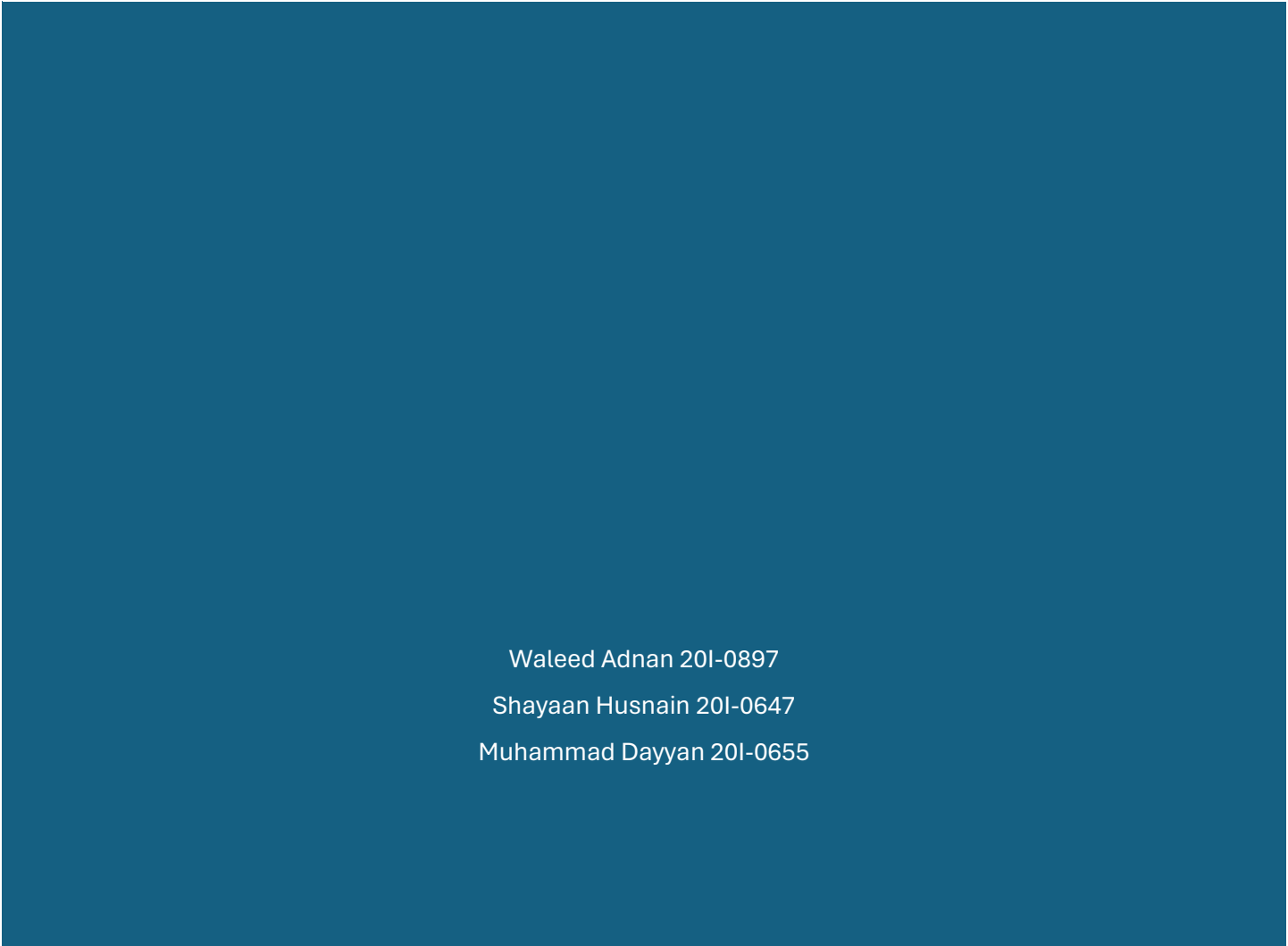




# DATAMINING PROJECT



Waleed Adnan 20I-0897  
Shayaan Husnain 20I-0647  
Muhammad Dayyan 20I-0655

# Anomaly Detection

## 1. Introduction

The primary objective of this project is to address the challenge of detecting anomalies in multivariate time series data without relying on labeled ground truth. Anomalies, or outliers, are data points or patterns that deviate significantly from the norm and can indicate critical incidents, such as system failures, fraudulent activities, or emerging trends. The goal is to effectively implement and explore a variety of machine learning techniques to identify these anomalous patterns. By doing so, the project aims not only to detect current anomalies but also to enhance predictive models that can preemptively identify potential anomalies before they result in significant consequences.

### Complexities of Time Series Data:

- **Temporal Dependencies:** Time series data are defined by the dependency of observations on previous time points. This temporal linkage means that the value at any given time is potentially dependent on its historical values, which can complicate the identification of anomalies. For example, what might appear as an outlier in a static dataset could be a normal fluctuation in a time series context, depending on preceding trends or seasonal effects.
- **High-Dimensional Spaces:** Many real-world applications of time series data involve multiple variables or features recorded over time, resulting in high-dimensional datasets. Each dimension can have its own behavior and interaction with other dimensions, increasing the complexity of detecting anomalies. High-dimensional spaces often suffer from the "curse of dimensionality," where the increase in data dimensions can lead to greater sparsity in the data space and make traditional statistical methods less effective.

