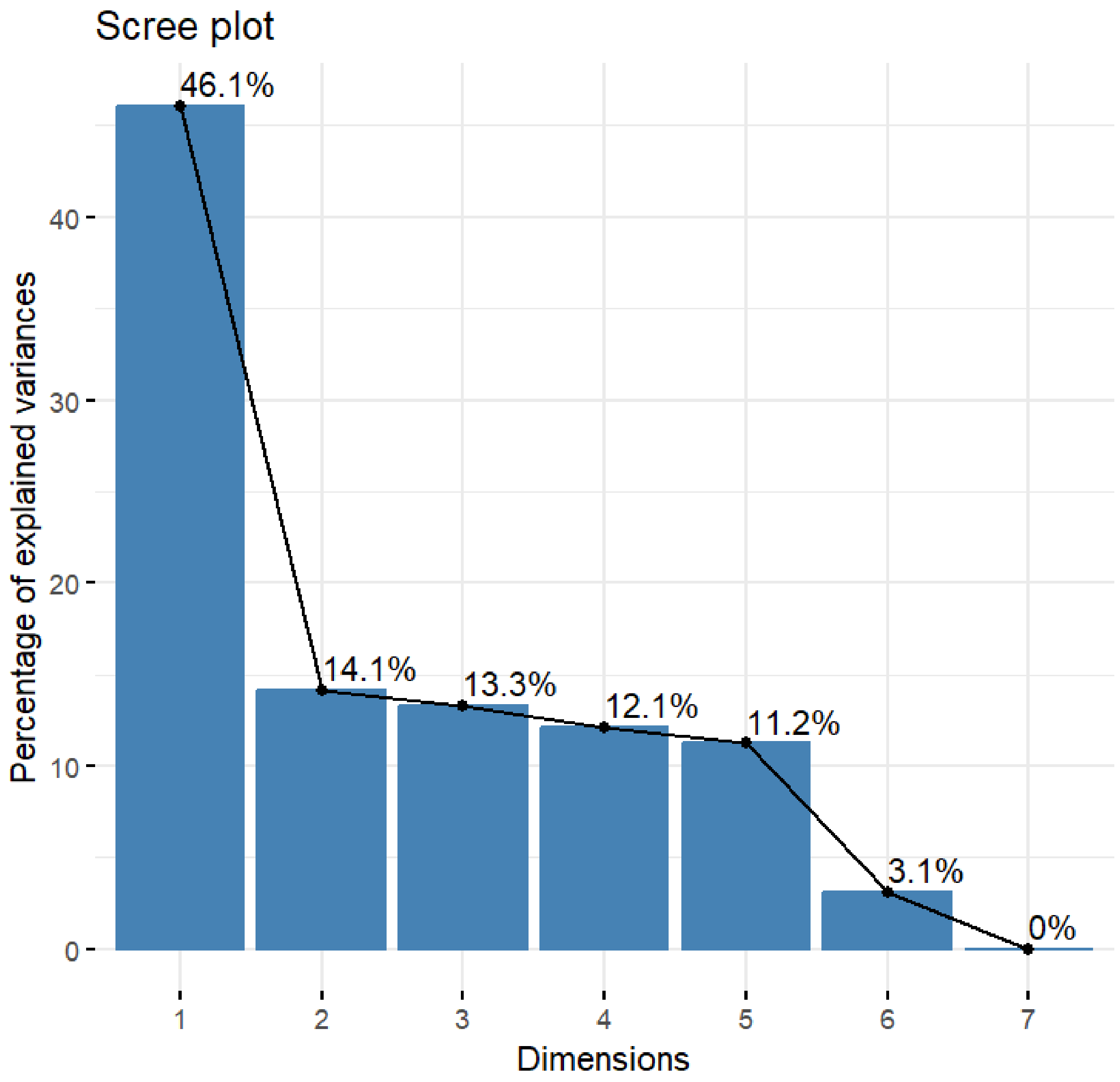

Figure 1: Scree plot of PCA explaining the percentage of variance by each principal component.

