Universiteit Leiden

Master Computer Science

Logistics Anomaly Detection

Name:	Bowei Gao
Student ID:	s3400654
Date:	March 17, 2024
Specialisation:	Data Science
1st supervisor:	Prof.dr. S.W. Pickl
2nd supervisor:	Pro. Yingjie Fan
project supervisor:	Dr. Truong Son Pham

Master's Thesis in Computer Science

Leiden Institute of Advanced Computer Science (LIACS)
Leiden University
Niels Bohrweg 1
2333 CA Leiden
The Netherlands

Abstract

This thesis explores the problem of anomaly detection in the logistics domain. the anomaly detection problem can be viewed as a problem of categorizing logistics records into anomalous and normal records. The classification will then be performed using multiple supervised learning classification methods and unsupervised methods. The reasons for using them will be explained, the performance and speed of these methods will be shown and compared, and the causes will be analyzed. It is found that on simpler logistic datasets either supervised or unsupervised learning methods can be used; however, on more complex datasets the guidance of labels is required, and therefore only supervised learning methods can be used. At the same time, an unsupervised logistics anomaly detection method that combines autoencoders and clustering methods is proposed, which is expected to improve the speed and accuracy of the detection. Finally, based on the feature importances obtained from supervised learning, factors that have a greater impact on whether the logistics arrive on time or not are presented and analyzed.

Keywords: logistics, anomaly detection, clustering methods, autoencoders, delay, ontime, one-class classification

Contents

Acknowledgment

This thesis marks the end of my two years as a master's student. Looking back at the past two years, there are so many people whose names I know or don't know who have given me help and support.

As a young person coming from another continent to study in Europe, I was surprised by the understanding and care I received from teachers and classmates from Netherland and other countries. I not only practiced my English in the classroom and through collaboration and activities with my classmates, but I also learned a lot of things that I missed in the East Asian classroom. Although I am not as good at presentations as my classmates, when I volunteered, they did not refrain from supporting me for fear of getting a lower grade. The reports I wrote in the beginning may have lacked structure and I was not as proficient with Office tools as my foreign classmates. Students who have worked with me before understand and give me kind and honest advice. The list goes on and on. When I didn't know how to behave at activities, my classmates would warmly greet me and introduce me to others. Gradually, I started to learn how to work in Europe and enjoy my life here with them. Among the many classmates, I would like to give a special thanks to Rohan Pastricha, Surya Parthipan, Pablo Zheng, Lili Darabi, and Jesse de Gans for their help and understanding.

I am also grateful to have had the honor of having Professor Pickl and Doctor Pham as my first and second mentors. Professor Pickl did more than tutor my thesis. He also helped me find a job as a software engineering intern in Germany when I mentioned that I wanted to start my career in Europe. Although, Professor Pickl is very busy with many jobs, he has been tutoring me through the thesis process and has been very encouraging when I have had difficulties. He also helped as much as he could when I raised hopes that might be difficult to do. During the course of my internship, my supervisor often mentioned how helpful Professor Pickl was to her as well. He is such a kind person!

I would like to thank Professor Pickl very much for introducing me to Doctor Pham and for the privilege of having him as my project supervisor. Doctor Pham pointed the way for my entire master's graduation program. Not only did he take me through the entire process of completing my research project, but he also gave me the right guidance when I was not lost and not sure what to do next. I was able to keep the project moving in the right direction without losing the opportunity for scientific exploration. Although Doctor Pham is also very busy with his day-to-day work, when I contacted him to discuss the results and understanding of what had been accomplished and to inquire about the next steps, he always responded within a short period of time and gave me a slot to meet with him in the near future. I remember when his daughter was sick, I was still embarrassed to email him about the next meeting because I was anxious about the progress of the experiment. He was very understanding and took time out of his busy schedule to help me. He has deep knowledge and authority in the field of data science and logistics, yet was always approachable in my discussions. When I didn't understand what he was telling me, he would go the extra mile to make me understand. I am truly grateful to Dr. Pham, without whom I would not have been able to complete my master's thesis project.

I would also love to thank my superiors during my internship at HOLM GmbH, Maryna Zharikova and Jakob Grubmueller, whose professional attitude and expertise were key to the success of my project implementation. They also gave me full confidence and freedom to choose technical solutions. Even though I didn't know much about using the Java Spring Framework to build the back-end of a website at that time. They gave me the opportunity to work and allowed me to use this tech stack. At the same time, they provided me with an

informative explanation of logistics and attracted the support of DPD, which enabled me to achieve unexpected results in my internship program.

Last but not least, I would like to thank my family in Beijing for the financial and emotional support they have given me in the past educational career as well as in these two years. If it were not for the hard work of my parents, coming from a developing country, I may not have had the opportunity to study in Europe, much less pursue my life's ambition and passionate career path in computer science without having to consider financial constraints. Without them, I would not have been able to come to the Netherlands and meet so many wonderful people, and I would like to thank my family for keeping in touch with me and letting me know that even though I am far away, I have a stable and loving family behind me. It also makes me feel safe in any situation and to believe in the kindness and friendliness of everyone. I love you, papa and mama!

1 Introduction

1.1 Importance of anomaly detection in logistics

Logistics costs account for about 13% of countries' GDP on average, and even more of the cost of commodity prices [1]. In a world where inflation is making the living cost of families higher and causing great trauma to the daily lives of families. It would be desirable and be of great benefit for all stackholders to lower this cost. In addition, according to the same paper, logistics costs account for a higher percentage of total commodity prices in less developed countries where logistics efficiency is lower. For example, it makes up 8 percent of GDP where logistics is the most efficient such as countries like the United States and the Netherlands. However, it makes up more than 25% in countries where logistics efficiency is the worst. And those countries with the worst efficiency are also the least developed nations. So improving logistics efficiency is about helping every suffering citizen. But it will be of more help people who are suffering the most. So it's also about promoting fairness in a world that's becoming more and more unbearable for the poor. In today's energy transition, the rising logistics costs resulting from the use of new energy sources and the inflation it causes are also one of the main arguments of green energy opponents. Therefore, reducing logistics costs and improving the stability of the logistics system will also help to accelerate the adoption of new energy sources. Overall, there are two benefits to improving logistics efficiency and reducing logistics costs, which is why I wanted to do this topic.

1. Lowering commodity prices.
2. Helping energy transition.

In this thesis, I want to explore the possibility of using the knowledge empowerment of data science that I learned in my master's to help improve the efficiency of logistics operations. It's found that a significant factor in logistics costs is the occurrence of anomalies. For example, when an item is delivered late, all of the processes behind it can't work properly and have to be adjusted. Just like during the pandemic, many important parts needed for automobile production could not be delivered properly because of the lockdowns. This took a heavy toll on the economy, and the price of automobiles everywhere skyrocketed as a result, leading directly to a rise in the cost of living for people. For example, in war, many ships need to go around specific locations, which not only leads to increased time and costs but also increases carbon emissions during transportation. The economic and environmental impacts are significant. But there are also many situations in which much less visible things affect the logistics network. Although it's sometimes not possible to avoid the occurrence of accidents, if anomalies in the logistics system can be detected, this will enable logistics companies to deal with the anomalies as early as possible to avoid further losses. It's also possible to anticipate which logistics routes are more prone to anomalies and give them more time in the system so that these anomalies don't affect the next steps. Better yet, it's also possible to backtrack through anomaly detection to determine what factors are greatly influencing tasks to be anomalies and focus on improving those aspects of the logistics operations to truly improve the efficiency and reliability of logistics. So the focus of this master's graduation project is how to use data science tools for anomaly detection in logistics operations.

Support Vector Machine	Logistic Regression	Decision Trees
Random Forest	Gradient Boosted Trees	XGBoost

Table 1: Supervised methods that I used

1.2 The experiment methods

The problem of logistics anomaly detection can be viewed as a problem of categorizing a logistics dataset. Detection methods will categorize logistics records into two categories, which are normal data records and anomalous data records. Ideal anomaly detection can correctly categorize anomalous logistics records in a dataset with a mixture of anomalous and normal data records as anomalous logistics records without categorizing the other logistics records as anomalous, i.e., categorizing normal data records as anomalies.

1.2.1 Anomaly detection methods used by previous articles

When selecting the methods to be used for logistics anomaly detection, a large quantity of literature is referred to. Survey papers are a good starting point, The article [2] categorizes the methods of anomaly detection and summarizes representative articles of the methods within each class, which is very informative. Articles [3], [4], [5], and [6] all mentioned anomaly detection using categorization, consistent with my understanding and assumptions about anomaly detection here. Among these, articles [3], [4], and [6] mentioned using clustering for classification, and since this thesis mostly focused on anomaly detection in the context of training on unlabeled data, these articles piqued my particular interest. [7] It discusses the application of deep-learning-based methods in anomaly detection and it focuses on deep-learning models including CNN, RNN, and LSTM. The types of data records it deals with are mainly images, sensor data, video, speeches, and other content. They are way too complex in comparison to logistics data records, so deep-learning-based methods are not used in this project.
Reading through the literature, it is found that classification methods can be categorized as multi-class classification and one-class classification. Muti-class [8] [9] is no different from what we normally use in other supervised classification cases and contains labels that mark different classes (here the labels are "anomalous" and "normal"). However, the one-class classification methods are a new concept for me. Since they are unsupervised, they draw the attention of this project. Common one-class methods are one-class svms [10], one-class Kernel Fisher Discriminants [11] [12], and one-class autoencoders. Many other articles mentioned doing anomaly detection without using labels, for example, [13], [14], [15], and [16].

1.2.2 The methods in my experiments

To achieve the goal of categorizing records as normal and anomalous, it is intuitively best to use a supervised learning approach to train a classification model using "anomaly" and "normal" labels. This is the kind of method I first adopted in my research. Supervised learning methods in Table 1 are used to train anomaly detection models on logistics records and they achieved good accuracy. However, the problem is that by reading the paper, it is realized that in most cases, it is difficult to obtain the labels in practical applications. Most of the datasets used in the relevant papers were also unlabeled. Labeling anomalous data records by hand requires a lot of expertise as well as time, effort, and money in practice.

one-class SVM	KMeans	hierarchical clustering	HDBSCAN
one-class autoencoders	BIRCH	spectual clustering	OPTICS

Table 2: Unsupervised methods that I used

So I decided to use the perfermance of the supervised learning method used in Table 1 as a benchmark so that I can see if the unsupervised learning methods could be used for anomaly detection, and compare how accurate the unsupervised learning method was. After reading the literature and my supervisor's suggestions, I adopted two one-class classification methods and multiple clustering classification methods as the unsupervised methods on which I focused my research and which can also be used as a base benchmark to measure the performance of my proposed methods. The unsupervised learning methods I used are listed in Table 2.

It is expected that combining autoencoder with clustering and utilizing the preprocessing features of autoencoder can improve the speed and accuracy of clustering methods. A specific description and explanation of this model can be found in Chapter 5. So this model is proposed and will be compared with supervised learning methods and other unsupervised methods to see if it can get the results that improve significantly the results of other methods.

Apart from the content above, the feature importances obtained after training the supervised learning model as a byproduct will be studied to find out which features in the dataset help the model to determine whether the logistic records are anomalies or not. These features can be used to invert the factors that influence whether or not the goods are delivered accurately during transportation. The reasons that these factors have a greater impact will be guessed and analyzed.

1.3 Novelties of the thesis

The novelties of this thesis lie in the following areas:

- Publicly available data collected from real logistics operations was used. This facilitates the reader to validate and extend basis on this thesis.
- Different supervised learning methods, one-class classification methods, and clustering methods were used. Performances of logistics anomaly detection algorithms were systematically compared and analyzed.
- Because different methods have different anomaly rates for the data used in different datasets, it is not possible to compare these methods directly. The accuracy, precision, and recall values of random classification are obtained by theoretical calculation. Thus it verifies the effectiveness of each method by comparing its performances with the performances of the corresponding random model.
- The factors that are significant to models that detect anomalies in logistics are analyzed, which will help to optimize the logistics system as well as the delivery time prediction model. Predicting the delivery time more accurately provides a better customer experience and avoids losses due to delays.
- An ensemble method is proposed for supervised and clustering anomaly detection. The basic idea of the methods is to use the latent space obtained by encoders as input.

The ideas are tested by experiments and it is found that the method can improve the performance of many models.

- A model that combines latent space and three clustering methods for anomaly detection in the logistics domain is proposed. This will be an interesting research direction for future work.

By examining the performance of different methods on a dataset collected from real logistics operations, it is hoped that this thesis will be able to offer its original perspectives on how anomaly detection in the logistics domain can be carried out in reality, how different aspects of the logistics process can be improved, and how ETA prediction models can be optimized. By providing a little more certainty to the logistics system, losses can be avoided in a world where uncertainty is rampant in all regions, making the supply chain extremely affected. This will reduce logistics costs as well as mitigate the impact on basic life caused by the dramatic inflation that people feel day in and day out.

1.4 Structure of the thesis

The content of this thesis will be organized as follows. In Chapter 2, the thesis will present the results of the previous researchers on anomaly detection in the field of logistics and will mainly focus on the methods they used and the datasets they used. In Chapter 3, the thesis will describe in detail the basic supervised and unsupervised learning methods this project has used and the method this graduation project has proposed. In Chapter 4, the thesis will describe the two datasets this project used, the value distribution of their features, and the reasons for choosing them. In Chapter 5, this thesis will describe how the project processed the dataset before inputting the model, the design of the experimental process, the metrics used to evaluate the results, and how performances for random models are calculated. In Chapter 6, the thesis will present the experimental results in detail and analyze them. In Chapter 7, the thesis will discuss which factors have a greater impact on whether logistics tasks arrive on time. A summary and future work will be provided in Chapter 8.

2 Related Work

During the literature search, I read the papers with 2 things in mind.

- Which methods were used for logistics anomaly detection?
- Which logistics datasets were used for the experiments on anomaly detection algorithm?

Two review papers are great inspiration for this project. One of them is a paper reviewing recent applications of machine learning in transportation by K. Tsolaki et al [17]. While the paper by M. Riveiro et al [18]. focuses on anomaly detection in the maritime domain.

2.1 Types of tasks

There are mainly two types of anomaly detection tasks in the logistics domain.

1. With the basic information of a trip as well as the current time and location shared by the vehicle, the algorithm determines in real time whether the present state of the vehicle is anomalous or not.
2. After the logistics tasks are completed, anomalous tasks are detected by basic information of the trip such as the location of the start and end points, and the starting time of the task.

Papers studies problem type 1 make up the majority. In particular, most of the anomaly detection papers in the maritime and flight domains address type 1.
Type 2 can be represented by the Hofer dataset that Dr. Pham shared with me, as well as the NYC taxi trip dataset [19]. Papers [20] and [21] examine the New York Taxi dataset.

2.2 Related articles

In this subsection, the papers have been read, and the methods and datasets the papers used will be discussed.

2.2.1 A framework for anomaly detection in maritime trajectory behavior

In this paper [22], the authors present a framework for anomaly detection in maritime trajectory behavior, called MT-MAD (Maritime Trajectory Modeling and Anomaly Detection). The highlight of it is that not only does it consider whether each location data point exhibits anomalous characteristics, but also whether the relationships between data points are anomalous. The paper also creatively categorizes anomalies as spatial, sequential, and behavioral.
In the article, the frequent region is defined as the region in which the location data occurs frequently, and the grid-based clustering approach is used to find the frequent region.
Its dataset
The article uses AIS trajectory datasets but doesn't provide the dataset he used. However, inspired by it, some AIS trajectory datasets can be found on the internet and in papers. These datasets will be described in detail later.
This article assumes that the real AIS trajectory data is normal, and adds anomalous data to it. These anomalous data have anomalous behavior with maximum speed and arbitrary direction.