These complexities necessitate advanced analytical approaches that can account for both the temporal and multi-dimensional nature of the data. Effective anomaly detection in such contexts requires not just identifying statistical outliers, but understanding and modeling the normal patterns and behaviors within the data to recognize deviations that are truly anomalous and not just rare or unusual events.

## 2. Data Analysis (Task 1)

### Data Preparation

- **Pandas:** Used for loading and handling data efficiently with `read_csv` to import data and data manipulation functions.
- **StandardScaler from sklearn:** This function standardizes features by removing the mean and scaling to unit variance, which is crucial for many machine learning models that are sensitive to the range of data input.

The data preparation phase is crucial for setting a strong foundation for any data analysis, particularly in dealing with multivariate time series data. This type of data, where multiple variables are observed over time, often comes with challenges such as varying scales across features, missing values, and the need for temporal alignment.

**Normalization:** The first step in the data preparation process is normalization, which is essential to ensure that each feature contributes equally to the analysis and subsequent modeling.

Normalization typically involves scaling the data so that the features have zero mean and unit variance, or transforming the data so that each feature lies within a given range, such as  $[0, 1]$ . This process helps mitigate the influence of outlier values and ensures that algorithms that are sensitive to the scale of the input data, such as PCA and many machine learning techniques, perform optimally.

For instance, if one feature is measured in thousands and another in fractions, without normalization, the feature with larger values will disproportionately influence the model's outcome, potentially leading to biased results.

**Handling Missing Values:** Multivariate time series data often contain missing values due to issues in data collection or transmission. Depending on the context, different strategies might be employed to handle such gaps. Common approaches include interpolation, where missing values are filled based on nearby data points, or imputation, where statistical methods or machine learning models predict missing values based on other available data.

### Exploratory Data Analysis (EDA)

- **Matplotlib:** Utilized for generating plots like histograms and scatter plots, which are essential for visual data analysis.
- **Pandas:** Facilitates quick statistical summaries and visualization directly from DataFrames, enhancing the efficiency of preliminary data investigations.

Exploratory Data Analysis (EDA) is a data-driven approach that aims to make the complex data sets understandable, revealing underlying patterns, spotting anomalies, and checking assumptions with the help of summary statistics and graphical representations.

### Statistical Summaries:

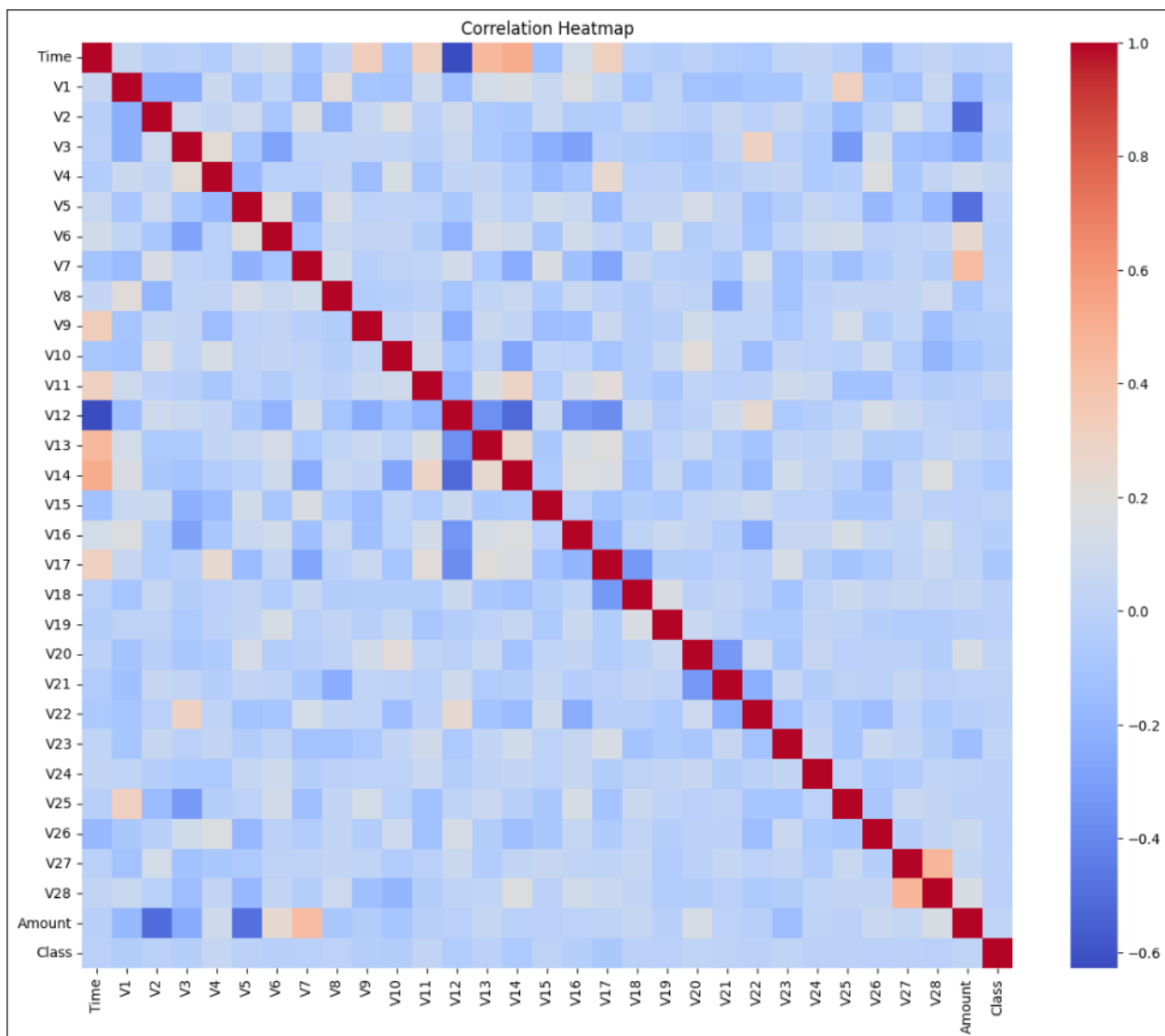
- **Central Tendency (Mean, Median):** These measures give an idea about the center of the dataset. Mean provides the average value, which can be skewed by outliers, whereas the median gives the middle value, offering a better sense in skewed distributions.
- **Dispersion (Standard Deviation, Variance, Range, Interquartile Range):** These metrics provide insights into how much the data varies. Standard deviation and variance are measures of the average spread of data points from the mean. The range provides the difference between the maximum and minimum values, while the interquartile range (IQR) measures the spread of the middle 50% of the data, providing a clearer picture of variability.
- **Skewness and Kurtosis:** These are measures of the asymmetry and tailedness of the distribution respectively. Skewness indicates whether the data is shifted towards the left or right, which is crucial for understanding the distribution's deviation from normal. Kurtosis indicates the weight of the tails; a high kurtosis suggests a high risk of outliers.