As demonstrated in Figure 1, the scree plot shows that PC1 further shows the percentage of explained variance is 46.1 percent, displaying that almost half of the variability in our dataset can be captured by PC1. Now when we look at Figure 1, the line plot shows that we should take into account PC1 and PC2, suggesting that the number of components to retain should be two, giving us the total variance of 60.2 percent.

How we interpret these components in the context of their features is by their loading scores which is shown in Table 2, we can observe that ”Global active power” and ”Global intensity” has a high value,

Table 2: Absolute Load Scores of Principal Components

	PC1	PC2	PC3	PC4	PC5	PC6	PC7
Global_active_power	**0.5547**	0.0022	0.0856	0.0121	0.1148	0.4229	0.7021
Global_reactive_power	0.2115	0.1666	0.4797	0.7490	0.3585	0.0869	0.0158
Voltage	0.2842	0.1889	0.1734	0.5241	0.7416	0.1692	0.0101
Global_intensity	**0.5572**	0.0057	0.0616	0.0055	0.1010	0.4108	0.7119
Sub_metering_1	0.2526	0.6132	0.4231	0.3454	0.3654	0.3582	0.0039
Sub_metering_2	0.3323	0.6745	0.1016	0.1060	0.3684	0.5259	0.0054
Sub_metering_3	0.2855	0.3248	0.7344	0.1831	0.1697	0.4597	0.0063

indicating that both strongly influence PC1.

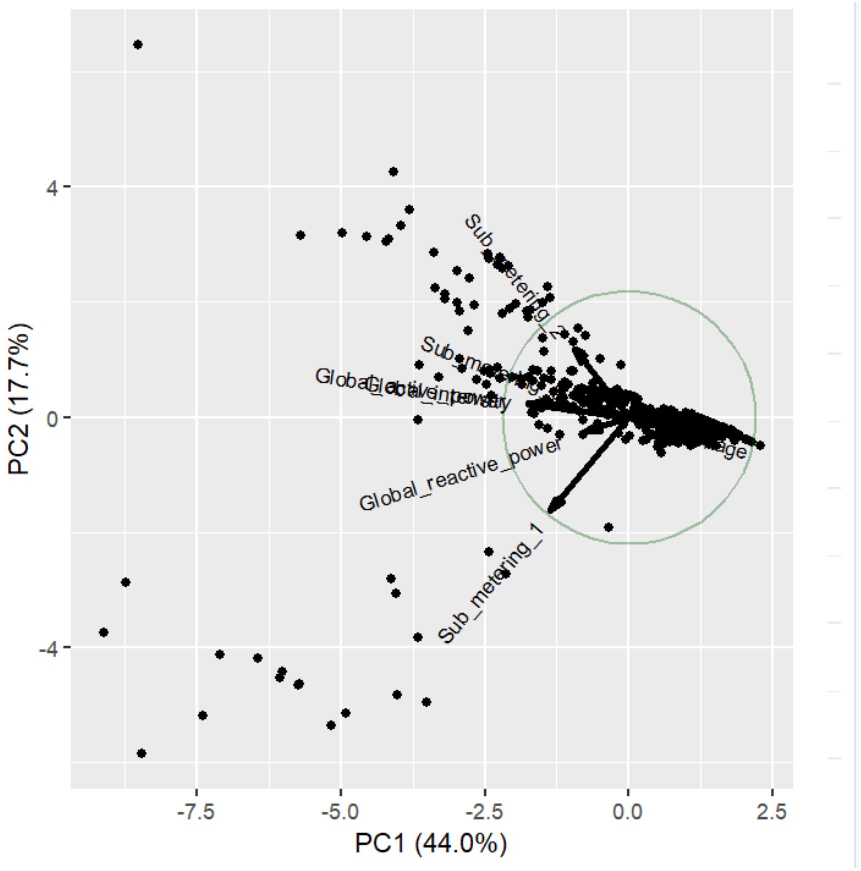

Figure 2: Biplot of the first two principal components showing the projection of variables and distribution of observations.

When displaying this ggbiplot for this PCA, it almost looks like a black "blob" of points and it doesn't allow us to fully see the vector components, so we decided to take a small sample of 400 points to just show the vector components in the plot as shown in Figure 2. To put it simply, the smaller the angle between the two "arrows" the stronger the positive correlation will be between the two variables. From this we can observe that "Global active power" and "Global intensity" arrows are almost overlapping each other which means they have a strong positive correlation between them. This further justifies why we choose just the two features.