2.2.2 Detecting flight trajectory anomalies and predicting diversions in freight transportation

The area of interest of this paper [23] is anomaly detection in flight trajectory.
The novelty of the article is that it proposes a prediction model such that one-class SVM acts on the metrics Distance completed, Distance gained, Velocity deviation, and Altitude deviation rather than directly on the raw data. By doing so, this model achieves the effect of detecting anomalies from aircraft trajectories only by minimizing the amount of publicly available data. The data used in this paper was obtained from flightstats [24] and flightradar24 [25]. It also does not provide those datasets.

2.2.3 Anomaly Detection in Driving by Cluster Analysis Twice

This article [26] used the HDBSCAN anomaly detection algorithm twice for anomaly detection on driving data. The dataset used in this work is from [27]. The great thing about this article is that it is an experiment on a publicly available dataset with algorithms that can be verified. It successfully detects anomalies in vehicles in the middle of the logistics process without the need for tags and the implementation of a training model. But the downside is that the problem it studies is the first type of anomaly detection problem in logistics that this thesis mentioned earlier, so the content of the dataset does not match my needs.

2.2.4 The Concept of Detecting and Classifying Anomalies in Large Data Sets on a Basis of Information Granules

[28] This paper presents a more robust approach to the use of information granular concepts in anomaly detection problems. This approach makes it possible to use generic algorithms in the field of anomaly detection through a higher level of abstraction. It has chosen data from the logistics domain, this is because data from this domain contains various types of columns (e.g. discrete, continuous, categorical). This thesis uses the classic New York taxi courses dataset.

2.2.5 K-Means-based isolation forest

The authors of this article [21] used the K-means clustering method to complement the traditional isolation forest to obtain a more intuitive search tree. It uses the publicly available New York City Taxi Trip dataset and also the private Intermodal transportation dataset.

2.2.6 Predictive Analytics for Truck Arrival Time Estimation: A Field Study at a European Distribution Center

[29] This article focuses on predicting the time of arrival rather than the time of travel. Arrival time is determined by both departure time and travel time. Departure time is affected by a lot of human factors, but travel time is less affected by human factors. So there is a significant difference between arrival time and travel time. The research methodology of this article is relatively new and relevant. It first used literature reviews and interviews with truck drivers to find out the important factors affecting arrival time and then used real data and data science to verify these factors. The article found that weather and traffic congestion as well as driver incentives and driver proficiency on the route all affect on arrival times. Therefore it is important to include human factors in the anomaly prediction model.

2.2.7 Predicting On-time Delivery in the Trucking Industry

[30] This article from MIT utilizes real datasets from logistics companies to predict on-time delivery using the logistics regression method which is interpretable. It features the use of statistical methods to select features in the data that can be interpreted as having an impact on on-time delivery. By brainstorming with experts, the paper yields important variables that impact on-time delivery and classifies the features that will have an impact on logistics into six categories: load, lane, carrier, process, operation, and faculty. A simple logistics regression model was used to successfully predict on-time arrival, saving 76% of resources. This article is inspiring because it can be realized for the paper that in real logistics anomaly detection applications it may not be necessary to have a very complex model to achieve good performance. In reality, the interpretability of the model is very important. This article also shows that through the real needs of partner companies and not missing anomalies is more important in real-life applications than avoiding identifying normal logistics records as anomalies. In other words, recall is much more important than precision when evaluating anomaly detection models.

2.2.8 Anomaly Detection based on Machine Learning Dimensionality Reduction using PCA and Classification using SVM

[31] In this article, the authors use the dataset that has been downscaled by the PCA method and the original dataset as inputs to the support vector machine, in an attempt to improve the classification method. It found that after dimensionality reduction by support vector machine the model has the following two advantages in classification.

- The speed of classification has increased.
- The probability of misclassification is reduced.

In this thesis, it is desirable to obtain these two benefits by downscaling the data input to the classification model. The thesis is different from this article in that it chose an autoencoder to achieve the effect of dimensionality reduction. The specific reasons for doing so are explained in Chapter 6. Also, in this project, it does not only use SVM as a classification model. Other supervised learning methods and clustering methods are also heavily used as classification models. Their experimental results are compared and analyzed in a theoretical way. Therefore it is safe to say that this thesis is more comprehensive and scientific in its approach compared to this one. But still, it was an inspiration for the graduation project and it led directly to the innovative approach this thesis proposed. So I am very grateful to it and its author.

2.2.9 A Deep Learning Method Combined Sparse Autoencoder with SVM

[32] This is another article that was very inspiring to this thesis. It uses the latent space obtained from the encoder part of the autoencoder as an input to the SVM classification model, improving the classification correctness of the SVM model. However, the article doesn't mention if this method speeds up the SVM model classification. This article inspired the experiment when it was clear that PCA can't reduce dimensionality while retaining information that helps the model determine anomalies. The important difference between this thesis and this article is mainly:

1. While this article focuses on classification problems in the general domain, my focus is on anomaly detection in the logistics domain (which can be considered as a classification problem).

2. This article uses only one classification method, SVM. My thesis will use other supervised learning methods as well as clustering methods to test whether the encoder preprocessing can improve the classification results.

2.3 datasets related to the discovery of logistics anomalies

2.3.1 AIS datasets

Most of the articles in this area use private datasets, more papers in this area are not studies. The article also states that there is no public AIS dataset containing labeled anomalies. However, it is possible to synthesize datasets containing labeled anomalies by the method proposed in section 2.2.1.

2.3.2 Delivery truck trip data

This dataset from kaggle.com is one of the datasets that this experiment is going to use. The details of this dataset can be found in the introduction in Chapter 4. And here the previous informative experiments using this dataset are discussed.

Both [33] and [34] preprocessed this dataset. [33] used XGBoost model on this dataset for anomaly identification on this logistics dataset while [34] used logistics regression and random forest on this dataset for anomaly identification. However, they both explored this dataset using only a handful of supervised learning methods. They have all been very inspiring in the way they pre-processed the datasets. However, there is a lack of comparison and analysis to systematically study the difference in performances of supervised learning methods on this dataset. Both works even less explored the application of unsupervised learning methods on this dataset. Most importantly, they do not use a theoretical approach to analyze how the model performs in the case of random classification as a benchmark to explore the effectiveness of different methods to identify anomalies. In this thesis, theory and a systematic approach will be combined to explore the performance of different supervised and unsupervised learning methods on this dataset.

In addition, [34]'s work includes content on determining the important factors affecting whether or not an arrival is accurate using feature importances obtained through supervised learning methods. This also influences this graduation project greatly. In this thesis, for all the datasets used, more supervised learning methods will be used to more systematically explore the factors that are important for anomaly logistics task detection.

3 Fundamentals

3.1 Anomaly detection

As the name suggests, anomaly detection focuses on the problem of how to identify records that behave differently from other records (normal records). Anomaly can be categorized into three types:

1. Point Anomaly: The behavior of one data record is markedly different from the behavior of other records. For example, one trip took much longer than the others.

2. Collective Anomaly: The behavior of one data record is not significantly different from the behavior of other records. However, it is extremely rare for a collection of multiple records to occur. For example, a credit card is used repeatedly in both Japan and the Netherlands within a short period.

3. Contextual Anomaly: These records do not behave abnormally in themselves, but they are abnormal under certain conditions, such as the arrival of large quantities of Christmas presents at a store's warehouse during the summer.

Anomaly detection has a wide range of applications in many fields, such as identifying credit card skimming, network intrusion, and outbreak detection. Here, it is studied in the context of logistics management.
After reading the literature, it can be realized that when studying logistics management anomaly detection, the problems studied can be divided into two categories.

- The first category is anomalous detection after knowing all the basic facts. It will contain departure times, arrival times, estimated arrival times, transportation distances, and other relevant features. Our goal in studying this problem is to improve supply chain and transportation efficiency by analyzing the records after the fact.

- The second category is anomaly detection during transportation. It usually contains departure time, transportation distance, estimated arrival time, position of the vehicle at a certain time, and other relevant information, but not the actual arrival time. The goal of this type of problem is to detect and respond to logistics anomalies in time to minimize the impact (loss).

The problems studied in this thesis are of the first type. To improve the supply chain as well as the transportation process, there are several methods we can use. In this thesis, the project will use the standard supervised and unsupervised methods as a baseline and propose the idea of combining the two unsupervised methods to investigate the impact of such a combination on the performance of anomaly detection.

3.2 Supervised learning

When using a supervised learning approach, the anomaly detection problem can be viewed as a classification problem that is labeled according to whether the data record is an anomaly or whether the arrival time of the shipment is delayed. A variety of supervised learning methods are used as benchmarks for comparison.

Support Vector Machine

Support vector machine [35] [36] classifies records by finding the best hyperplanes in the partitioned space. Here are some important concepts:

- Hyperplane: A boundary that divides a space into subspaces of different classes. In n-dimensional space, the hyperplane is n - 1 dimensional. For example, in two dimensions, a hyperplane is a straight line, and in three dimensions, a hyperplane is a plane.
- Support vectors: The data records that determine the location of the boundaries, i.e., the data records closest to the boundaries.
- Margin: The sum of the distances between a hyperplane and support vectors from both sides that it separates.

The training goal of support vector machines is maximizing these margins.

In this project, the data points in the multidimensional space will be categorized into two classes: anomalous data and normal data. So next only the case where an SVM is used to separate data records into two types will be discussed. The most basic support vector machine is the hard margin SVM, which assumes that a hyperplane can divide the space into two, with one subspace for one type of data and another subspace for another type of data. However, in real applications, such as our logistics dataset, there are some outliers. These outliers are on the wrong side of the hyperplane. This requires soft margin support vector machines that can allow for errors. The function of the hyperplane is:

$$\mathbf{w} \cdot \mathbf{x} + b = 0$$

Here, $\mathbf{w}$ is the weight vector and b is the bias. ξ is used to represent errors (the degree of misclassification of x_i). The objective function of a soft-boundary support vector machine is:

$$\min_{\mathbf{w},b,\xi} \quad \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\xi_i$$

$$y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad \forall i = 1, \ldots, n.$$

Naive Bayes

This method [37] [38] calculates the probability of the data belonging to each class based on Bayes' theorem. Bayes theorem is as follows:

$$P(C_k|x) = \frac{P(x|C_k)P(C_k)}{P(x)}$$

C_k represents class k and x represents a vector of combinations of values of observed features. $P(C_k|x)$ denotes the probability that x belongs to class k when the combination of the observed features is x. $P(x|C_k)$ represents the probability of occurrence of phenomenon x among all the data records belonging to class k.

Naive Bayes assumes that all features are independent, which is where "naive" comes from. In reality, there are often correlations between individual features, but Naive Bayes can still yield good results in practice. The label of each data corresponds to the highest class in its probabilities. The function that selects the class for the data is:

$$\hat{y} = \arg\max_{C_k} P(C_k) \prod_{i=1}^{n} P(x_i|C_k)$$

Here both $P(C_k)$ and $P(x_i|C_k)$ can be obtained statistically.

Logistic regression

In logistic regression [39] [40], all features of the input set are treated as independent variables $X_1, X_2 \cdots X_k$. The dependent Y is a binary value, which means that this method divides the data record into two classes. Data can be categorized as an anomaly when the probability of the dependent variable being an anomaly exceeds a threshold, which is usually 0.5. The training process tries to find suitable values of $\beta_1, \beta_2 \cdots \beta_k$. The logistics regression can only be used for binary classification, which is suitable for this project. The result obtained by this model is the probability that data record x belongs to a class, and if this probability exceeds a threshold, the data record can be considered as belonging to this class. Usually threshold is set to 0.5, and in this project, it is set to 0.5. Probability's computational function is:

$$P(Y isanomaly) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_k X_k)}}$$

The actual class is denoted as y_i (1 when the record is anomaly and 0 when the record is normal), and the probability that the record is anomaly predicted by the model is $\sigma(z_i)$. The cross function is used as the loss function here. So the objective function is:

$$J(\beta) = -\frac{1}{m} \sum_{i=1}^{m} [y_i \log(\sigma(z_i)) + (1 - y_i) \log(1 - \sigma(z_i))]$$

Decision trees

Decision trees is a supervised learning method [41]. Its structure is similar to a tree structure in data structures Figure 1. It has a root node, many internal nodes, and many leaf nodes. It chooses which child node to flow next in the internal node based on the features of the input data. The choice of features is often based on information theory. Each leaf node represents a class. The class to which the leaf node belongs is the class that has the largest percentage of the data records contained in the leaf node.

Random forest

Random forest [42] is based on decision trees. It is an ensemble learning method. It contains multiple decision trees with parallel relationships. Each decision tree identifies whether a data record belongs to an exception class. Finally, these decision trees vote on the categorization of the data records. Its randomness is characterized by two aspects.

1. In each decision tree of the forest, only a portion of the data records is randomly selected.

2. Among the data records selected for each decision tree in the forest, only a portion of the features is randomly selected.

After getting the labels chosen for each decision tree, a majority vote is used to finalize the class of the data record. The advantage of such randomization is to avoid overfitting. Also, random forest allows the features of the data record to be of type numerical as well as categorical. However, in this experiment, the data is uniformly converted to numerical type.

Gradient Boosted Trees

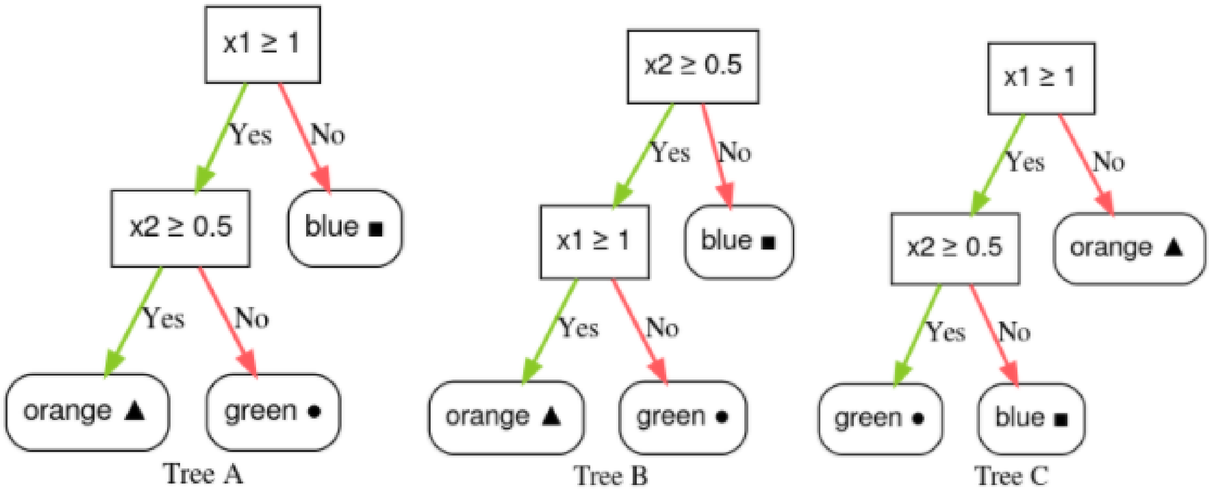

Figure 1: 3 examples of decision trees

Gradient-boosted trees is another supervised learning method that utilizes the idea of an ensemble of trees [43]. It is different from Random forest in which each decision tree is independent. Gradient-Boosted trees are based on two important concepts.