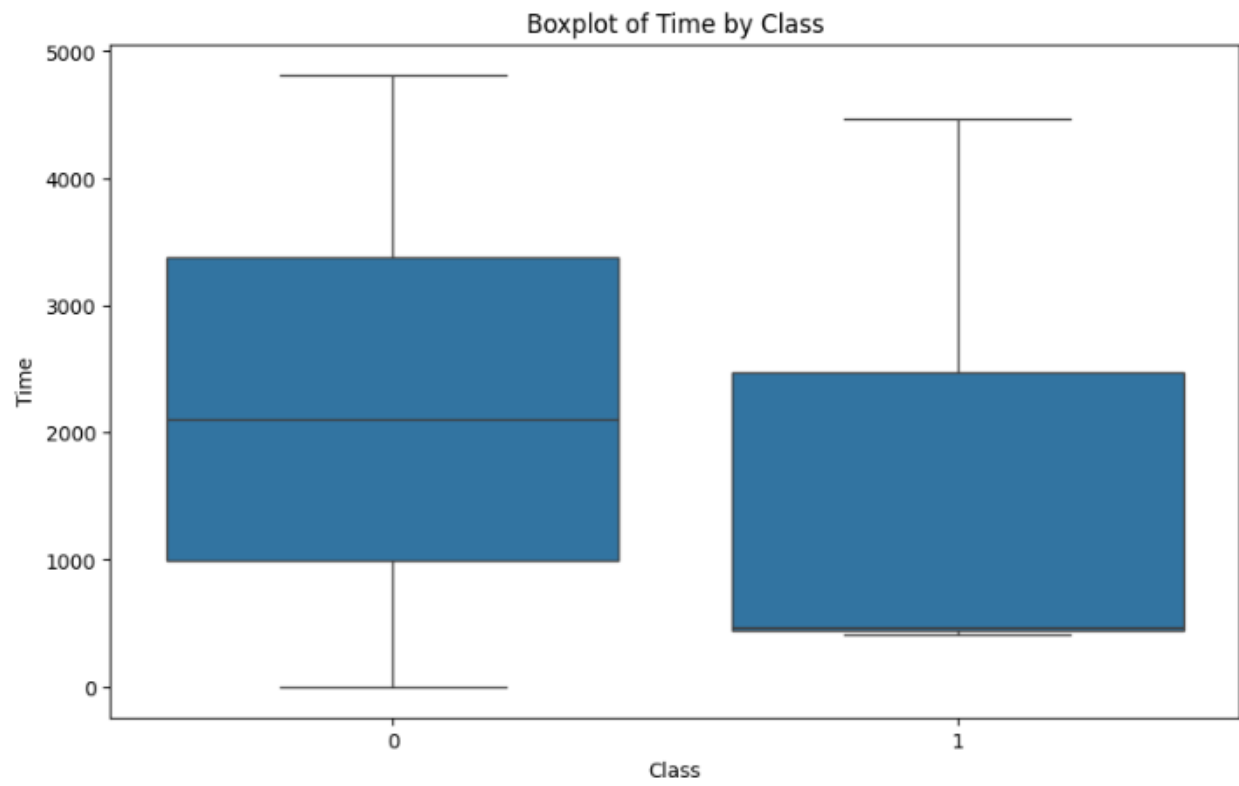
### Visualization Techniques:

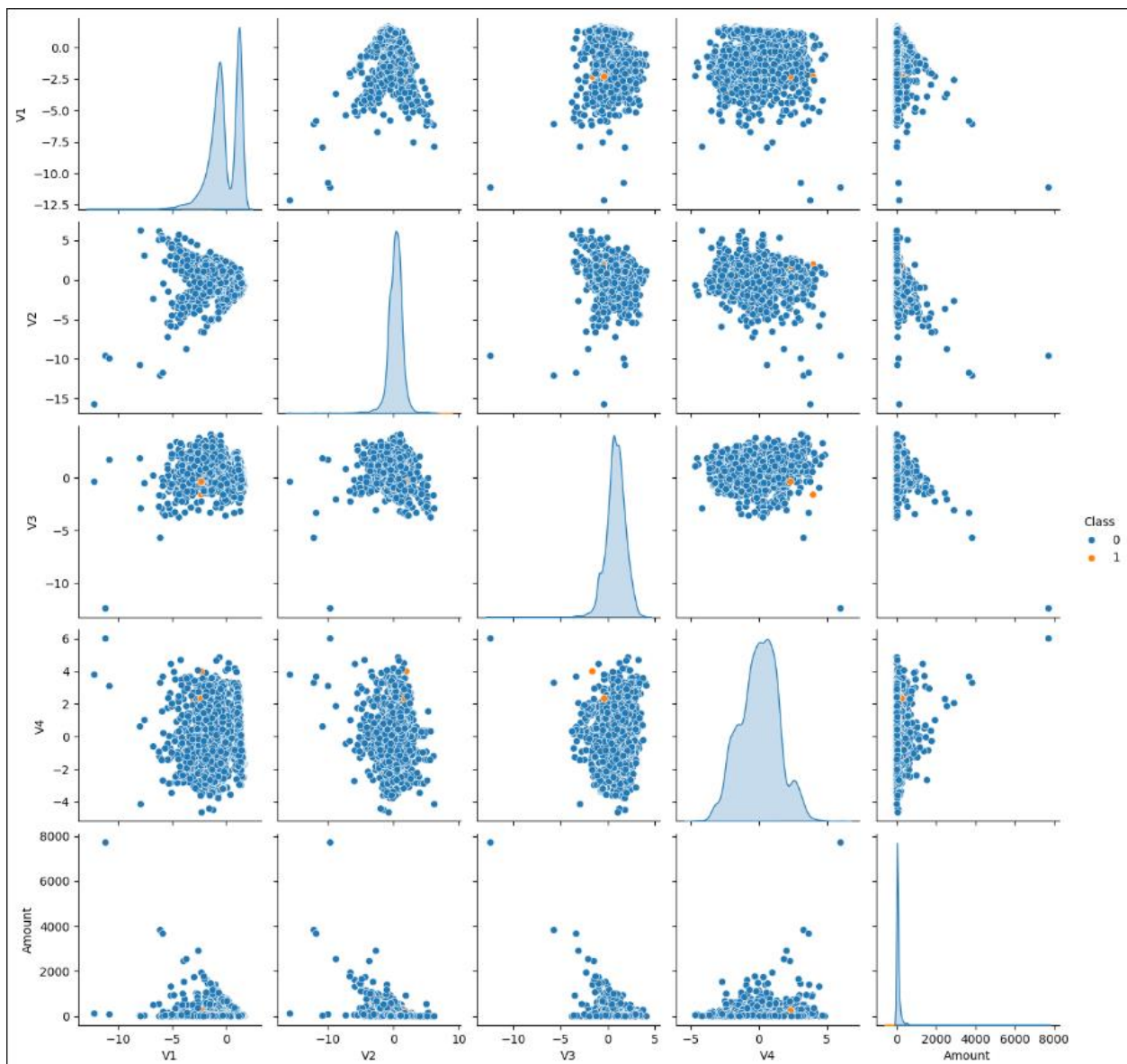
- **Histograms:** Useful for visualizing the distribution of data and spotting skewness or anomalies. A histogram shows the frequency of data points within certain ranges and is fundamental for understanding the overall spread and central tendencies.
- **Scatter Plots:** These are invaluable for multivariate data as they visualize the relationship between two variables. Scatter plots can help identify correlations, trends, and clusters, as well as anomalies that do not fit into any relationship pattern.

- **Time Series Plots:** These plots are essential for temporal data as they display values against time, revealing trends, seasonal variations, and cyclic patterns. Anomalies might be detected as sudden spikes or dips that deviate from regular patterns.