4 HMM Training and Testing

4.1 Introduction

4.1.1 Overview of HMM

HMMs are instrumental in the field of anomaly detection due to their distinctive ability to model complex stochastic processes. HMMs operate based on the principle that the system's future state depends only on its current state (the Markov property), making them particularly adept at analyzing sequences of events or observations for anomaly detection purposes. By training HMMs on "normal" behavior, characterized by sequences of observations deemed non-anomalous, they provide a compact representation of process behavior. This representation is crucial for predicting expected behavior and, consequently, for identifying deviations that signify anomalies.

4.1.2 Purpose of HMMs

Hidden Markov Models (HMMs) are adept at encapsulating the underlying dynamics of sequential data which is time-series data. This capability is particularly valuable in this scenario where the system's internal states are obscured or not directly observable. Instead, these states manifest through observable outputs, allowing HMMs to infer the hidden states based on the evidence provided by these observations.

4.1.3 Markov Property

The foundational principle behind HMMs is the Markov property, which posits that the likelihood of transitioning to any future state depends solely on the current state and not on the sequence of events that preceded it. This assumption simplifies the complexity of modeling time-series data by focusing on the current state's influence on future states, reducing the need for historical data. The Markov property enables HMMs to efficiently analyze sequences of events or observations, making them particularly suited for this anomaly detection.

4.1.4 System Analysis with HMMs

The Markov property's application to time-series data becomes particularly relevant when analyzing the patterns of "Global intensity" and "Global active power" from our dataset. These parameters are crucial for understanding the behavior and efficiency of electric power grids, which are part of the broader supervisory control systems our project aims to secure through anomaly detection. The future state of these systems, predicted through the current readings of "Global intensity" and "Global active power," allows us to model the system's normal operational pattern. Given the critical nature of the infrastructure involved, any deviation from this pattern could signify a potential intrusion or system malfunction, necessitating immediate attention.

4.2 Methodology

4.2.1 Model Parameters

Given the structured time-series data from our supervisory control system, we define the Hidden Markov Model (HMM) parameters as follows:

1. **Number of States (N):** The total number of unique states in our HMM, representing various conditions of power consumption inferred from "Global_intensity" and "Global_active_power" patterns.

2. **Number of Distinct Observation Symbols per State (M):** The set of discretized observations derived from the continuous measurements of power consumption variables.

3. **State Transition Probability Distribution (A):** A matrix $A = \{a_{ij}\}$, where a_{ij} is the transition probability from state S_i at time t to state S_j at time $t + 1$, learnt from minute-interval Wednesdays data between 4 and 8 PM.

4. **Observation Symbol Probability Distribution in State j (B):** For each state S_j, a distribution $B_j = \{b_j(k)\}$ specifies the probability $b_j(k)$ of observing the k-th symbol given the system is in state S_j.

5. **Initial State Distribution (π):** The probability distribution $\pi = \{\pi_i\}$ representing the likelihood of the system starting in each state at the initial observation time.

4.2.2 Three Fundamental Problems of HMMs

Within the context of our power consumption analysis project, our Hidden Markov Model (HMM) facilitates the execution of three fundamental tasks:

1. **Likelihood of a Sequence of Observations:** Our model is capable of evaluating the likelihood of sequences of ”Global_intensity” and ”Global_active_power” observations. Given an observation sequence $O = \{o_1, o_2, \ldots, o_t\}$ and the model $\lambda = (A, B, \pi)$, we compute the probability $P(O|\lambda)$. The computational challenge is to consider all possible hidden state sequences to find the likelihood of the observed data.

$$P(O|\lambda) = \sum_{\text{all } Q} P(O, Q|\lambda) = \sum_{\text{all } Q} \left(\pi_{q_1} \prod_{i=1}^{t-1} a_{q_i q_{i+1}} \prod_{i=1}^{t} b_{q_i}(o_i) \right) \tag{1}$$