- Boosting: It is the creation of a more accurate learning model using multiple less accurate learning models. The principle of the boosting method is to give higher weights to data records with more significant errors to make the model optimize the prediction of these erroneous data records. Here a sequence of decision trees is used to correct the errors of their predecessors.
- Gradient descent: The goal of Gradient descent is to reduce learning errors. It achieves this goal by finding the direction in which errors decrease the fastest and making the model shift in that direction.

Gradient Boosted trees can be divided into two parts, a strong model F and a weak model f. It can be broken down into the following steps:

1. Initialize a strong model F that treats all data records as normal.
2. Calculate the gap between the predictions of a strong model and the true labeling. The gap is treated as an error.
3. Use a decision tree to generate a weak model f to predict the preceding error.
4. Use the formula $F = F - \eta f$ to update the strong model.
5. Reuse the strong model to predict the class to which each data record belongs.
6. Repeat step 2 through step 5 until convergence or a specific number of iterations is exceeded.

Here learning rate is used to control the speed of model evolution.

XGBoost

XGBoost [44] is an important implementation of gradient-boosted trees. Its advantages over other gradient-boosted trees lie mainly in the highlights.

1. Preventing overfitting: XGBoost uses both L1 and L2 regularization to prevent overfitting.
2. Better trees: XGBoost develops the tree to maximum depth before pruning, which makes it more optimal for each decision tree used.

The second point is referred to as post-pruning, while the general gradient-boosted trees are referred to as pre-pruning, which means that the decision tree stops growing when the information gain is less than a certain threshold or exceeds a certain number of iterations.
Both XGBoost and regular gradient descent trees will be in this experiment.
Note: Since these supervised methods are only benchmark methods, all of the above methods are implemented using functions provided by scikit learn. And only a brief introduction will be made here.

3.3 Clustering

The clustering method [45] is an unsupervised learning method that divides the data records into different clusters, treating each cluster as a class. The knowledge of most of this subsection is obtained from [46]. Each cluster has the following features.

- Similar data records are assigned to the same cluster. Unsimilar data records are assigned to different clusters.
- The difference between data records in the same cluster is much smaller than the difference between data records in different clusters.

In this experiment, there are only two labels for the data records: anomaly, not anomaly. When doing clustering, it also has to be divided into one of the two clusters. So one of the important limitations of the clustering method used is that the number of clusters can be specified as 2, which also means that the number of clusters the clustering process ends up with can be specified as an input parameter. The clustering methods that meet this requirement are K-Means, spectral clustering, hierarchical clustering, HDBSCAN, OPTICS, and BIRCH. The main differences between these methods are shown in Table 4. Because of the different features of clustering methods, they have different use cases Table 3 and K-Means is the general-purpose and most popular clustering method. Methods that work for uneven-cluster-size datasets also work for even-cluster-size datasets. The KMeans method used here employs a mini-batch pattern [47].
Here, non-flat geometry means that the shape of the resulting clusters is non-flat, and the Euclidean distance can not be used as a metric. Different effects of the 2 kinds of clustering methods are shown in Figure 2.
For the trained model, the transductive method can only be used to determine whether a data record similar to the previous one is anomalous or not, that is, it can only detect anomalous behavior similar to the previous one. Whereas the inductive method can detect more novel anomalous behaviors.
We chose these six models because their clustering results can fit one of the following two scenarios.

1. You can specify the number of clusters, such as KMeans, hierarchical clustering, and BIRCH.

name	cluster size	geometry	
K-Means	even	flat	inductive
spectral clustering	even	non-flat	transductive
hierarchical clustering	uneven	flat	transductive
HDBSCAN	uneven	non-flat	transductive
OPTICS	uneven	non-flat	transductive
BIRCH	uneven	flat	inductive

Table 3: Usecases of clustering methods

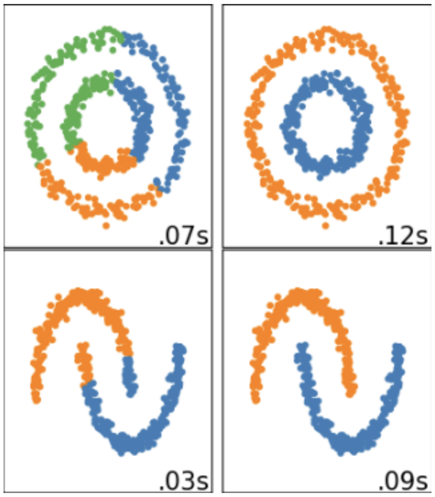

Figure 2: On the left is the result of a clustering method for flat geometry clusters, and on the right is the result of a clustering method for non-flat geometry clusters.

2. It is possible to label outliers as -1, such as HDBSCAN and OPTICS.

Among these six models, spectral clustering is not considered to be of practical value because of its extreme slowness (runtime of more than 10 hours), and it is not used in this experiment. In the clustering experiments, there were close to 37% anomalies in the data. These anomalies also form clusters rather than individual outliers, and recall values are of more interest, i.e., the ability to find all the anomalies rather than the ability to make sure all the anomalies found are actually anomalies. Therefore, in principle, neither OPTICS nor HDBSCAN can correctly identify enough anomalies in the logistic anomaly detection dataset used. nevertheless, these two methods are included in the experiment as a comparison with other methods and as a test of the proposed method against pure clustering methods.
To summarize, three clustering methods are our main focus, which are KMeans, hierarchical clustering, and BIRCH. Specific details about these three clustering methods are learned through ChatGPT and [46].

3.3.1 KMeans

Here k refers to the number of clusters. It can be specified, and for this problem, k will be specified to 2. KMeans can be divided into the following steps.

name	Scalability (number of samples)	Metric
K-Means	MiniBatch mode Very Large	Euclidean distances between points
spectral clustering	Medium	graph distace in nearest neighbors graph
hierarchical clustering	Large	pairwise distances
HDBSCAN	Large	Distances between nearest points
OPTICS	Very Large	distances between pairs
BIRCH	Large	Euclidean distances between points

Table 4: Important features of clustering methods

1. Randomly select k points from the dataset as centroids of k clusters.
2. Assign each data point in the dataset to the cluster closest to it by a centroid.
3. Update the centroids for each cluster. the new value for the new centroid is the average of the data points in the cluster.
4. Repeat steps 2 and 3 until convergence or iteration is exceeded a certain number of times.

If the dataset is too large, mini-batch KMeans can also be used. mini batch KMeans differs in that it randomly selects a sub-dataset for use in steps 2 and 3 before step 2. This method will greatly improve the speed of clustering, and the centroids obtained at the end are very close to the centroids of the normal KMeans. But in this project, ordinary KMeans is enough.

3.3.2 hierarchical clustering

Hierarchical clustering performs clustering by building a hierarchy of clusters, which can be categorized into two types.

- Agglomerative hierarchical clustering: This is a bottom-up approach. It starts by treating each data record as a separate cluster, and in each iteration, it merges the two closest clusters into one. Until the number of clusters matches the specified number.
- Divisive hierarchical clustering: This is a top-down approach. It starts by treating all data records as a cluster, and at each iteration divides a cluster into two. Until the number of clusters is the same as specified. (In my experiment there was only one iteration.)

Agglomerative hierarchical clustering is used in this project. There are 4 ways to calculate the distance between clusters, which are:

1. single linkage: Distance is the minimum value of the distances between the data records of two clusters.

2. complete linkage: Distance is the maximum value of the distances between the data records of two clusters.
3. average linkage: Distance is the average value of the distances between the data records of two clusters.
4. ward's method: Minimizes the variance in the cluster.

In this project, Ward's method is used. This will make the clusters obtained more uniform in size.

3.3.3 BIRCH

BIRCH's incremental feature is especially useful for very large datasets that can't fit in memory. At the same time, it is dynamic and can discover new and novel anomalies as they emerge. BIRCH is based on the CF Tree, and each node in this tree is a representation of a subcluster containing the following information.

- The number of data records in a subcluster.
- Centroid of the subcluster.
- Squared norm of the centroid.
- Linear sum of data records in the subcluster.
- Squared sum of data records in the subcluster.

Two other important pieces of information are the branching factor and threshold. The branching factor limits how many subtrees a non-leaf node can have at most, while the threshold limits the diameter of the cluster represented by a leaf node. This BIRCH method can be divided into the following steps.

1. When a new data record appears, the nodes whose centroids are the closest is recursively selected until a leaf node are met.
2. If the radius of the leaf node exceeds the threshold after adding this data record, the two furthest data records in the leaf node are selected and they are treated as points in two different clusters. Then the data records in the leaf node are assigned to two clusters according to their distance from the previous two points.
3. If the branching factor is violated after splitting a leaf node, recursively split the intermediate nodes upwards.

BIRCH does not perform well on high-dimensional data, so it is anticipated that the proposed method may be able to improve the performance of this algorithm across metrics and simultaneously increase the speed of clustering.
Note: Although these clustering algorithms using Python were implemented from scratch in Python in this project, they might not be reliable enough on large datasets. Since these clustering methods are only benchmark methods, all of the above methods are implemented using functions from sklearn. Here is just a brief introduction to the clustering methods used.

3.4 Autoencoder

The autoencoder is a neural network for unsupervised learning [48]. It consists of three parts: encoder, latent space, and decoder, and the original data passing through the neural network will also include two stages: encoding and decoding. The basic structure is shown in Figure 3:

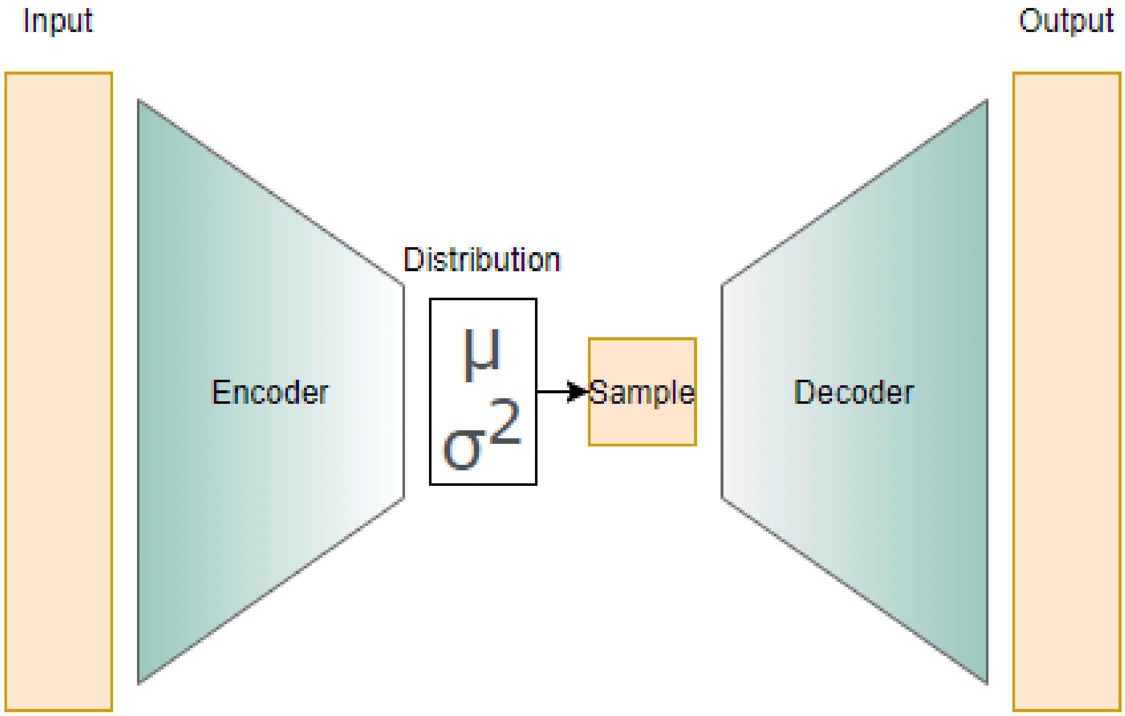

Figure 3: The basic structure of autoencoders

- Encoder: This part of the work is to compress and downsize the original input. Its output is latent space.
- Latent space: It is the output of the encoder as well as the input of the decoder. It compactly represents the original input.
- Decoder: The function of this piece is to decompress and reconstruct the latent space, and we want the reconstructed output to be as similar as possible to the original input.

The training goal of this neural network training is to make the original input of the decoder and the reconstructed output of the encoder as similar as possible.
This neural network is used for three main reasons:

1. The autoencoder itself can be used for unsupervised anomaly detection.
2. Remove noise from the original data.
3. The dimensionality of the original data is reduced and the latent space is used as input for other methods. Because many learning methods do not perform well in high dimensions.

3.5 One-class classification

3.5.1 one-class classification by autoencoders

One-class classification by autoencoders is another unsupervised anomaly detection approach [49]. It is based on the observation that normally anomalous and non-anomalous data have a dissimilar distribution of errors between original input and reconstructed output Figure 6. Here, one class means that our training set contains only normal data records. The autoencoder is trained and normal error distribution with only normal records is obtained.

1. Determine the structure of the autoencoder based on the characteristics of the input data and build the autoencoder based on this structure.

2. A dataset with all normal data records is used to train the autoencoder. The training goal is to reduce the reconstruction error between the input data and the reconstruction output.

3. Obtain the distribution of reconstruction errors for the normal data set. From the distribution, determine the appropriate threshold for distinguishing normal data records from anomalies.

4. In testing and subsequent applications, datasets that contain both normality and anomaly are used as inputs to the autoencoder to determine whether data records are anomalies by the relationship between their reconstruction error and threshold.

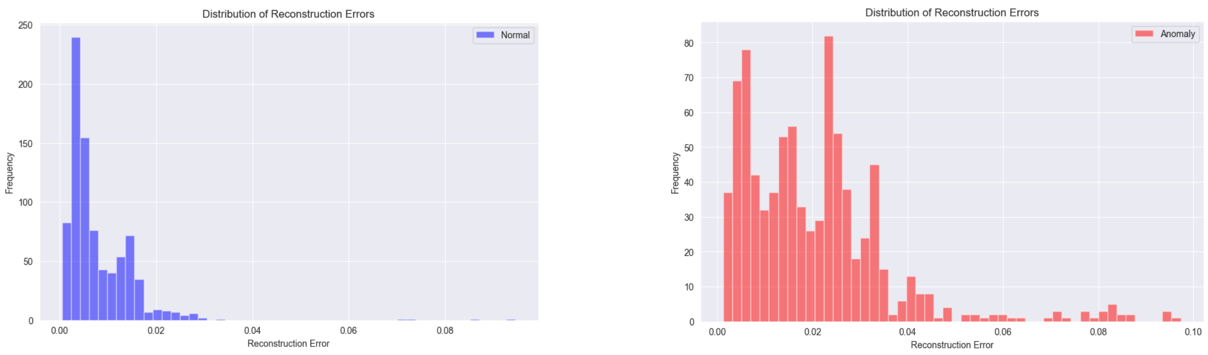

Figure 4: Reconstruction error distributution of normal records

Figure 5: Reconstruction error distributution of anomalous records

Figure 6: Reconstruction errors of autoencoders

In this section, since the autoencoder is one of the key parts of the thesis and it must achieve the right results, its structure was designed by the author with extra care.