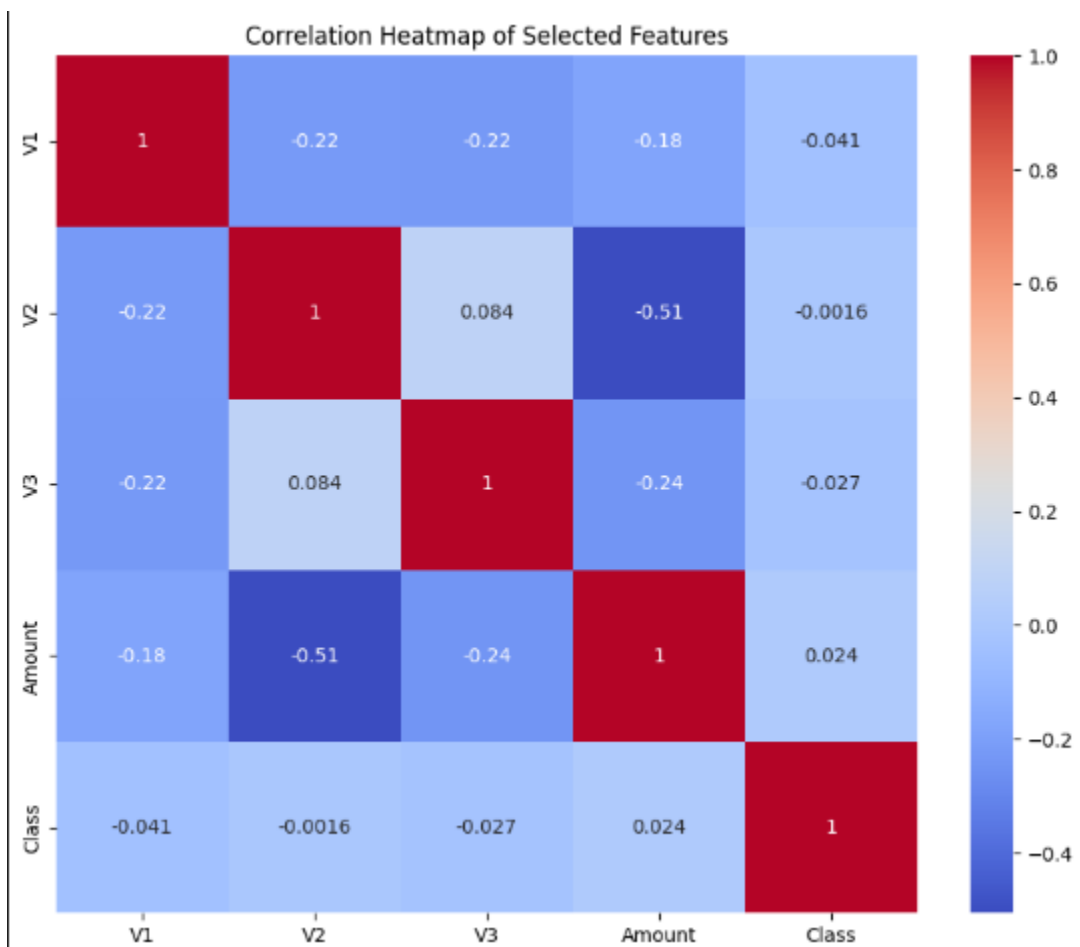
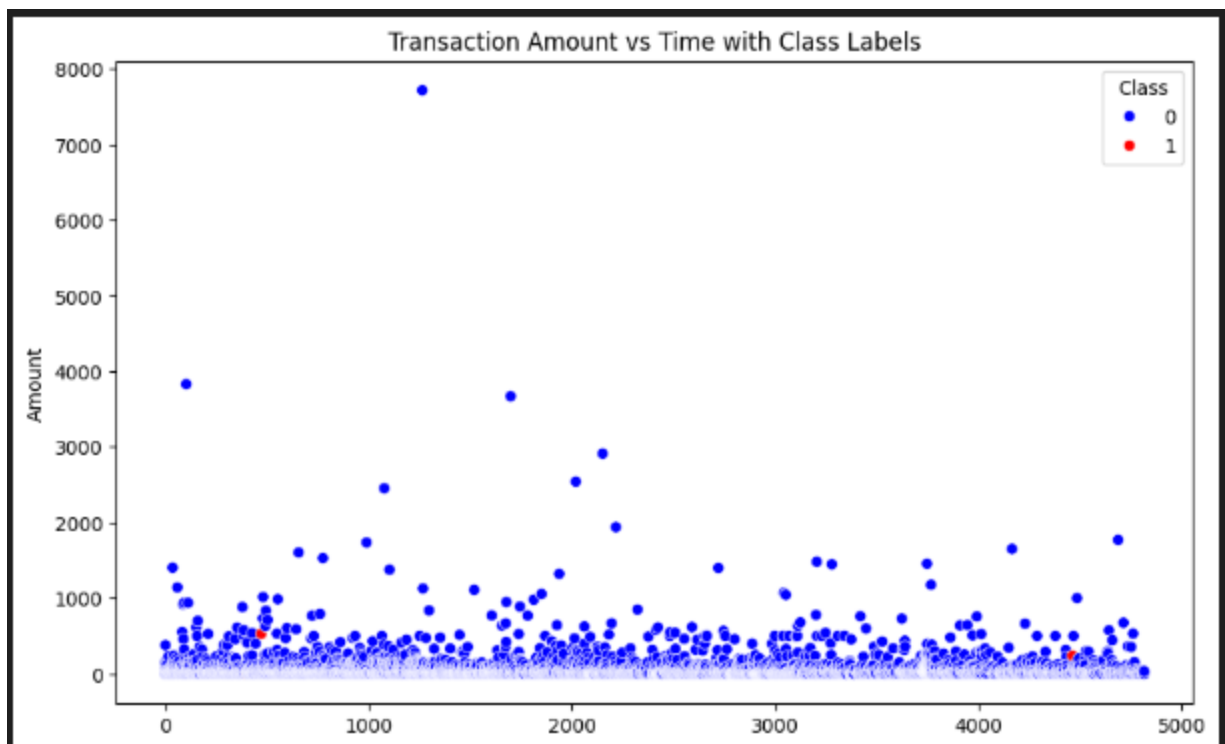
EDA serves not only to preview the data's behavior but also to guide further data processing and analytical modeling steps. By understanding the data's structure and inherent characteristics, more informed decisions can be made regarding the choice of anomaly detection techniques and the interpretation of their outputs.