2. **Determination of an Optimal Sequence of States:** The model seeks the state sequence $Q = \{q_1 q_2 \ldots q_t\}$ that best explains the observations. This is done by finding the sequence that maximizes the joint probability of the observed sequence and the state sequence.

$$Q^* = \arg\max_{Q} \left(\pi_{q_1} \prod_{i=1}^{t-1} a_{q_i q_{i+1}} \prod_{i=1}^{t} b_{q_i}(o_i) \right) \tag{2}$$

3. **Training the Model:** In the training phase, the goal is to adjust the parameters of the model $\lambda = (A, B, \pi)$ to maximize the likelihood of the observed sequence. This is the most crucial step in preparing the model for anomaly detection.

$$\lambda^* = \arg\max_{\lambda} P(O|\lambda) \tag{3}$$

The ‘depmix‘ package employs the Baum-Welch algorithm, a variant of the Expectation-Maximization algorithm, to optimize the training of the Hidden Markov Model by iteratively updating the model parameters to maximize the likelihood of the observed data.

4.3 Results

4.3.1 Data Preparation and Model Training

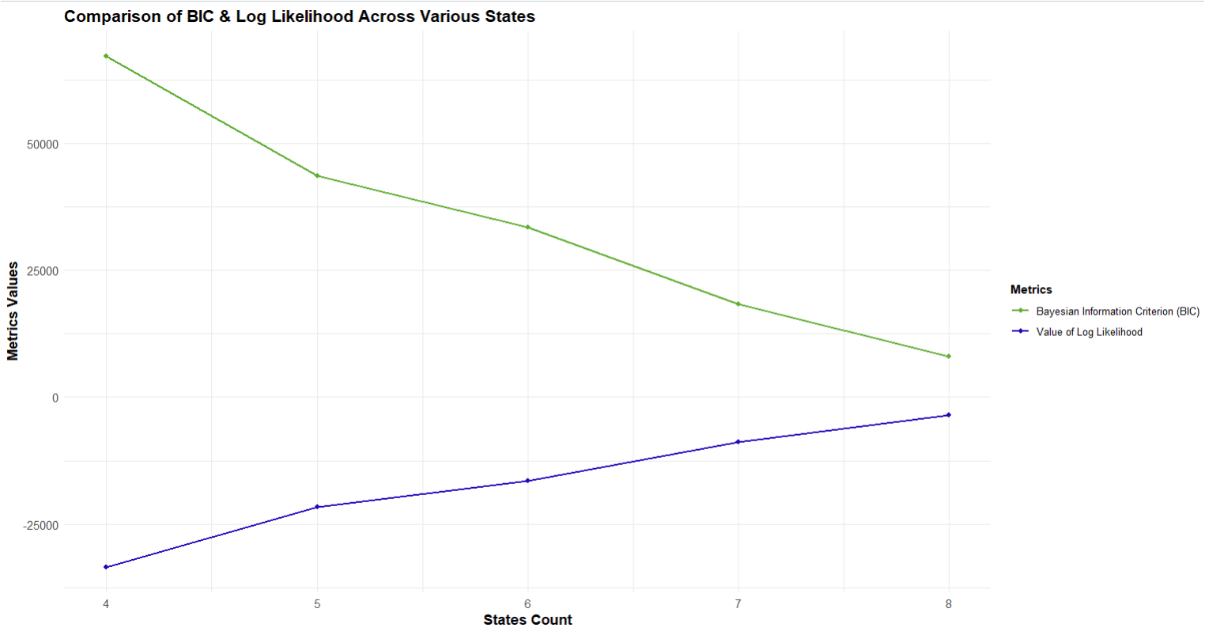

Figure 3: Training model with multivariate distributions

For the effective training and evaluation of our Hidden Markov Model, the dataset was partitioned into two subsets: approximately 80% was used for training and the remaining 20% for testing. To ensure the randomness and generalizability of our model, we employed a random sampling method with a fixed seed for reproducibility. This approach guarantees that the training and test sets are representative of the overall dataset while also maintaining consistency across different runs of the model training process.