3.5.2 one-class classification by SVMs

One-class classification by SVMs is an unsupervised learning method [50]. That is, it's up to the normal data record to determine how to set the hyperplane (the boundary between normal and anomaly data points). In this approach, all data points within the learned boundaries are considered normal and those outside the boundaries are considered abnormal. The one-class SVM can be broken down into the following 4 steps:

1. Get an input dataset that only includes data records from the normal class.
2. Map the data records to a multi-dimensional space with a kernel.
3. Determine suitable hyperplanes as boundaries such that most of the training data points are encompassed in the boundaries.
4. Use datasets containing both normal and anomaly to test and apply the model. We regard data falling outside the boundaries as an anomaly and those within them as normal.

The algorithm is used by importing OneClassSVM from sklearn.svm.

4 Datasets

Finding the right dataset for the problem studied is difficult. Datasets are needed to fulfill the following conditions:

1. The dataset describes transportation-related data.

2. The dataset contains, explicitly or implicitly, information about the distance between the receiver and the sender, the desired transportation time, the actual transportation time, and so on.

3. The dataset should be labeled.

4. The dataset should have high usability.

The discovery of datasets starts by researching the relevant literature. However, it soon became clear that most of the literature did not disclose the datasets they used because of factors such as business confidentiality involved. And it's difficult to find a partner who would provide the project with datasets. So attention turned to kaggle.com again, and two datasets from many that fit the bill are found. They will be introduced in the following two subsections through their important features and the distribution of important features.
Note: Features that were not mentioned in the following two subchapters were also used in the experiments.

4.1 Delivery truck trips data

As the names of the datasets suggest, this dataset is a collection of truck records collected by an Indian logistics company and the important features are shown below Table 5.

Feature name	Feature description
Org_lat_lon	Origin location's latitude and longitude.
Des_lat_lon	Destination location's latituede and longitude.
Planned_ETA	Estimate time of arrival based on plan.
Actual_ETA	actual time of arrival.
ontime	Indicates whether or not the truck reaches the destination before the Planned ETA.
delay	Indicates whether or not the truck reaches the destination after the Planned ETA.
Transportation_Distance_in_km	Transportation distance (km)

Table 5: Important features of Delivery truck trips data

In this dataset, records that did not arrive on time (on time has a value of False) are considered anomalies. In total, more than half of the records are anomalies. This dataset can be considered as a balanced dataset. And distribution of transportation distance is shown in Figure 7.
All columns in the dataset are shown in the table.

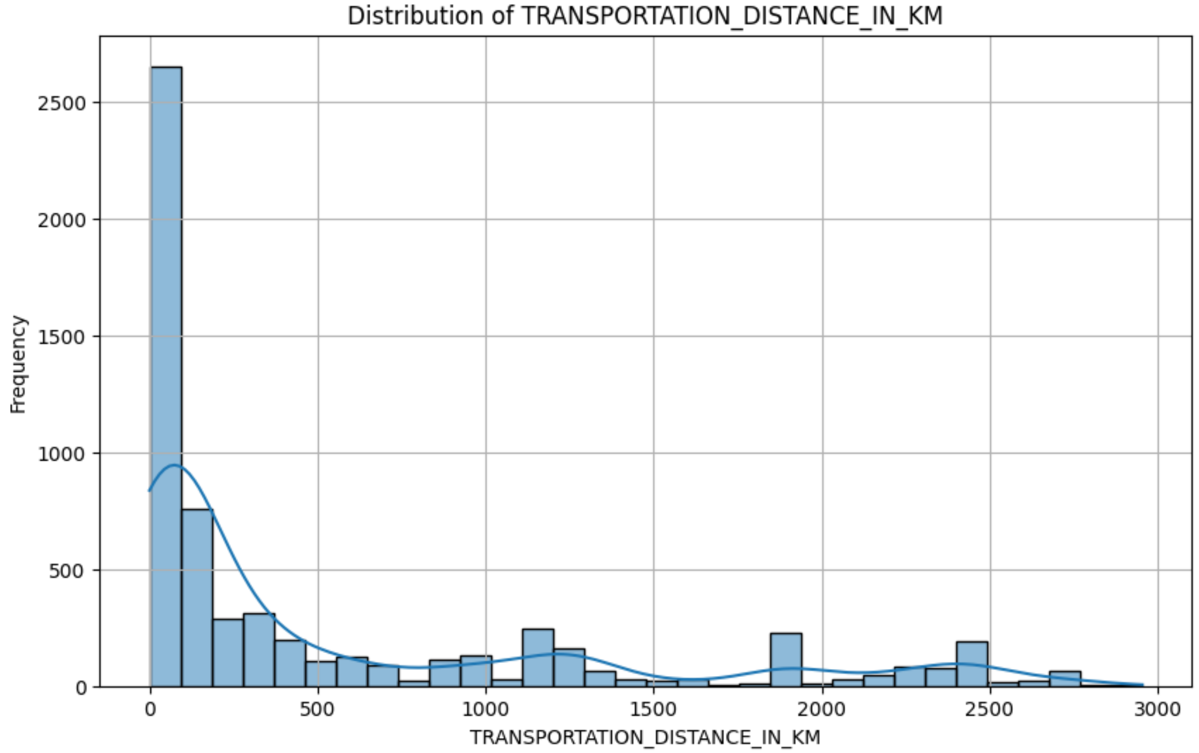

Figure 7: Distribution of TRANSPORTATION_DISTANCE_IN_KM

GpsProvider	BookingID	Market/Regular	BookingID_Date	vehicle_no
Org_lat_lon	Origin_Location	Destination_Location	Des_lat_lon	Curr_lon
Planned_ETA	Current_Location	DestinationLocation_Code	actual_eta	Curr_lat
Data_Ping_time	Driver_MobileNo	DestinationLocation	delay	ontime
trip_start_date	trip_end_date	transportation_distance	vehicleType	
customerID	customerNameCode	OriginLocation_Code	supplierID	
supplierNameCode	Material Shipped	Minimum_kms_in_a_day	Driver_Name	

Table 6: Column names of Delivery truck trips data

4.2 Brazilian E-Commerce Public Dataset by Olist

This is a dataset about orders shared by Olist, a Brazilian e-commerce platform, at Kaggle. This dataset contains multiple files, and the relationship of the records in the files can be obtained by their IDs. Therefore it can be considered as a relational data model, where each file is equivalent to a data table and the whole dataset is equivalent to a database. The relationship between the data is shown below in Figure 8.

As the figure shows, this dataset contains information on multiple aspects of an order such as product, payment, and ratings, and can be used for all-encompassing research on a variety of e-commerce-related topics. However, only files related to logistics will be studied. The core file related to logistics is olist_orders_dataset, and other files provide important supporting information.

The important features are shown below Table 7.

As Figure 7 shows, the distances exhibit a long-tailed distribution. For most of the senders and receivers, the distances are in the range of 200 kilometers, with 100 kilometers being the

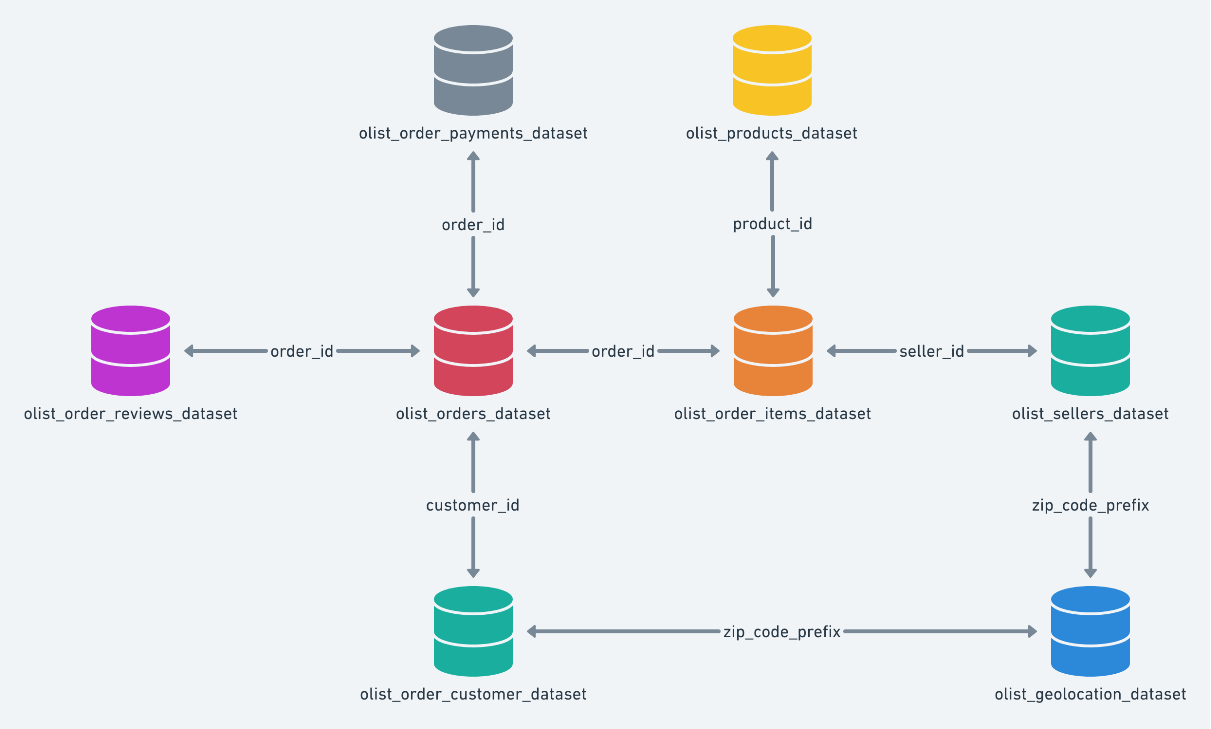

Figure 8: Brazilian E-Commerce Public Dataset

Feature name	Feature description
geolocation_lat	Seller or customer location's latitude.
geolocation_lng	Seller or customer location's longitude.
order_delivered_carrier_date	The date on which the goods were transferred from the sellers to the carriers.
order_delivered_carrier_date	The date on which the goods were transferred from the sellers to the carriers.
order_delivered_customer_date	The actual date of goods arrival.
order_estimated_delivery_date	The estimated date of goods arrival.

Table 7: Important features of Brazilian E-Commerce Public Dataset

majority. So it's a safe bet that the model that predicts delivery times may be inaccurate in the rare case of long distances, leading to anomalies. Therefore, the distance factor between the sender and receiver is expected to have a significant impact on whether the shipment is delivered on time, and the results of the experiments will confirm this expectation.

Records, where the actual arrival time exceeds the estimated arrival time, are considered as anomalies. In this dataset, only 7% of the records are anomalous. So this is an unbalanced dataset. Unbalanced datasets can lead to many models that find anomalies not treating anomalous and normal records fairly. So the way to create balanced datasets from unbalanced datasets will be described and balanced datasets will be used for the training data of many of the algorithms.

5 Methods

5.1 Experiment Design

The flow chart of the experimental design is shown below in Figure 9.
The experiments are designed to accomplish two main goals.

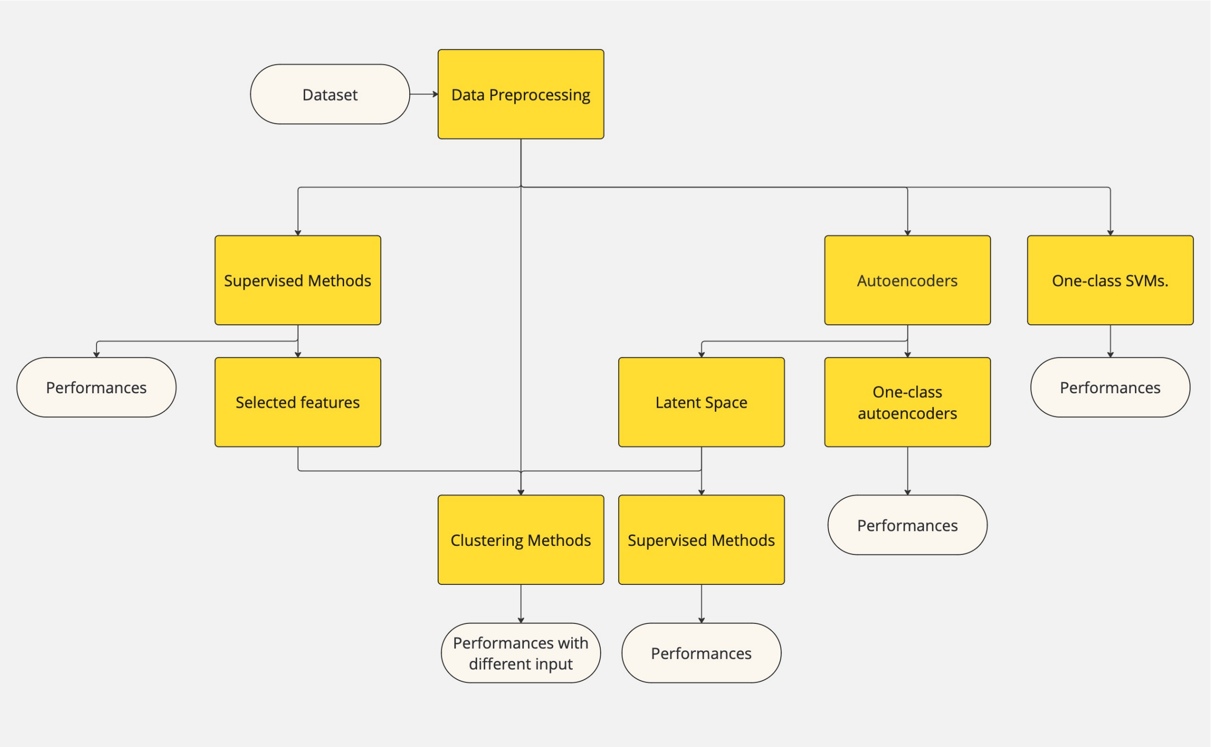

Figure 9: Experiment Design

- Comparing the performances of supervised learning methods, clustering methods, one-class autoencoders, and one-class SVMs for logistics anomaly detection and trying to understand the differences between them.
- Comparing the performance of supervised and unsupervised learning methods using latent space with other methods and testing whether they can be better.

The process of the experiment can be explained in detail as follows.

1. Preprocessing both of the datasets.
2. Experimenting and getting the performances of these models using multiple supervised learning methods and one-class SVM provided by scikit learn.
3. By analyzing the feature importances from the supervised learning results, manually selecting features of higher importance and selecting the columns that contain these features to be merged into a dataframe that will be used as input in the future.

4. Building and training autoencoders with raw input data as described in 5.2. After training, the raw input data will be passed through the encoder to get the latent space. This latent space will also be used by the unsupervised and supervised learning methods.

5. Using the same autoencoder architecture as in the previous step, train the autoencoder again using only normal data and use the one-class autoencoder method for anomaly detection. And get the performances of this method based on this result.

6. The latent space obtained in the previous step is used as an input to the supervised methods, and once again the performances of the supervised learning methods are obtained.