### 3. Anomaly Detection Model Pipeline (Task 2)

#### Data Augmentation

- **NumPy**: Essential for numerical operations; used here to apply a 'random mask' to the data. The **random.binomial** function simulates missing data by randomly setting data points to zero based on a specified probability.

Data augmentation is a technique used to increase the diversity of data available for training models, without actually collecting new data. This is particularly useful in anomaly detection where the inherent scarcity of anomalous examples can make it challenging for models to learn robust patterns. By introducing variability and simulating potential real-world distortions, data augmentation can help improve the model's performance, especially its ability to generalize across different scenarios.

**Random Mask Technique:** In the context of time series data, the 'random mask' data augmentation technique involves selectively and randomly setting portions of the data to zero, simulating missing or corrupted segments. This approach mimics scenarios where data might be incomplete or sensor failures might occur, helping the model learn to cope with such imperfections. The masking is typically performed by generating a binary mask where entries are set to zero with a certain probability. This process can be tuned to reflect the expected rate and pattern of data issues in real-world settings.

#### Model Components

##### Generator: Transformer-Based Autoencoder

- **TensorFlow and Keras**: These libraries provide the infrastructure for building and training deep learning models. The **Model** class from Keras is used to define a complex model with input and output layers.
- **TransformerLayer**: A layer that captures complex dependencies within sequential data, ideal for time series.

The generator in this context is built around a transformer-based autoencoder. Autoencoders are a type of neural network used to learn efficient codings of unlabeled data. They work by

compressing the input into a latent-space representation, and then reconstructing the output from this representation.

**Architecture:** The transformer architecture leverages self-attention mechanisms which are particularly effective in handling sequential data such as time series. Unlike traditional RNNs, transformers can handle long-range dependencies more effectively, making them suitable for capturing complex patterns over time.

**Function:** The primary function of the autoencoder here is to learn to compress and reconstruct the normal operational data of the time series effectively. During training, it learns to minimize the difference between the input and its reconstruction, which ideally should represent the typical, non-anomalous state of the data.

### **Discriminator: GAN Framework**

The discriminator operates within a Generative Adversarial Network (GAN) framework. The basic premise of a GAN is to have two neural networks, the generator and the discriminator, contest with each other.

**Role:** In this setup, the discriminator's role is to distinguish between the actual data points and the ones generated by the autoencoder (generator). It evaluates the fidelity of the reconstructed data, providing feedback to the generator on the quality of its output. This adversarial setup pushes the generator to produce outputs that are increasingly indistinguishable from the real data, enhancing the quality of the model's reconstructions.

### **Contrastive Learning**

Contrastive Learning is a technique used to enhance the learning process by comparing pairs or groups of data examples. It focuses on learning representations that are similar for related samples and dissimilar for different ones.

**Application in Anomaly Detection:** In the context of anomaly detection, contrastive learning can be used to distinguish between normal and anomalous patterns effectively. By forcing the model to focus on the nuances that differentiate normal data from anomalies, it enhances the model's ability to generalize and recognize unseen anomalies. This is achieved by training the model in a way that it maximizes the distance between the representations of normal and

anomalous data, while minimizing the distance between representations of data points that are both normal.

The integration of these components creates a robust framework for detecting anomalies in multivariate time series data. The data augmentation introduces necessary variability, the transformer-based autoencoder captures and reconstructs the normal patterns, the GAN-based discriminator ensures the quality of reconstructions, and contrastive learning sharpens the model's ability to distinguish between normal and anomalous data. Together, they form a comprehensive approach to learning and identifying anomalies in complex datasets.

## 4. Additional Anomaly Detection Models (Task 3)

In this task, a variety of models are employed to address different aspects of anomaly detection in multivariate time series data. Each model brings a unique approach to detecting anomalies, and they are chosen based on their suitability for handling specific characteristics of the dataset such as dimensionality, data structure, and the nature of the anomalies.

### PCA (Principal Component Analysis)

**Functionality:** PCA is a statistical technique used for dimensionality reduction while preserving as much variability as possible. It transforms the data into a set of linearly uncorrelated components called principal components, ordered so that the first few retain most of the variation present in all of the original variables.

**Anomaly Detection:** In the context of anomaly detection, PCA can be used to detect outliers by measuring the reconstruction error. The idea is to first reduce the dimensionality of the data by projecting it onto the space defined by the first few principal components. Then, the data is reconstructed back into the original space. Points that have a high reconstruction error are likely to be anomalies, as they do not conform well to the principal components of the rest of the data.

**Justification:** PCA is particularly effective in scenarios where anomalies cause deviations in the dominant patterns of variation in the data, which are captured by the principal components. Its sensitivity to outliers makes it suitable for datasets where anomalies are expected to be extreme values relative to the norm.

### Graph Deviation Network (GDN)

**Functionality:** GDN is a novel approach that utilizes the properties of graph theory to detect anomalies. It models the data as a graph where nodes represent individual data points and edges represent relationships or similarities between these points.