Upon preparing our datasets, we initiated the training phase, focusing on the data collected on Wednesdays between 4 PM and 8 PM. This specific time frame was selected based on its relevance to the expected patterns of power consumption. In this crucial step, we aimed to determine the optimal number of states for our HMM that would best capture the underlying structure of the data with the highest log likelihood and lowest BIC value. While we evaluated models with varying numbers of states, we ultimately selected a model with 8 states as the best performer during the training phase with performance of -3527.587 log likelihood. This decision was influenced not only by the model's performance metrics but also by computational constraints as our range of states were from 4 to 8.

4.3.2 Analysis of Model Fit Based on Normalized Log-Likelihood

[H]

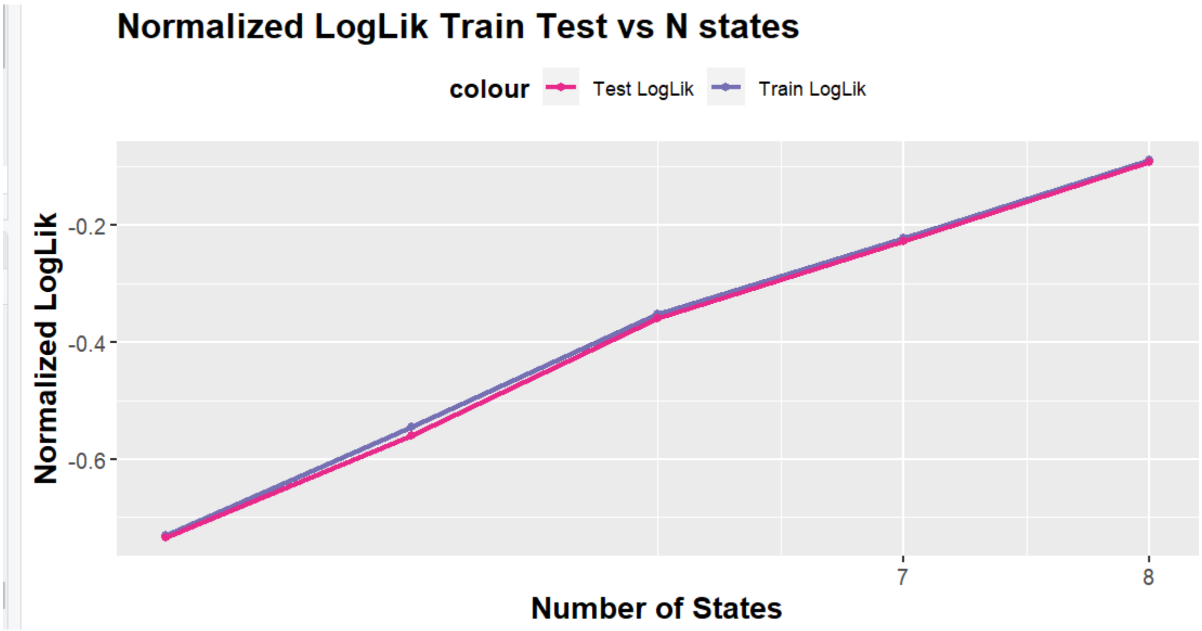

Figure 4: Analysis of Model Fit Based on Normalized Log-Likelihood

Comparing the normalized log-likelihoods for the training and test datasets provides insight into the model's performance across different states. Initially, we observe that the log-likelihood for the training data is lower than that of the test data for states up to 7, suggesting that the model is underfitting.

However, as we progress to states beyond 7, the log-likelihood values converge, indicating a model that fits the data appropriately. This convergence is a hallmark of a well-fitting model, where the training and test log-likelihoods are in agreement, suggesting that the model generalizes well to unseen data. Although computational constraints limited the evaluation to 8 states, the eighth state demonstrates a robust fit. Hence, despite the computational limitations that restricted the number of states we could consider, the model with 8 states performs admirably well, as evidenced by the comparable log-likelihood values, indicating a good fit to both the training and the test datasets.

4.3.3 Using Multinomial Distribution as family

In the context of our anomaly detection framework, we encountered a specific challenge related to the model's distribution family. Initially, our model utilized a positive log likelihood loss, which necessitated a switch to a Multinomial distribution. This transition, however, introduced a parameter mismatch issue during the testing phase. The root cause was identified as the discretization method applied to the variables in assignment 3, which proved to be incompatible with the current dataset. Specifically, the test dataset, being significantly smaller than the training dataset, resulted in disparate groupings due to the original random sampling approach. This discrepancy was observed as missing values within the ranges established by the training sample.