7. The original input data, the latent space, and the manually selected data obtained in Step 2 are all used as inputs to the clustering methods, and then the performances corresponding to the various methods are obtained separately.

8. Compare and analyze performance data.

Here, the performances obtained in step 7 are of great direct comparative significance. It's estimated before the experiment that the order of performances should be:

$$original\_input\_data < latent\_space < manually\_select$$

This means that the result after autoencoder preprocessing should be somewhere between no preprocessing and manual preprocessing by a human. The performance of supervised methods that have been preprocessed by the autoencoder is also compared with those that have not. And one-class autoencoder's performance shows how well the autoencoder understands the data. Thus explaining the difference in performance before and after preprocessing.
Novel findings from the comparison of all the methods are also expected.

5.1.1 Comparison to existing approaches

The focus of this experiment differs from most of the previously mentioned methods used by previous authors in the following ways.

1. Most of the articles are divided into those that present a specific method [51] [52] and summarize it [2] [7], and those that summarize those previously mentioned but do not perform experiments. However, this thesis will directly apply the different methods to the same datasets. This allows a direct comparison of the performance of these methods.

2. Comparing the methods here more than just in parallel like [45] does, and experimenting more through the hierarchical structure shown in the figure. In this way, the performances of combining different methods for anomaly detection can be explored, and this combination of simple methods is closer to the real applications in the industry.

3. The datasets used in most articles are either unlabeled or not publicly available. Only labeled public datasets are used here. It is easy to check the performances of the methods and later on the comparative tests of others.

4. All the relevant articles have only focused on the impact of different methods on the accuracy of anomaly detection. But no mention of which factors (distance, shipping time...) would have a greater impact on the occurrence of anomalies in transportation. This aspect of anomaly detection is explored here.

5.2 Data preprocessing

The preprocessing methods on the two logistics datasets used in this final project do not differ much from the other datasets [53]. For both datasets, the preprocessing they underwent consisted of:

- Changing the data type from object to int, float, bool, and datetime, etc.
- Getting year, month, day from datetime.
- Removing invalid data records.
- Removing invalid features.
- Removing duplicate features.
- Removing irrelevant features.
- Getting trip distance with trip start and destination coordinates.
- Transforming categorical features from type object to float with OrdinalEncoder [54].

A flowchart showing the data preprocessing process is shown in Figure 10.

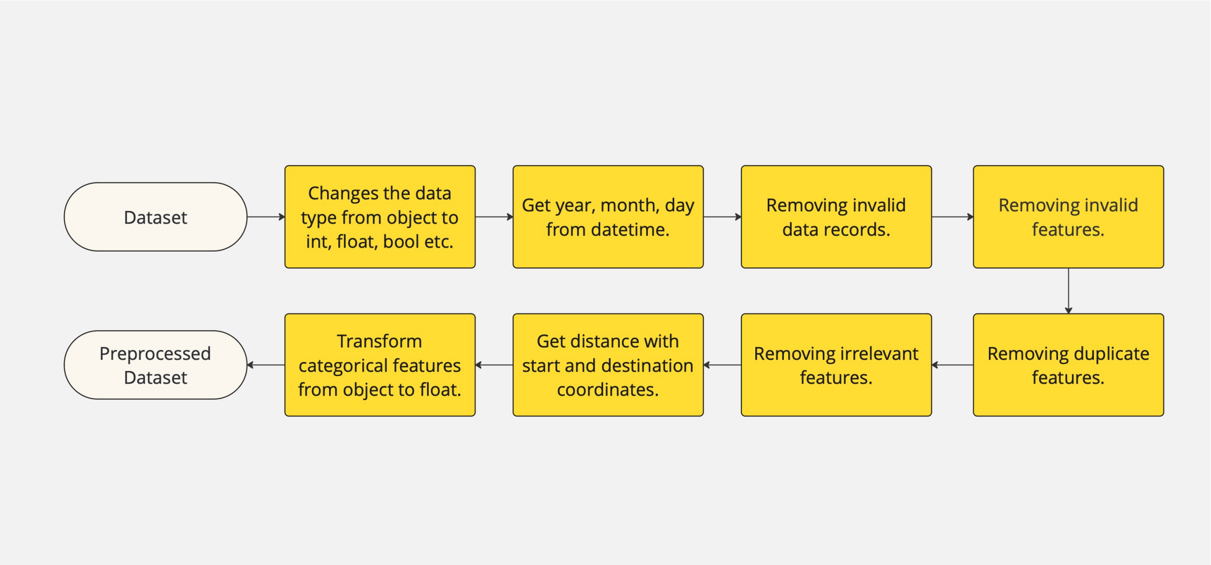
Dataset
Changes the data type from object to int, float, bool etc.
Get year, month, day from datetime.
Removing invalid data records.
Removing invalid features.
Removing duplicate features.
Removing irrelevant features.
Get distance with start and destination coordinates.
Transform categorical features from object to float.
Preprocessed Dataset

Figure 10: Flowchart of preprocessing

For the Brazilian E-Commerce Public Dataset, due to the relational database-like structure shown in Figure 8, getting a dataframe containing all relevant features by merging them by id is needed. The operation of connecting to a file in the database is similar to the Join operation in the SQL language. The actual code uses the merge operation from pandas [55].

The proportion of anomalies and the distribution of values for each feature are essentially unchanged for the results after data preprocessing. However, the preprocessed data removes duplicates, errors, and missing data. This keeps bad data records from affecting model training and testing. There is also a lot of data that is categorized, and the data type is "object". However, the latter model cannot handle this type of data. So ordinalEncoder is used to turn them into integers. But there is a downside to this. For example, in clustering methods, for two data records, we may label their vehicleType as 1 and 2 respectively, or label them as 1 and 10 respectively, in both cases, their distance in vehicleType dimension is different, which will affect the final clustering result.

After preprocessing the data there will still be a correlation between the data features. For example, the latitude, longitude, and distance of both the sender and receiver will be kept, even though the distance can be calculated from the latitude and longitude. Also, the zip code has a high correlation with the time latitude and longitude. This correlated information will introduce more dimensionality as well as the problem of the curse of dimensionality. Subsequent experiments will show that it is necessary to keep the data that is correlated. the autoencoder method will also be used to reduce the correlation and mitigate the negative effects of the curse of dimensionality.

It is also important to note that this is only the first stage of preprocessing that the raw data undergoes, and it is the necessary preprocessing that the input data for all methods undergoes. Later on, the data will be preprocessed using an encoder and manually selecting features for certain models. These methods are attempts to optimize existing methods and will not be used on all input data for all methods.

5.2.1 Data preprocessing for Delivery truck trips data

In this dataset, there are a large number of duplicate columns, such as delay and ontime. In the table 8 [56], their correspondence and the relationship of the exception labels will be shown. In addition, the actual ETA and planned ETA also act as labels to indicate whether the data is an exception or not. So all the duplicate label columns are dropped and only one labeled column is left.

ontime	delay	anomaly
Yes	No	No
No	Yes	Yes

Table 8: Compare label columns

Destination_Location and DestinationLocation are also duplicate lines. But both of them are dropped because DestinationLocation_Code can record the same information more accurately. Origin_Location is also more accurately represented by OriginLocation_Code, so it also dropped. Also, customerNameCode and supplierNameCode can be represented by the IDs of the supplier and customer, and they are removed as well. The booking-related information for the order is also dropped because logistics operation is the focus and the trip start time is already available. The Datetime type trip_start_date is also broken down into day, month, and year, and broken down lat_lon into separate latitude and longitude.

Drop columns about driver information, because about 50% of the records don't have driver information. Even for those who have, there can be drivers with the same name. But vehicleType, OriginLocation_Code, DestinationLocation_Code value NaN are also replaced with a categorical value. This is because the vast majority of entries in these columns contain useful information. All information conveyed by GPS is removed, which contains 'GpsProvider', 'Data_Ping_time', and current locations. This is because the logistics anomaly detection problem studied is not concerned with the state of the truck in the midst amid its travels.

After all this processing the type of the classified features is changed from object to float to make it easier for the following methods to process them. The preprocessed columns are shown in Table 9.

OriginLocation_Code	market/regular	Org_lat	customerID	ontime
DestinationLocation_Code	vehicle_no	Org_lon	supplierID	day
transportation_distance_in_km	vehicleType	Des_lat	trip_time	month
minimum_kms_to_be_covered_in_a_day	material shipped	Des_lon	dayofweek	year

Table 9: Features of dataframe after preprocessing for Delivery truck trips data

5.2.2 Data preprocessing for Brazilian E-commerce Public Dataset by Olist

The most important thing to preprocess this database is of course to join them by ID, like the join operations in SQL for relational database. Here, the data tables needed to connect to are olist_orders_dataset, olist_order_customer_dataset, olist_order_items_dataset, olist_products_dataset, olist_sellers_dataset, and olist_geolocation_dataset. After joining them, all the ids connecting them can be dropped.

Another important fact is that the dataset itself does not contain labeled columns that directly indicate whether a data record is an exception or not. So it's needed to determine if a record is an anomaly by comparing order_estimated_delivery_date and order_delivered_customer_date. When the former is less than the latter, the data is an anomaly. After getting the labeled columns, we can also drop these two columns.

The order of task processing steps is: purchase, approve, send from supplier to carrier, and send from carrier to customer. So a rule is enforced: $order\_purchase\_timestamp < order\_approved\_at < order\_delivered\_carrier\_date < order\_delivered\_customer\_date$. Variables that do not meet this condition are broken data records. In addition, transportation is the focus, that is, the time from the carrier collection to the delivery of goods to the customer. The expected time spent during this period is kept as one of the features. Like above, the time the shipment was collected by the carrier is also broken down into days, months, and years.

Based on the values of latitude and longitude, the straight line distance between the supplier and the customer is obtained using the Haversine formula below, which is used to approximate the distance of transportation.

$$d = 2r \arcsin\left(\sqrt{\sin^2\left(\frac{\Delta\phi}{2}\right) + \cos(\phi_1)\cos(\phi_2)\sin^2\left(\frac{\Delta\lambda}{2}\right)}\right)$$

It's also processed by OrdinalEncoder after the above procedures. The columns of the input dataset to the models are shown in Table 10.

5.3 A balanced dataset based on Brazilian E-Commerce Dataset by Olist.

As mentioned at the end of Section 4.2, the Brazilian E-commerce Dataset contains only about seven percent of the anomalous data. At the beginning of the experiments, an unbalanced dataset was used, and the predictions were severely affected by the high imbalance of the dataset. In general, using an unbalanced dataset for classification (anomaly detection problem studied in this thesis is essentially a classification problem) faces the following problems:

anomaly	seller_geolocation_lat	product_category_name	freight_value
year	seller_geolocation_lng	product_weight_g	seller_zip_code_prefix
month	customer_geolocation_lat	product_length_cm	customer_zip_code_prefix
day	customer_geolocation_lng	product_height_cm	distance
price	time_estimate_delivery	product_width_cm	

Table 10: Features of dataframe after preprocessing for Brazilian E-commerce Public Dataset

- **Bias:** This is because anomaly data is underrepresented in an unbalanced dataset, and thus may result in being ignored during training. Creating a balanced dataset helps to eliminate bias.
- **Overfitting:** This unbalanced dataset may introduce overfitting. So it performs well in the case of the training set; whereas the proportion of anomalous data in new situations (e.g., epidemics, wars) can be significantly higher, and the model is not able to adapt to this situation.
- **Clustering:** In clustering algorithms, since the percentage of abnormal data is extremely low, they may be ignored by the clustering algorithm, which goes on to categorize the normal data.
- **Visualization:** Even if the data is successfully classified into anomalous and non-anomalous classes, at the end of the clustering, when the clustering results are analyzed, it is difficult to analyze the causes of the anomalies through visualization tools if the percentage of anomalies is extremely low.

If this problem is described figuratively, it is as if when a batch of products has a very low failure rate, the quality inspector may carry the bias that this product should all pass, so much so that when he encounters a product that fails, he also has a higher probability of believing that this product is okay because of his bias. Since the data records in our problem are the products and our anomaly detection model is the quality inspector, the purpose of creating a balanced dataset is to remove the bias.
In this dataset, the specific method for creating a balanced dataset is:

1. Extract all anomalous data from the unbalanced dataset.
2. Extract the same amount of normal data as the anomaly data extracted in the previous step from the set of normal data records.
3. The normal and anomaly data records extracted above are merged and shuffled.

But creating balanced datasets is not exactly the same for supervised and unsupervised learning methods.

5.3.1 For unsupervised methods

For unsupervised learning, the experiments are simpler, and balanced datasets can be created directly from the original dataset. Clustering is performed on the balanced dataset.

5.3.2 For supervised methods

The unbalanced dataset is divided into two parts, training and testing, and the balanced dataset is used in the training part to guarantee that both abnormal and normal data are represented to the same extent. However, in the testing part, it's important to simulate the real-world environment and hence the unbalanced dataset should be used.
So, first, the original dataset is divided into training and testing datasets. An additional step of balancing preprocessing is performed for the training dataset.

5.4 Exploring the dataset by PCA

The full name of PCA is principal component analysis. It's used here to reduce dimensionality so the data can be visualized and explored. The goal of this method in the process of dimensionality reduction is to keep the data as different from each other as possible. If after visualizing it and using different colors to mark normal and abnormal data records, there are clusters with mainly abnormal data points and some clusters with mainly normal data points then. Then it will be sure that the clustering method can give better results directly. But, spoiler alert, the PCA didn't have that effect.
The process of this algorithm can be summarized as follows:

1. Standardize the dataset so that the range of each feature is the same. This ensures that the algorithm can treat the importance of each feature without bias.

2. Calculate the principle components such that: 1. the components are orthogonal to each other. 2. the first component maintains the maximum variance between data records, and the second component maintains the second largest variance between data records, provided that condition 1 is satisfied.

3. The projections of each data record onto the individual principle components comprise the transformed data record.

This method can also be used for preprocessing data, and it may have these several advantages:

- Accelerating subsequent classification algorithms.

- Reduce the extent of overfitting.

- Reduce the correlation between features.

In Article [31], a combination of PCA and SVM classification methods is used to analyze that PCA not only improves the speed of SVM model classification but also improves the accuracy. Article [57] also proposes the use of PCA as a means of dimensionality reduction to improve the performance of other anomaly detection methods. However, during data exploration, it was found that PCA was not effective in achieving successful dimensionality reduction and preserving anomalous signals for logistics datasets. In this thesis, PCA is only used to explore the data.

5.5 A proposed Method

Here, an unsupervised learning method is designed to try to solve some of the problems of traditional clustering methods by combining the advantages of autoencoders and clustering methods. This method simply takes the latent space obtained from the original data records by theencoder of autoencoders as the input of the clustering method.

5.5.1 Inspirations

The initial inspiration came from visualizing and then performing clustering on the results when exploring the dataset after the preprocessing of it is finished. Since each record in the dataset is nearly 20 dimensions, and the human eye can only comprehend content in two or three dimensions, dataset downscaling is needed. However, the author found that the results coming from PCA are less than ideal and can't be clustered in a way that separates normal data points from anomalous ones. The results of PCA can be seen in Section 6. So the author dug into the literature and attempted to find a better way to do dimensional reduction. That's how the author remembered that the encoder part of autoencoders can be used for dimensionality reduction. There are multiple articles about using autoencoders as the preceding step of autoencoders. And all of them are focusing on temporal because of the complexity of this kind of input data. [58] focuses on temporal data gathering from IOT devices and [59] focuses on videos which is also temporal. The author want to explore the possibility of using it on non-temporal data such as our datasets and in the field of logistics.