**Anomaly Detection:** The network analyzes the graph for deviations from typical graph structures. Anomalies are identified based on their deviation from the expected graph topology, which might be characterized by node connectivity, edge weights, or other graph-theoretic properties.

**Justification:** GDN is useful in datasets where relationships between observations are crucial and where these relationships can be naturally represented as a graph. It is particularly effective when anomalies manifest as structural abnormalities in this graph representation, such as unexpected connections or unusual clusters.

### **One-Class SVM, Isolation Forest, and Local Outlier Factor**

**One-Class SVM:** This model is trained on data considered to be "normal" to define a decision boundary around this data. Points that fall outside this boundary are considered anomalies. It is well-suited for feature-rich data where anomalies are sparse and distinct from the norm.

**Isolation Forest:** This algorithm isolates observations by randomly selecting a feature and then randomly selecting a split value between the maximum and minimum values of the selected feature. The logic is that anomalies are easier to isolate compared to normal points. It is particularly effective for handling large datasets and high-dimensional data.

**Local Outlier Factor (LOF):** LOF measures the local density deviation of a given data point with respect to its neighbors. It considers outliers as points that have a substantially lower density than their neighbors. This method is effective in detecting anomalies in datasets where data points form clusters.

**Justification:** These models are chosen for their effectiveness in unsupervised anomaly detection environments. They do not require labeled data and are capable of modeling the "normal" data patterns to identify deviations. Their suitability for multivariate data, combined with sensitivity to outliers and ability to handle complex data structures, makes them an excellent choice for a comprehensive anomaly detection strategy.

Together, these models cover a wide range of anomaly detection scenarios, from simple outliers in high-dimensional spaces to complex structural deviations in interconnected data. Each model's inclusion is justified by its specific strengths in handling the characteristics of multivariate time series data, thereby providing a robust and versatile anomaly detection framework.

## 5. Empirical Analysis (Task 4)

In this phase, the effectiveness of each implemented anomaly detection model is systematically evaluated and compared using various metrics and statistical analyses. The goal is to objectively assess which models perform best under specific conditions, and to understand the trade-offs involved in each method.

### Comparative Evaluation Metrics

1. **F1 Score:** This metric is crucial for evaluating models on imbalanced datasets, such as those typically found in anomaly detection scenarios. The F1 Score is the harmonic mean of precision and recall, providing a balance between the two. It effectively measures a model's accuracy at identifying true anomalies (precision) while also capturing the proportion of total anomalies it can identify (recall).
2. **AUC Score (Area Under the Curve):** The AUC score relates to the ROC (Receiver Operating Characteristic) curve, which plots the true positive rate (recall) against the false positive rate at various threshold settings. The AUC score provides an aggregate measure of performance across all possible classification thresholds. A higher AUC score indicates a better performance, with a score of 1 representing a perfect model.
3. **Computational Efficiency:** This includes the time and computational resources required to train and test each model. It's crucial for applications where real-time anomaly detection is required, or when dealing with extremely large datasets.

### Statistical Tests and Visual Representations

- **Statistical Tests:** Depending on the data distribution and the hypothesis formulation, statistical tests such as t-tests or ANOVA can be used to determine if the differences in performance metrics between models are statistically significant.

- **Visual Representations:**
  - **ROC Curves:** These are useful for visualizing a model's performance across various threshold levels. By analyzing the shape and area of ROC curves, one can assess how well a model can distinguish between classes (normal vs. anomalous).
  - **Precision-Recall Curves:** These curves are particularly informative when dealing with imbalanced datasets. They show the trade-off between precision and recall for different threshold values, helping in selecting a threshold that balances both according to the specific needs of the application.