To address this issue, we shifted our sampling strategy from random to stratified sampling, focusing on two key variables: *global_intensity* and *global_active_power*. This strategic adjustment ensured that both training and testing datasets exhibited consistent value groupings, ranging from 1 to 15. Furthermore, to mitigate computational constraints associated with multinomial modeling—known for its intensive computational demands—we adjusted the range of state from an original span of 4 to 8, narrowing it down to 1 to 2. This modification made us able to run the code as we did not have compute power.

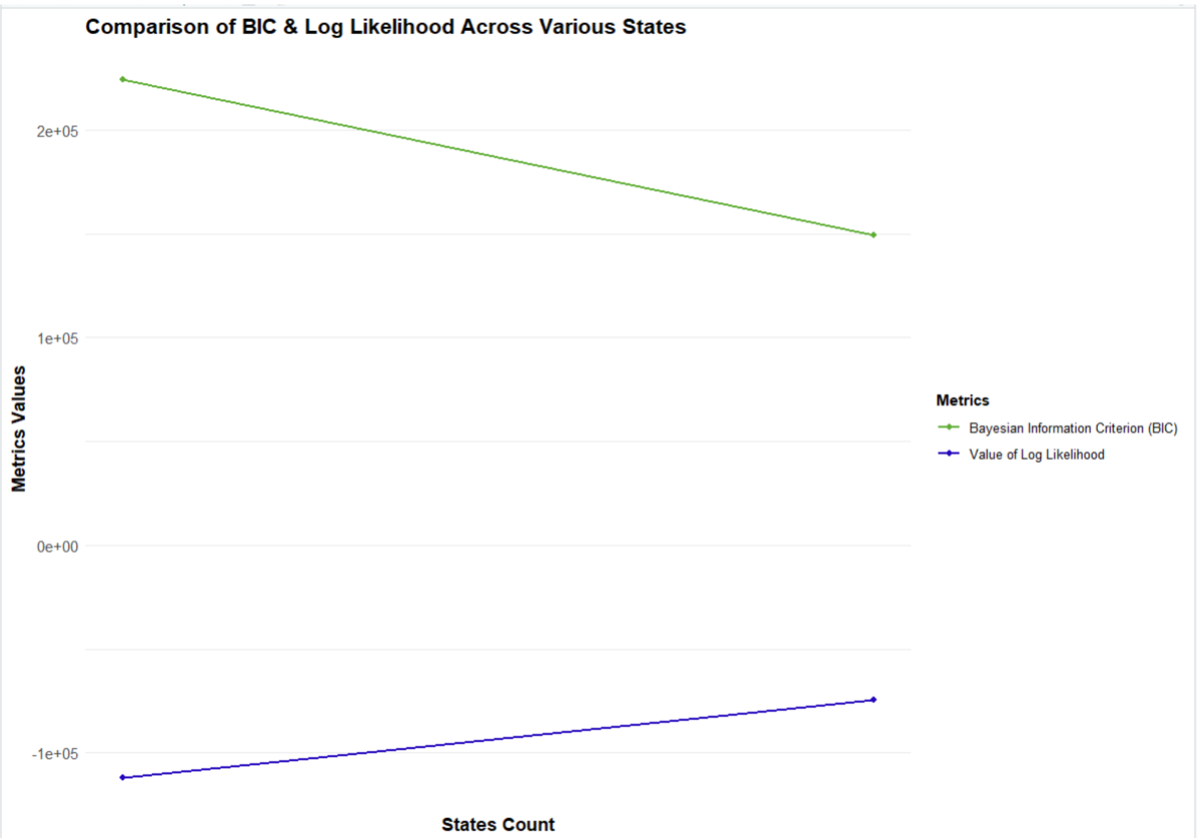

Figure 5: Multinomial training

5 Anomaly Detection

5.1 Introduction

Anomaly detection in time series data is a critical task, especially in mission-critical systems where deviations from expected behavior can indicate potential issues or threat. Identifying the maximum deviation from the normalized train log-likelihood is an important step in evaluating the performance of the anomaly detection model.