5.5.2 Introduction to the purposed method

It can be broken down in the following steps:

1. Building a suitable autoencoder architecture based on the characteristics of the input data.
2. The raw data records are used to train the autoencoder with the same training goal of reducing reconstruction error.
3. Refinement of the structure of the autoencoder (number of layers, number of dimensionality reductions per layer). The goal is to reach the optimal autoencoder structure and parameter values.
4. Raw data that would otherwise be classified directly by clustering methods is first passed through the encoder portion of autoencoders. Then we get the latent space of the original data.
5. The latent space of the raw data is classified using clustering methods to determine whether they are anomalies or not.

The procedures of this method is shown in the flowchart (Figure 11).
This has the following benefits over the traditional approach of using only clustering methods.

- **Dimension Reduction:** This is because our datasets are all very high dimensional (19 or 20 dimensions). So it is very important to reduce the dimensionality of the input data. Doing so can effectively alleviate the curse of dimensionality problem.

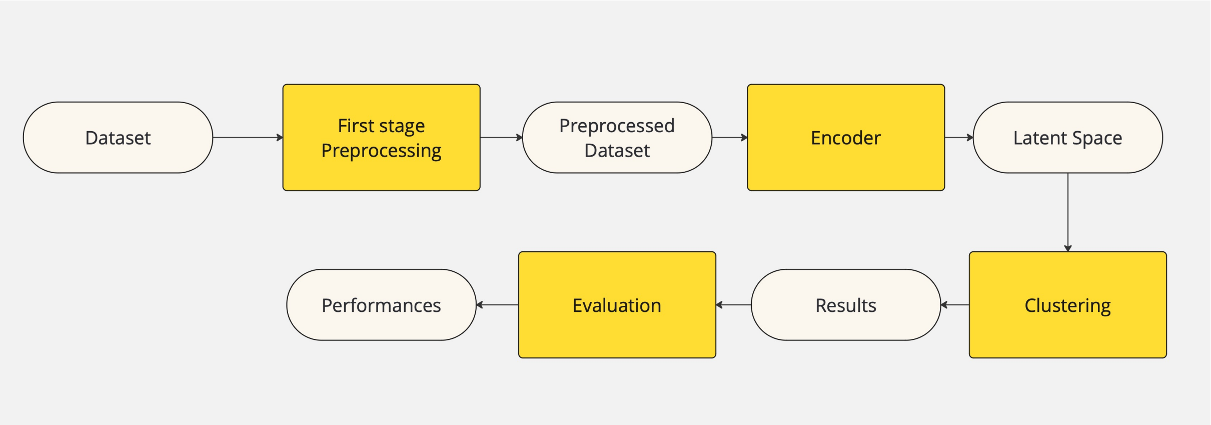

Figure 11: Procedures of the purposed method

- **Acceleration:** Because the datasets, as well as datasets common in the logistics domain, are very large, training and using them is time-consuming. In real applications, sometimes the system has the requirement to find anomalies within a certain time limit. By reducing the dimensionality of the input data, we can greatly accelerate the training and application.

- **Generalization:** Through dimensionality reduction, an encoder can filter out potentially misleading information, leaving only the more important features of the data. This makes its clustering more generally applicable and more robust when encountering new data in the future.

- **Noise reduction:** The data contained in the latent space is more important for identifying whether it is an anomaly or not because it removes a lot of irrelevant information through the encoder. This extraneous information is the noise. This makes the clustering methods not to be disturbed by this information.

- **Higher accuracy:** Generally speaking clustering methods and many other classification methods do not perform well in high dimensional states, and lowering the dimensionality will help them to get higher accuracies.

- **Standard interface:** From a software engineering point of view, if the dimensions of the data records in the obtained latent space are the same, then the interfaces of the supervised and unsupervised learning methods used in the next step do not need to be changed. Here it is still changed to achieve better training results, but a standardized interface can be designed for practical applications.

- **Visualization:** If the data can be meaningfully reduced to 2 or 2 dimensions, it will be possible to see the results of the clustering directly through visualization tools. This leads to a better understanding of clustering methods. However, the logistic dataset is still too complex to be reduced to 2 or 3 dimensions and still retain enough information.

- **Transfer knowledge:** The encoder obtained contains the knowledge to extract important information from a transportation dataset. By utilizing this encoder on other similar data, the knowledge on top of this dataset is utilized to help extract important information on other data. But in this problem, the auxiliary features in both the datasets are more different so we are not able to validate this aspect.

One of the main tasks of this graduation project of mine is to verify whether this method obtains these benefits. To this end, the experiments designed to verify these will be presented in the next subsection.

5.6 Performance Evaluation

To compare the benchmark method with the proposed method, it is necessary to find the appropriate metrics. and for supervised learning methods and clustering methods, there are different ways of determining data anomalies. In this subsection, the metrics used to compare these methods will be discussed.

5.6.1 Performance Evaluation for all methods

Assumption: For unsupervised methods, enough labels also exist to tell how to correctly label each cluster.
Before introducing the confusion matrix, the following definitions will be introduced:

- True Positive (TP): The number of anomaly elements that are correctly identified as anomalies by the clustering methods.
- True Negative (TN): The number of normal elements that are correctly identified as normalities by the clustering methods.
- False Positive (FP): The number of normal elements that are incorrectly identified as anomalies by the clustering.
- False Negative (FN): The number of anomaly elements that are incorrectly identified as normalities by the clustering.
- N: The total number of elements.

Now with these previous concepts in place, introducing several evaluation metrics for supervised learning methods is possible.
Confusion matrix: In a confusion matrix, how many elements are in each of the four types above can be observed to get a rough idea of the quality of the clustering method. The structure of the matrix is shown in Table 5.6.1.
Accuracy: The percentage of elements that are correctly classified:

$$Accuracy = \frac{TP + TN}{N}$$

Precision: The percentage of all elements classified as an anomaly by the clustering method that is correctly categorized.

$$Precision : \frac{TP}{TP + FP}$$

Recall: The percentage of all anomalies that are correctly categorized as anomalies by the clustering method.

$$Recall : \frac{TP}{TP + FN}$$

		Predicted	
		Positive	Negative
Actual	Positive	True Positives (TP)	False Negatives (FN)
	Negative	False Positives (FP)	True Negatives (TN)

Table 11: The structure of confusion matrix

5.6.2 Performance Evaluation for Clustering Methods

One of the problems encountered when evaluating unsupervised classification methods such as clustering methods is that because there are no labels, it is difficult to know which label in the ground truth each cluster should correspond to after obtaining the classification using clustering methods. For example, for the two categories in Table 12, the scores obtained for Precision and Recall are 0. But they are two identical classification results.

	Class 1	Class 2
Ground truth	b, c	a
Clustering	a	b, c

Table 12: An example.

The previous method used borrowed the label information to determine which cluster is normal and which is not. However, it's possible and important to test the performance of the clustering method using an evaluation method specifically designed for clustering classification.
When evaluating clustering classification methods, it's also needed to be careful that the value of the label should not be a factor that affects the evaluation of the method. The following two aspects should be focused on:

1. Whether pairs of similar (or of the same class under the ground truth label) elements in the dataset are assigned to the same cluster.
2. Whether pairs of elements in the dataset that are dissimilar (or different classes under the ground truth label) are assigned to different clusters.

To quantitatively evaluate the clustering method in these two aspects, the following metrics are introduced.

- C: The total number of pairs of elements in the Dataset.
- a: The number of pairs of elements belonging to the same set under both the ground truth label and the clustering method label.
- b: The number of pairs of elements belonging to different sets under both the ground truth label and the clustering method label.

Rand Index
The Rand Index (RI) indicates the similarity between the classification based on ground truth labels and the classification based on labels obtained by clustering. The value of RI ranges from 0 to 1. The value of RI ranges from 0 to 1, where 1 means that the two classifications

are the same and 0 means that they are completely different. The mathematical formula for the random index is as follows:

$$RI = \frac{a+b}{C}$$

Mutual Information based scores

Mutual Information (MI) based scores are another metric to calculate the similarity between the classification based on ground truth labels and the classification based on labels obtained by clustering. For this type of metric, a bad clustering may obtain scores with negative values. The mathematical formula for the scores is as follows:

$$P(i) = \frac{|C_{1i}|}{N}$$

$$P'(j) = \frac{|C_{2j}|}{N}$$

$$MI(C_1, C_2) = \sum_{i=1}^{|C_1|} \sum_{j=1}^{|C_2|} P(i,j) log(\frac{P(i,j)}{P(i)P'(j)})$$

P(i) is the probability that an element is categorized as class i in the distribution C_1 and P'(j) is the probability that an element is categorized as class j in the distribution C_2.

Note: in this experiment, Adjusted Mutual information (AMI) is used. The main difference between it and MI is that it is normalized against chance. This means that two identical clusters get an AMI score of 1. This is not the case for MI, which makes it difficult to judge a clustering method.

Homogeneity and completeness

Looking at the problem of evaluating clustering methods from another perspective, it can be divided into two aspects.

1. Whether all elements belonging to the same class according to ground truth labels are put into the same cluster.
2. Whether all elements put into the same cluster belong to the same class according to the ground truth labels.

Metric 1 is called completeness and metric 2 is called homogeneity. Their values range between 0 and 1. For two identical classifications, both indicators should have a value of 1.

V-measure

V-measure is the index for the above two metrics are combined to evaluate the clustering method, its mathematical formula is as follows:

$$v = \frac{(1+\beta) \times homogeneity \times completeness}{(\beta \times homogeneity + completeness)}$$

Here, when β is 1, the weights of completeness and homogeneity are the same, which is the value used. Completeness has a higher weight when β is smaller, and homogeneity has a higher weight when it is larger.

For all of the above metrics to be computed specifically, the metrics package sklearn.metrics provided by scikit learn will be used.

5.6.3 Comparison between traditional evaluation metrics and metrics for clustering methods

Overall, the difference between evaluation metrics that apply only to clustering and generalized evaluation metrics is in two ways.

1. Evaluation metrics based on clustering can only evaluate the quality of clustering. They don't require labels.
2. Generic evaluation metrics require labeling. They can evaluate the accuracy and recall of clustering.
3. Clustering quality can only indirectly evaluate the fitness of the algorithm. Our ultimate concern is accuracy and recall.

In the experiments, since unsupervised learning methods are assumed to exist without labels, and to compute traditional evaluation metrics such as accuracy, precision, and recall, labels are needed, a better way is to look for metrics commonly used for evaluating the quality of clustering to help us evaluate and compare the effectiveness of clustering methods for anomaly detection. The methods found to accomplish this are rand index [60], mutual information based scores [61], and v-measure [62]. These methods tend to evaluate the quality of clustering rather than the accuracy of classification (because of the assumption that labels do not exist we cannot evaluate the accuracy of classification either).
However, determining the accuracy, precision, and recall of the clustering is still needed so that the clustering method can be directly compared with other methods. These three indicators are also the ultimate goal. To get these three indicators, the clusters have to be labeled as "normal" or "abnormal". Here, it's possible to just borrow labels to count the ratio of abnormal data records to normal data records in each cluster, and clusters with a high ratio of abnormal data records are labeled as "anomalous". In subsequent experiments, it will be found that of the two datasets obtained by most clustering methods, there is usually a much higher proportion of anomalous data records in one than in the other. This indicates that the cluster method does accurately identify anomalies. This may also indicate that it makes sense to borrow labels to evaluate whether the model is successful in identifying anomalies.

5.7 Random classification

In this chapter, the global baseline, i.e., what the values of accuracy, recall, and precision look like over two datasets after a randomized classification., in a theoretical way will be explored. First, let's start with the Law of Large Numbers [63].
Law of Large Numbers: The average of the results obtained from a large number of trials will converge to the expected value as the number of trials increases.
Looking at the two datasets used, the number of records they have is shown in the table.

Table 13: Number of records that the datasets have

Delivery truck trip data	Brazilian e-commerce public dataset
6750	10228

It can seen that both datasets are large enough so that the law of big numbers applies here 14. Because the proportion of anomalies in the two datasets is different, they are discussed separately in the next section. It is important to state that only training and test data are used when using supervised methods. The training set was balanced because the public dataset of e-commerce in Brazil is highly unbalanced; the training set for Delivery truck trip data was not balanced.

Table 14: Proportion of anomalies in the two datasets

	Delivery truck trip data	Brazilian E-commerce public dataset
Entire dataset	37%	7%
train set	37%	50%
test set	37%	7%

For each method and dataset, the x and y values are shown in the table. 15

Table 15: Example Table

methods & dataset	x	y
supervied & delivery truck trip data	37	37
supervied & brazlian e-commerce dataset	7	50
clustering & delivery truck trip data	37	50
clustering & balanced brazlian e-commerce dataset	7	50
clustering & unbalanced brazlian e-commerce dataset	50	50

5.7.1 How to calculate the metrics

It's the assumption here that there are a total of N data records and y% of data records in the training dataset are anomalous. Then the values of TP, TN, FP, and FN are calculated as shown below.

1. Assume there are $x\%$ anomalies in the test data, there are a total of $N \times x\%$ anomalies recorded.
2. There are $N \times (1 - x\%)$ of normal data records.
3. TP: $N \times x\% \times y\%$.
4. TN: $N \times (1 - x\%) \times (1 - y\%)$.
5. FP: $N \times (1 - x\%) \times y\%$.
6. FN: $N \times x\% \times (1 - y\%)$.

The values of accuracy, precision, and recall are calculated as follows.

- accuracy: $\frac{TN+TP}{N} = x\% \times y\% + (1 - x\%) \times (1 - y\%)$
- recall: $\frac{TP}{TP+FN} = \frac{y\%}{y\%+(1-y\%)} = x\%$
- precision: $\frac{TP}{TP+FP} = \frac{x\%}{x\%+(1-x\%)} = y\%$

Model	accuracy	precision	recall
supervised	0.5338	0.37	0.37

Table 16: Expected values of metrics using supervised models on delivery truck trip data.

Model	accuracy	precision	recall
supervised	0.5	0.07	0.5

Table 17: Expected values of metrics using supervised methods on Brazilian E-commerce public dataset.

5.7.2 For supervised methods

For supervised learning methods, the definition of stochastic classification is more specific, this doesn't mean that the model does not learn any information, but only learns information about the proportion of the anomalous data.
That is, assuming that there are x% abnormal data in the training data, a stochastic method would have a x% likelihood of randomly classifying the data as abnormal when confronted with any data record in the data set in the test method, and a 1 - x% likelihood of randomly classifying the data as normal.
For delivery truck trip data
How the values of the metrics should be like when using random classification is shown in Table 16.
For Brazilian E-commerce public dataset
How the values of the metrics should be like when using random classification is shown in Table 17.

5.7.3 For clustering methods

For clustering methods, no assumptions are made about the proportion of anomalous data in the dataset because there is no training phase. This means that there is a 50% chance that any data record encountered will be randomly categorized as normal and a 50% chance that it will be randomly categorized as abnormal in a random classification scenario (y is 50).
As discussed earlier, the Brazilian e-commerce public dataset is very unbalanced. It would be interesting to explore the performance of clustering methods on balanced and unbalanced datasets respectively. There is a total of roughly 7% anomalous data in the unbalanced dataset. In the balanced data set, there are 50/50 percent of anomalous and normal data records.
For delivery truck trip data
How the values of the metrics should be like when using random classification is shown in Table 18.
For unbalanced Brazilian E-commerce public dataset

Model	accuracy	precision	recall
clustering	0.5	0.37	0.5

Table 18: Expected values of metrics using clustering models on delivery truck trip data.