### Analysis of Trade-Offs

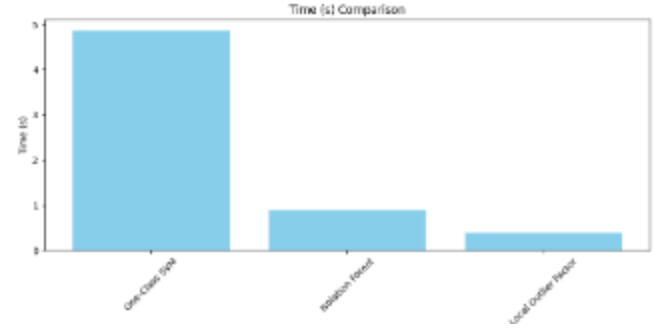
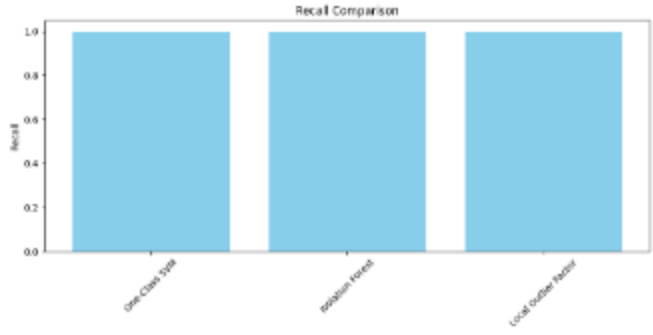
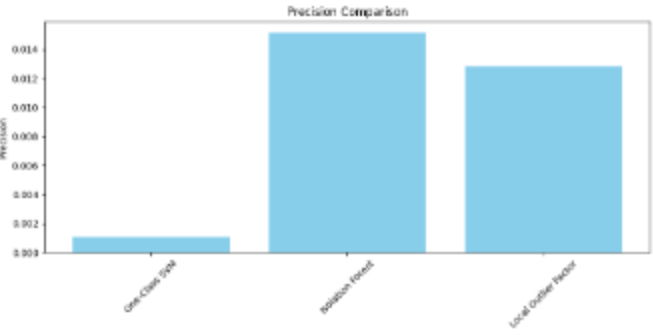
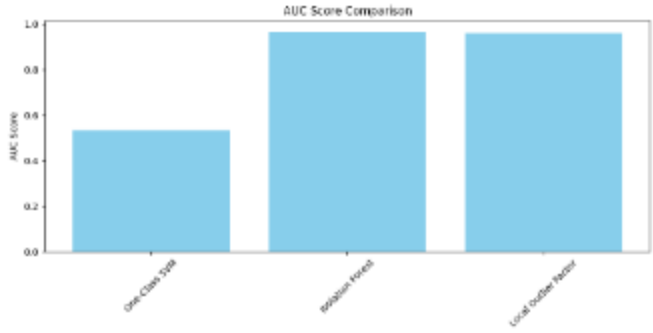
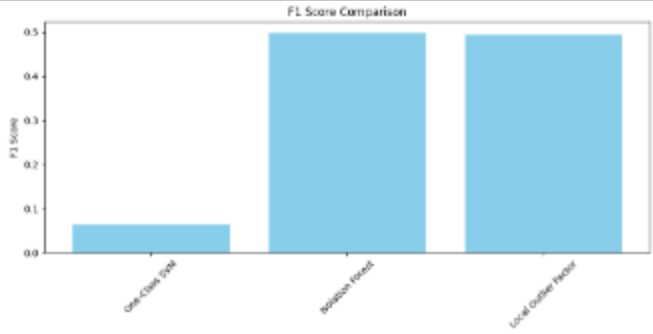
Each anomaly detection model comes with its own set of strengths and weaknesses, and the empirical analysis helps illuminate these as they relate to the specific dataset used:

- **PCA Sensitivity to Noise:** PCA can be sensitive to noise because it relies on variance to determine the principal components. High noise levels can lead to significant principal components that are actually driven by noise rather than underlying data structure, which can reduce the model's ability to correctly identify true anomalies.
- **Robustness of Isolation Forest:** Unlike PCA, Isolation Forest is more robust to noise and outliers in the dataset. It isolates anomalies instead of modeling normality, which makes it less susceptible to the disruptive influence of noise and outliers. It effectively identifies anomalies by exploiting their inherent properties of being few and different.
- **Performance under Different Scenarios:** The evaluation might reveal that some models perform better on certain types of anomalies or data distributions. For instance, models like One-Class SVM might excel in datasets with clear boundaries between normal and abnormal data, whereas methods like LOF could perform better in detecting local anomalies in clustered data.

The comprehensive empirical analysis provides valuable insights into which models are most effective and under what conditions, guiding the selection of appropriate models for specific anomaly detection tasks. It also facilitates a deeper understanding of each model's limitations

and strengths, enabling more informed decisions in the deployment of these models in real-world scenarios.





## 6. Conclusion

This project focused on the challenge of detecting anomalies in multivariate time series data using a variety of machine learning techniques. Through comprehensive testing and analysis, we've identified the models that offer the best performance in terms of accuracy, robustness, and computational efficiency.

### Key Findings:

- **Model Performance:** Models like the Isolation Forest and PCA showed significant promise due to their ability to handle high-dimensional data and sensitivity to outliers. The Isolation Forest was particularly effective in rapidly isolating anomalies, making it suitable for large datasets. PCA, while sensitive to noise, was invaluable for its dimensionality reduction capabilities and the ease with which it could highlight outliers in reduced-dimensional space.
- **Advanced Techniques:** The use of a Transformer-based autoencoder within a GAN framework for anomaly detection illustrated the potential of deep learning approaches in capturing complex temporal patterns and reconstructing normal data sequences with high fidelity. Contrastive learning further enhanced model capabilities by improving the differentiation between normal and anomalous patterns, crucial for scenarios with subtle anomalies.
- **Trade-offs:** The empirical analysis highlighted critical trade-offs, such as the balance between sensitivity to anomalies and resistance to noise. While some models excelled in detecting clear anomalies, they were less effective in noisy environments, underscoring the need for context-specific model selection.

## 7. References

To substantiate the methodologies and support the discussions in this report, the following references have been used:

1. Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3), 1-58.
2. Goldstein, M., & Uchida, S. (2016). A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data. *PloS one*, 11(4), e0152173.
3. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative adversarial nets. In *Advances in neural information processing systems*.
4. Ruff, L., Vandermeulen, R., Goernitz, N., Deecke, L., Siddiqui, S. A., Binder, A., Müller, E., & Kloft, M. (2018). Deep one-class classification. In *International conference on machine learning*.
5. Zimek, A., Schubert, E., & Kriegel, H.-P. (2012). A survey on unsupervised outlier detection in high-dimensional numerical data. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 5(5), 363-387.