5.2 Methodology

From the optimal HMM, the training log-likelihood provides a baseline of normal behavior. To determine normal from anomalous observations, the log-likelihood of new data streams is computed to determine the threshold of deviation from the model's learned behavior. The test data was partitioned into 10 roughly equal sized subsets representing consecutive weeks. Each subset was then assessed using the optimal HMM model to compute the log-likelihoods. The maximum deviation serves as a threshold for identifying anomalies in the test data. Observations with normalized log-likelihood deviations exceeding this threshold are considered anomalous which helps the model in distinguishing between normal and anomalous behavior.

5.3 Results and Findings

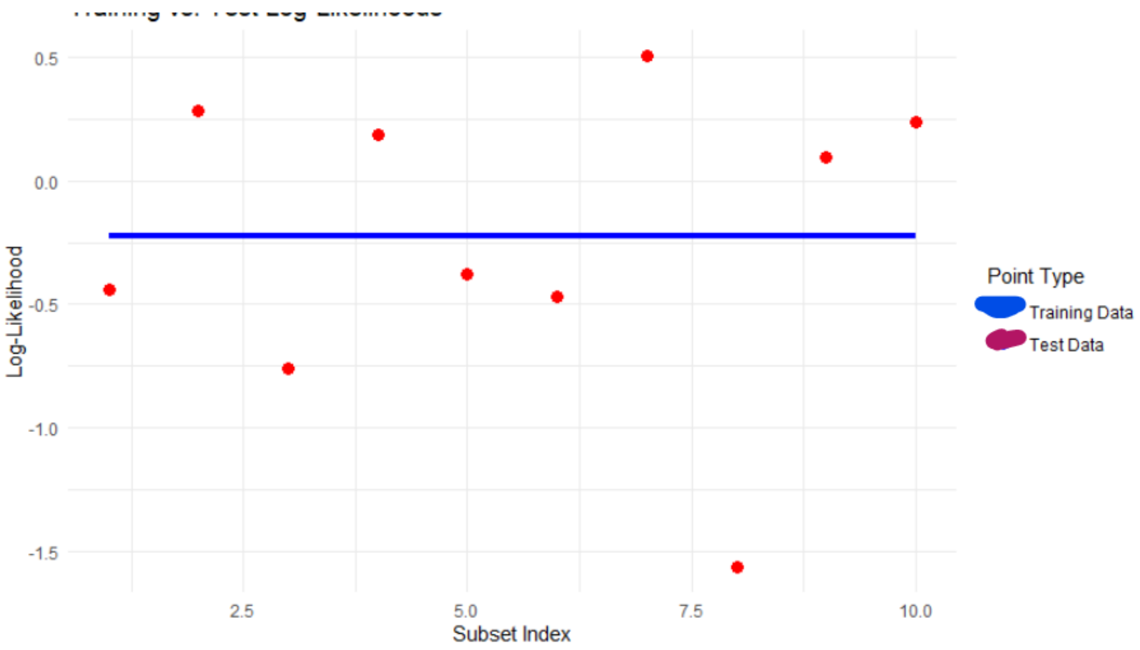

Figure 6: Normalized train log-likelihood vs test log-likelihood with continous variables

With the continuous variable, the normalized training log-likelihood value was -0.222. The table below shows the normalized test log-likelihood numbers for each subset and the deviation to the normalized train log-likelihood. The deviation is calculated by taking the absolute distance between the train and test normalized log-likelihood values.

Table 3: Test log-likelihood values.

	1	2	3	4	5	6	7	8	9	10
TestLog-likelihood	-0.442	0.282	-0.756	0.183	-0.377	-0.468	0.503	-1.561	0.094	0.235
Deviation	0.220	0.504	0.534	0.405	0.155	0.246	0.725	1.339	0.316	0.458

The maximum deviation identified was 1.339 which represents the threshold for acceptable deviation. By comparing the normalized log-likelihood values and analyzing the maximum deviation, it provides insights into how well the model performs on unseen data compared to its training data. The maximum deviation helps in setting thresholds for acceptable deviations, and define normal behavior based on the learned patterns. The figure below plots the normalize test log-likelihood against the train log-likelihood.