Model	accuracy	precision	recall
clustering	0.5	0.07	0.5

Table 19: Expected values of metrics using clustering models on unbalanced delivery truck trip data.

How the values of the metrics should be like when using random classification is shown in Table 19. Since both supervised and clustering methods have a y-value of 50 and both supervised methods and unsupervised methods on unbalanced data have an x-value of 7, their expected values are equal across metrics.

For balanced Brazilian E-commerce public dataset

Model	accuracy	precision	recall
clustering	0.5	0.5	0.5

Table 20: Expected values of metrics using clustering models on balanced delivery truck trip data.

How the values of the metrics should be like when using random classification is shown in Table 20. Since both supervised and unsupervised methods have an x-value of 50, their expected values are different from the previous one.

6 Results and Analysis

In each of the subsections of this section, the results of supervised methods, clustering methods, one-class methods, and composite methods will first be compared, and try to explain the differences in the performances of the different datasets and methods. Also, if no parameter setting is mentioned for a method, this means that it uses the method's default parameters. The entire code of this graduation project and important results are saved on my GitHub site [1].

6.1 PCA

Before using common classification algorithms for the datasets, they are visualized and explored.

The full name of PCA is principal component analysis [64]. The main use of PCA is to reduce the dimensionality of the dataset. It is characterized by maintaining the differences between data records as much as possible in the process of dimensionality reduction. Here PCA is used for the main purpose of visualization by reducing datasets to 2D data. In the process, two principal components are obtained. The first principle component preserves the maximum difference between the data records and the second one preserves the maximum difference between importing the data while being orthogonal to the first one. The PCA function from sklearn.decomposition to accomplish this. The result of the dimensionality reduction is visualized as Figure 14 shows.

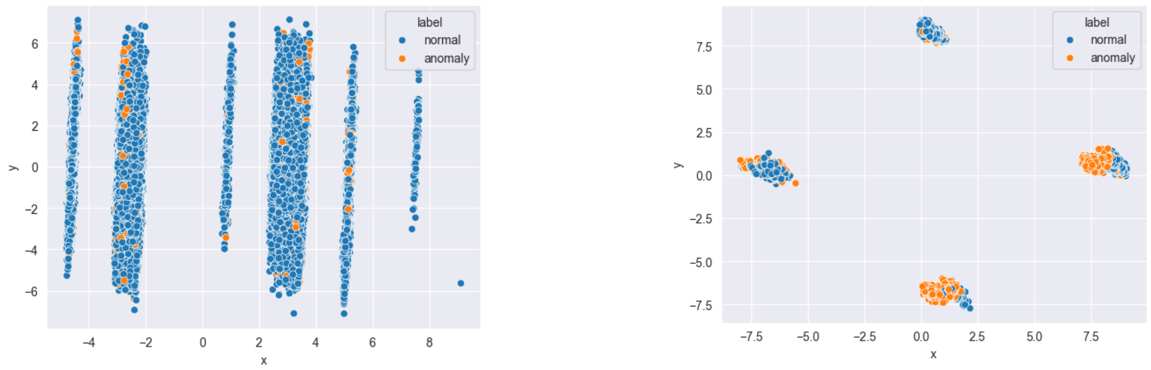

Figure 12: PCA result of Brazilian E-commerce Dataset.

Figure 13: PCA result of Delivery truck trip data.

Figure 14: PCA results

In this subsection, the balanced dataset of the Brazilian E-commerce dataset and the original Delivery truck trip data are chosen for the experiment. As can be seen in the figure, when reduced to 2 dimensions, the data are divided into clusters.

Ideally, PCA would split the dataset into two clusters, where one cluster is full of blue anomaly data and the other cluster is full of yellow normal data. However, it's found that there is a large amount of both normal and abnormal data in each cluster. This means that it is difficult for us to directly reduce the dimensionality of the data and summarize it into two or more

[1] https://github.com/Bowei-Gao/MasterGraduationProject.git

clusters by this method so that each cluster can be categorized into anomalous and normal data. The anomalies can not be identified through PCA visualization.
This can mean the following two findings:

1. PCA and visualization methods may not be suitable for classifying these two datasets.

2. Both datasets are too complex on their own, and reducing to two dimensions loses important information about correct categorization.

The second finding provides an important reference for determining the dimension of the latent space among autoencoders. It's found that such results for PCA imply that he may not be suitable as a tool for pre-processing the original data for downscaling.
Since the anomaly identification problem can be seen as a classification problem, other papers that downscale raw data to improve the speed and performance of subsequent classification models are searched. It's found that autoencoder has a wide range of applications for this purpose [65] [66] [67]. Therefore, the hope can be put on the encoder part of autoencoders to realize the dimensionality reduction to remove the noisy data and at the same time retain enough information to identify the anomalies later.

6.2 Supervised Methods

The supervised learning method used here consists of Table 21. As mentioned before, random forest and gradient-boosted trees are all based on decision trees and they are ensemble methods (ensembles of decision trees). The decision tree itself is not an ensemble method. And XGBoost (extreme gradient-boosting) is an implementation of gradient-boosting trees. Their relationship is shown in Figure 15. Metrics are represented in the confusion matrix in the format [[TP, FP] [FN, TN]].

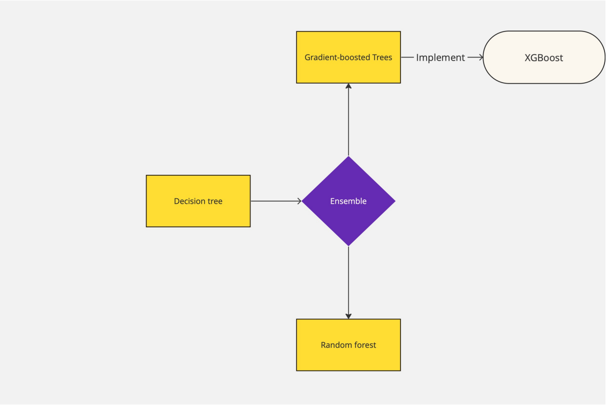

Figure 15: Hierarchy of decision tree based methods

For supervised learning methods, three metrics are used for evaluation: accuracy, precision, and racall. They are in order of importance:

$$recall > accuracy > precision$$

Because here, the main task is anomaly discovery. This means that as many anomalies as possible should be found. A high value of recall means that among all the exceptions, a

XGBoost	logistic regression	random forest	gradient-boosted trees
decision trees	naive bayes	support vector machine	

Table 21: Supervised methods used

Model	Confusion Matrix	accuracy	precision	recall
logistic regression	[[1102 208] [276 439]]	0.761	0.679	0.614
naive bayes	[[1039 271] [240 475]]	0.748	0.637	0.664
support vector machine	[[987 323] [225 490]]	0.729	0.603	0.685
decision trees	[[1217 93] [96 619]]	0.907	0.869	0.866
random forest	[[1258 52] [76 639]]	0.937	0.925	0.894
gradient-boosted trees	[[1266 44] [98 617]]	0.930	0.933	0.863
XGBoost	[[1258 52] [74 641]]	0.938	0.925	0.897
random		0.5338	0.37	0.37

Table 22: Results of Supervised Methods for delivery truck trip data.

higher percentage of them are found. Combined with the fact that the purpose of identifying anomalies is primarily to improve the supply chain. When an anomaly occurs it is often costly and if it can be avoided beforehand it can save a lot of resources. And for normal data records that are incorrectly recognized as anomalies. Once we encounter it, the time spent manually removing them in the anomalous dataset returned by the model is far less than finding the abnormality in the original dataset. Second, even without removing normal data records from the anomalous dataset, the resources spent on coping with these normal records should be far less than the resources lost to the anomalies.
For both of the datasets, each of these methods took less than 1 second to train as well as to test, so time and computational resources are not our main concern when evaluating these methods. The following focuses on their accuracy, recall, and precision.

6.2.1 Results and analysis of Delivery truck trip data

The results of the performances obtained using the supervised approach on this dataset are shown in the table 22. Based on their recall values, their performance ordering is shown in Table 23. Their performances based on accuracy and precision are similar. Decision-tree-based methods are all way ahead of the rest. This indicates that the other three models do not make enough correct modeling assumptions about probability. XGBoost is the best-performing method, and other ensemble methods based on decision trees perform slightly better than pure decision trees. Especially for the XGBoost method, it is close to or exceeds 90% for all metrics. In the case of this dataset, it is the recommended method when we have labels.

XGBoost	random forest	decision trees	gradient boosted trees	SVM
Naive bayes	logistic regression			

Table 23: method performances in descending order

On this dataset, synthesizing the feature importances of each supervised method, the important features manually chosen from this dataset were in Table 24. They will be used next as one of the inputs to the unsupervised learning method.

originLocation_code	vehicle_no	transportation_distance_in_km	vehicleType	supplierID
trip_time	day	minimum_kms_to_be_covered_in_a_day	month	year

Table 24: Manually selected features from delivery truck trip data

Model	Confusion Matrix	accuracy	precision	recall
logistic regression	[[17414 11650] [667 953]]	0.599	0.076	0.588
naive bayes	[[18505 10559] [622 998]]	0.636	0.086	0.616
support vector machine	[[14996 14068] [544 1076]]	0.524	0.071	0.664
decision trees	[[19747 9317] [448 1172]]	0.682	0.112	0.723
random forest	[[23016 6048] [401 1219]]	0.790	0.168	0.752
gradient-boosted trees	[[22158 6906] [396 1224]]	0.762	0.151	0.756
XGBoost	[[22391 6673] [368 1252]]	0.771	0.158	0.773
random		0.5	0.07	0.5

Table 25: Results of Supervised Methods for Brazilian E-commerce public dataset.

6.2.2 Results and analysis of Brazilian E-commerce public dataset

The results of the performances obtained using the supervised approach on this dataset are shown in the table 25. Their recall performances are ranked in Table 26. With this dataset, again XGBoost is the strongest method. The other conclusions summarized above are pretty much the same. The biggest difference is that it's found that even for XGBoost, accuracy, and recall are only 0.77, while precision is around 0.16. This suggests that finding anomalies in this dataset is harder than in the previous dataset.

XGBoost	gradient boosted trees	random forest	decision trees	SVM
Naive bayes	logistic regression			

Table 26: method performances in descending order

On this dataset, synthesizing the feature importances of each supervised method, the important features manually chosen from this dataset were Table 27.

6.2.3 Summary of findings from results of supervised methods

In the above, the results of using supervised learning methods were synthesized and analyzed on the problem of anomaly detection in the logistics domain. And there are the following findings.

1. XGBoost may be the best choice when there are labels provided.
2. Decision-tree-based methods usually perform better than other methods.
3. Decision-tree-based ensemble methods usually perform better than single decision-tree-based methods.

Until the accuracy, precision, and recall results for both datasets were obtained, it was still not clear that anomalies can be identified through categorization. However, after experimentation,

time_estimate_delivery	year	customer_geolocation_lng	day
seller_zip_code_prefix	freight_value	seller_geolocation_lat	seller_geolocation_lng
customer_zip_code_prefix	month	customer_geolocation_lat	distance

Table 27: Manually selected features from the olist dataset

it was found that on both datasets, the purely supervised learning approach gave better results than randomly classifying as anomalous and normal. This proves that the information on the datasets is sufficient to detect anomalies. As to why the algorithm performs better on individual metrics of Delivery truck trip data than on the Brazilian E-commerce public dataset, there are two guesses:

- It may be because when the Brazilian E-commerce public dataset was experimented with, there was a 50/50 split of abnormal and normal data in the training data to avoid bias, but in the test data there was only about 7% of abnormal data, which made the model more inclined to recognize the data as abnormal. This explains the very high false positive values as well as the very low precision.
- It may be that finding anomalies in the Delivery truck trip data is inherently simpler than in another dataset. If two datasets with the same method perform about the same in an unsupervised learning experiment, the former conjecture is verified, and if the performance on delivery truck trip data is much better than the other one, this conjecture is confirmed.

6.3 One-class SVM

When applying the One-class SVM method, the training set is a collection of data records that contain all normal data, and the test set is a collection of data that contains 50% of the training set as well as 50% of the test set. They are normalized before application so that each feature conforms to the normal distribution. And also scaling is done for each feature so that their values are in the same interval.
The Assignment of parameters is shown in Table 28. Here nu is an approximation of the outlier(anomaly) fraction in the training set and gamma is the kernel coefficient. When the value of gamma is 'auto', it is $\frac{1}{number\_of\_features}$.

gamma	nu	others
auto	0.05	default

Table 28: Parameters of one-class SVM

6.3.1 Random classification

Before discussing what a random assignment (when the algorithm only knows a pre-set percentage of anomalies and learns nothing) will yield here, there are a few important facts about random classification that are worth reviewing.

1. The training dataset contains all normal data.

2. The test dataset contains 50% normal data and 50% anomalous data.
3. Here the percentage of anomalous data is pre-set to 5

Based on the above facts, it will be found that a stochastic classification method that has not learned anything in the training process will assume that all but almost all (95%) of the data is normal. Thus based on the definitions of accuracy, precision, and recall, the values of these three metrics that a randomized classification algorithm would obtain are shown in the table 29.

Model	accuracy	precision	recall
One-class SVM	0.50	0.50	0.05

Table 29: metric values for a random classification

6.3.2 Results and analysis of Delivery truck trip data

Results of Delivery truck trip data using one-class SVM are shown in Table 30. The SVM results are also added as a comparison.

It can seen that due to the lack of labeling, the one-class SVM dataset performs nowhere near as well as SVM, which is already the weakest of the supervised learning methods. The value of the most important recall is only 0.216. The main reason for the low performances is that the abnormal data in the test set is much higher than in the training set, leading the model to assume that the vast majority of the DATA records are normal. This is an inherent problem that the one-class approach cannot avoid.

Model	Confusion Matrix	accuracy	precision	recall
SVM	[[987 323] [225 490]]	0.729	0.603	0.685
One-class SVM	[[806, 40], [663, 183]]	0.585	0.821	0.216
random		0.50	0.50	0.05

Table 30: Classification Report for delivery truck trip data

6.3.3 Results and analysis of Brazilian E-commerce public dataset by olist

Results of Delivery truck trip data using one-class SVM is shown in Table 30. The SVM results are also added as a comparison.

Here it's found that one class SVM model has almost no classification ability. accuracy and precision have a value of 0.5 which is not different from random classification and recall has a value of 0.05 which is the same as nu. Even if SVM performs poorly, it's still better than this method. This again proves how difficult it is to recognize anomalies without labels. Also, the results for this dataset are significantly worse compared to the previous dataset. This method on the last dataset still can detect anomalies. This, like the results in supervised learning, again proves that it is harder to recognize anomalies on this dataset.

However, as a benchmark, the results here demonstrate the meaningfulness of our performances using any unsupervised learning method that improves anomaly detection over random classification on this dataset.