Overall, the maximum deviation from normalized train log likelihood is important for evaluating model performance, setting anomaly detection threshold and providing context for interpreting anomalies detected in the test data.

Initially, after running the test data through the model, it yield both positive and negative log-likelihood values. This is because the data is continuous and when using probability density functions (PDF) for continuous variables, it can take on values greater than one. Calculating the log-likelihood involves computing the logarithm of the likelihood function, and as the logarithm of values between 0 and 1 are negative, while values above 1 are positive, positive log-likelihood is primarily observed when using a PDF.

However, using a probability mass function (PMF) for discrete variables, the input is limited to the range [0, 1], since the values cannot be negative and must sum up to one. Given that the variables are discrete, the probabilities are bounded between 0 or 1 which makes the log-likelihood of observing values negative or zero. This impacts the model because it reduces noise, reduces complexity and helps with interpretability. With all these benefits, it helps the model to more accurately capture underlying patterns in the model.

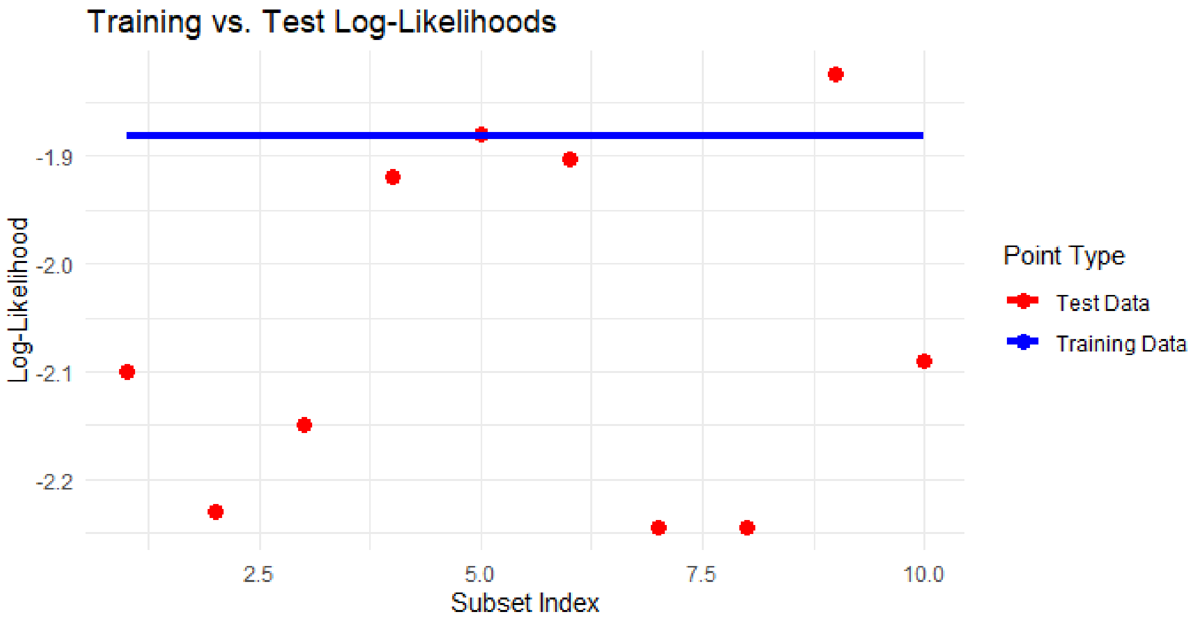

Figure 6: Normalized train log-likelihood vs test log-likelihood with discrete variables

6 Conclusion

Using an HMM, we were able to establish an anomaly 0.485 threshold for discrete variables or 1.339 for continuous variables, that could be used to accurately identify anomalies and distinguish them from normal system behavior and detect intrusions of this particular system.

References

[GGH$^+$22] Michael Greenacre, Patrick Groenen, Trevor Hastie, Alfonso Iodice D'Enza, Angelos Markos, and Elena Tuzhilina. Principal component analysis. *Nature Reviews Methods Primers*, 2:100, 12 2022.