Model	Confusion Matrix	accuracy	precision	recall
support vector machine	[[14996 14068] [544 1076]]	0.524	0.071	0.664
One-class SVM	[[5232, 272], [5233, 271]]	0.50	0.50	0.05

Table 31: Classification Report for the olist dataset

6.3.4 Summary of findings from results of one-class SVM

In the above, the results of using are synthesized and analyzed one-class SVM on the problem of anomaly detection on the logistics datasets. And there are the following findings.

1. Identifying anomalies using unsupervised learning methods is much more difficult than supervised learning methods.
2. Detection of anomalies on the Brazilian E-commerce public dataset by olist is much harder than Delivery truck trip data.
3. The results here can serve well as baselines.

There are also two important takeaways from this subsection of the experiment.

- Delivery truck trip data achieved better results than random classification, and there is at least one dataset that allows us to explore the effectiveness of other unsupervised anomaly detection methods and their variants for anomaly detection in the logistics domain.
- Even for delivery truck trip data, the recall of this algorithm is only 22%, so this method is not recommended for practical applications.

6.4 Autoencoders

In this project, autoencoders have two roles, one is to classify by one-class autoencoder to achieve anomaly detection; the other is to preprocess the original input dataset to improve the speed and accuracy of anomaly detection. For both purposes, we use the same structure of the autoencoder, but the input data and training process are different. The input data used for preprocessing is the original input data, and the input data used for one-class autoencoder is just a subset of the set of normal data. Because preprocessing has more and more complex input data, the number of training cycles will be greater. In this subsection, the autoencoders used as preprocessors are discussed.
Some important parameters are shown in the table 32.

Optimizer	learning rate	batch size
Adam	1e-4	64

Table 32: The parameters of autoencoders

6.4.1 Autoencoder for Delivery truck trip data

The structure of the autoencoder for Delivery truck trip data is shown in Table 33. After training for 1000 cycles, reconstruction errors are around 0.005 - 0.008. The latent space has 7 features.

Component	Layer 0	Layer 1	Layer 2
Encoder	(19, 15)	(15, 11)	(11, 7)
Decoder	(7, 11)	(11, 15)	(15, 19)

Table 33: The structure of autoencoder for delivery truck trip data

6.4.2 Autoencoder for Brazilian E-commerce public dataset

The structure of the autoencoder for Delivery truck trip data is shown in Table 33. After training for 300 cycles, reconstruction errors are around 0.003 - 0.007. The latent space has 6 features.

Component	Layer 0	Layer 1	Layer 2
Encoder	(18, 14)	(14, 10)	(10, 6)
Decoder	(6, 10)	(10, 14)	(14, 18)

Table 34: The structure of autoencoder for Brazilian E-commerce public dataset

6.5 Comparison between manual selected features and latent space

	Delivery truck trip data	Brazilian E-commerce public dataset
manually selected features	10	12
latent space	7	6

Table 35: Dimensionality of manually selected features and latent space

It can be seen that the autoencoder processed data has lower dimensionality on both datasets than after manual selection. This can lead to two effects:

- The data processed by the autoencoder is more helpful in mitigating the curse of dimension problem because it has a lower dimension.
- The data processed by the autoencoder may be over-compressed, resulting in the loss of some important information.

But for the second effect. When a closer look is taken at Tables 24 and 27, it can be found that after manually selecting features, there can still be a lot of content with duplicate information. For example, the distance can be inferred from the longitude dimension information, and there is also some overlap between the longitude latitude information and the zip code information. The correlations between features can be reduced using an autoencoder, so lower dimensions may not lead to lower performance in classification.

6.6 One-class autoencoder

When using one class autoencoders, it's assumed that the labels do not exist, so only the distribution of reconstruction errors for normal data records will be focused on, and the appropriate threshold is determined through the distribution graph, if the reconstruction error of a data record exceeds this threshold, it is considered to be an anomaly. In this section, the parameters and structures of autoencoders are set the same as in the previous section. The number of training cycles for both datasets is 300 Table 36.

Optimizer	learning rate	batch size	# cycles
Adam	1e-4	64	300

Table 36: The parameters of one-class autoencoders

6.6.1 Results and analysis on delivery truck trip data

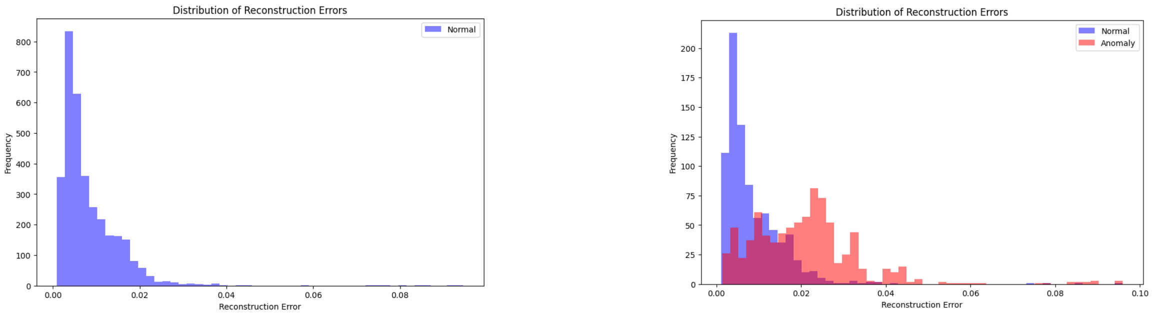

Figure 16: Distribution of errors on normal data records.

Figure 17: Distribution of errors on all records.

Figure 18: Distribution of reconstruction errors on Delivery truck trip data

By observing Figure 16, the threshold is set at 0.01. The distribution plot of Figure 17 verifies this threshold setting of mine. Performances of this model are shown in Table 37.

Model	Confusion Matrix	accuracy	precision	recall
One-class autoencoder	[[580, 266], [164, 682]]	0.746	0.719	0.806
random		0.50	0.50	0.05

Table 37: Classification Report for the delivery truck trip dataset

As an unsupervised learning method, this result is quite high. Recall even reaches 80%. Even when compared with supervised learning methods, it has higher recall values than all supervised learning methods that are not based on decision trees. This shows that the autoencoder has a strong understanding of this dataset. By the time the experiments had gotten this far, it was also expected that autoencoder would have improved the results of unsupervised and unsupervised learning considerably.

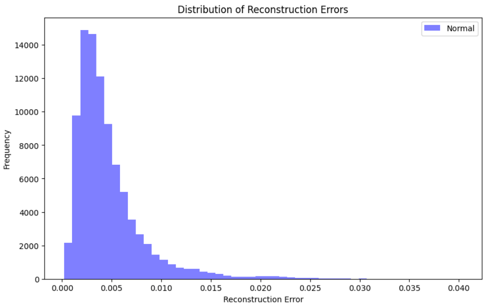

Figure 19: Distribution of errors on normal data records.

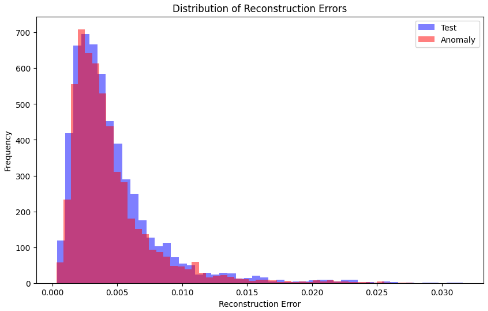

Figure 20: Distribution of errors on all records.

Figure 21: Distribution of reconstruction errors on Brazilian E-commerce public dataset

6.6.2 Results and analysis on Brazilian E-commerce public dataset

By observing Figure 19, the threshold is set at 0.01. But the distribution plot of Figure 20 suggests that it's impossible to use this method to do anomaly detection. This again proves that anomaly detection on this dataset is very difficult when there is no labeling to lead. It's also estimated that on this dataset auto encoder will not improve the performance of supervised and unsupervised learning. However, autoencoders can reduce the input data dimensions and thus reduce the training time. So even though it will make a slight loss in recall, precision, and accuracy, it is still interesting to study.

6.6.3 Summary of findings from results of one-class autoencoders

After analyzing one-class autoencoders, the following important findings and guesses are made.

1. In some cases, one-class autoencoders perform very well as an anomaly detection method. Even better than most supervised learning methods.

2. Autoencoder can be used as a preprocessor to improve the speed and performances of other methods on delivery truck trip data datasets.

3. Autoencoder can be used as preprocessing on olist datasets can improve the speed of other methods, but it is difficult to improve their performances.

The good performance of one-class autoencoders with unsupervised learning suggests that at least there is enough information to do anomaly identification in the absence of labels among Delivery truck trip data.
Although the one-class autoencoders method works well for delivering truck trip data, it has many shortcomings.

- On more complex datasets, one-class autoencoders are completely ineffective. So far only supervised learning methods have worked.

- Choosing the threshold itself is an esoteric matter, a lot of pictures labeled reconstruction error distributions are referred to before a good threshold is stumble upon, and if the project got very unfortunate and chose a less-than-good threshold, there would be a big problem.

algorithm	confusion matrix	accuracy	precision	recall
logistic regression	[[1195 110] [227 493]]	0.834	0.818	0.685
naive bayes	[[859 446] [204 516]]	0.679	0.536	0.717
support vector machine	[[1161 144] [194 526]]	0.833	0.785	0.731
decision trees	[[1163 142] [134 586]]	0.864	0.805	0.814
random forest	[[1235 70] [126 594]]	0.903	0.895	0.825
gradient-boosted trees	[[1222 83] [158 562]]	0.881	0.871	0.781
XGBoost	[[1217 88] [113 607]]	0.901	0.873	0.843
random		0.5338	0.37	0.37

Table 38: Results of autoencoder + supervised learning on Delivery truck trip data

- Select threshold This step cannot be reliably automated, which means that it will always require human intervention. So while the algorithm itself may not be slow, manual intervention takes a lot of extra time.

- Rolling out the same algorithm across logistics centers would add up to a long time and labor costs. At the same time, employees need to be trained to analyze reconstruction errors. This is also time- and labor-intensive. At the same time, if the employees are not specialized enough, it may lead to misjudgment. Because humans are not as reliable as computers.

It is for the many reasons above. Finding ways to a fully automated anomaly recognition in the logistics field without labels is needed. This is why the effort to explore the application of clustering methods and variants based on them for logistics anomaly detection continues in this project.

6.7 Autoencoder + supervised learning

For both of the datasets, each of the supervised methods took less than 1 second to train as well as to test, so time and computational resources are not our main concern when evaluating these methods. Even so, after preprocessing, supervised learning is still slightly faster.

6.7.1 Results and analysis on Delivery truck trip data

On this dataset, it's found that the performance of all decision tree-based supervised learning methods became a bit worse; the performance of supervised learning methods not based on decision trees became better. In the above results for one-class autoencoder, this model performs better than all supervised learning that is not based on decision trees, and worse than all supervised learning methods that are based on decision trees. More interestingly, the value of recall from the one-class autoencoder is 0.8, which happens to be in the autoencoder preprocessing. More interestingly, the value of recall obtained by the one-class autoencoder is 0.8, which is right between the value of recall with and without autoencoder preprocessing. All these signs point to the fact that autoencoder improves the performance of supervised methods approaches that are not based on decision trees.
Although autoencoder combined with supervised learning methods not based on decision trees gives a lower performance than pure autoencoder. However, this allows the anomaly-checking

algorithm	confusion matrix	accuracy	precision	recall
logistic regression	[[15804 13226] [646 1008]]	0.548	0.071	0.609
naive bayes	[[15521 13509] [707 947]]	0.537	0.066	0.573
support vector machine	[[13508 15522] [611 1043]]	0.474	0.063	0.631
decision trees	[[16442 12588] [647 1007]]	0.569	0.074	0.609
random forest	[[18462 10568] [581 1073]]	0.637	0.092	0.649
gradient-boosted trees	[[16594 12436] [568 1086]]	0.576	0.080	0.657
XGBoost	[[18422 10608] [600 1054]]	0.635	0.090	0.637
random		0.5	0.07	0.5

Table 39: Results of autoencoder + supervised learning on Brazilian E-commerce public dataset

to omit the manual selection of thresholds, making the anomaly-checking process more automated.

6.7.2 Results and analysis on Brazilian E-commerce public data

As expected earlier, autoencoders did not improve the prediction to accuracy, precision, and recall. this is because the autoencoder model does not understand this dataset correctly. Although training on this dataset is faster, this doesn't matter considering how fast these models are already.

6.7.3 Summary of findings from results using autoencoder + supervised methods

The results obtained using supervised methods with autoencoder preprocessing do provide interesting new findings.

1. After preprocessing with autoencoders, both the training and testing processes of supervised methods are indeed accelerated.

2. Autoencoders improve the performance of supervised learning methods that are not based on decision trees.

3. Unsupervised preprocessing (preprocessing by autoencoders) likewise fails on the olist dataset.

Also, it's found that on both datasets, for each method, the value of feature importances is high, indicating that in both datasets autoencoder implements compression as well as extraction of important information. As for the reason it is difficult to understand the data for anomaly detection by autoencoder, it may be because it is difficult to identify anomalies by the characteristics of the data itself in the absence of label guidance.
The reason is that the autoencoder improves SVMs, logistics regression, and Naive Bayes while degrading the performance of decision tree-based methods. It is probably due to the following two reasons.

- SVM, logistics regression, and Naive Bayes are three methods that are more susceptible to the curse of dimensionality than decision tree-based methods.

- In the process of autoencoder compressing the data into latent space, important information related to data anomaly identification is indeed lost, and the lack of information leads to a decline in the performance of decision tree-based methods that are not much affected by the curse of dimensionality.

But it's also important to see the performance of the decision tree-based methods does not drop much. This should indicate that not much information is lost in the autoencoder compression process. Clustering methods are extremely vulnerable to the problem of the curse of dimensionality, so I expect the performance of clustering methods clustering to improve after encoder preprocessing.

6.8 Clustering methods

mibs is a shorthand writeup of the mutual-information-based score, which I have abbreviated here because of the table presentation factor. The same abbreviation will be used for the next few subsections (6.8, 6.9, 6.10). Again, for presentation reasons, I've split the performance table for each method into two tables. One table contains the columns "confusion matrix", "accuracy", "recall", and "precision"; and the other table contains the columns "rand index", "mibs", "homogeneity ", "completeness", and "v-measure".

In this subsection and the following subsections, this thesis will focus on showing the performances of the same datasets with different preprocessing in different situations. Here preprocessing used contain:

- Only the necessary pre-processing mentioned in subsection 5.1 is carried out.
- The necessary preprocessing mentioned in subsection 5.1 + manual selection of features based on the value of feature importance in supervised learning.
- The necessary preprocessing that is mentioned in subsection 5.1 + latent space of autoencoders.

Here it's assumed that there is no prior knowledge of the labels. Being able to manually select features and then use clustering methods for categorization is certainly not possible in reality. The main reason to use manually selected features here is to compare it with the preprocessing of the autoencoder.

Both OPTICS and HDBSCAN can be viewed as variants of DBSCAN, and it's expected that the performances of both of them will be similar. Before experiments, there are the following expectations:

1. Applying the clustering method after the 2 additional preprocessing should be faster than without them.
2. The results of clustering classfication on an autoencoder processed dataset should be better than on an non-autoencoder-processed one.
3. Clustering classification on an autoencoder-processed dataset should be less effective than classification on a dataset pre-processed with manually selected features.
4. These methods should perform worse than supervised learning methods.

In this subsection, this thesis will focus on the results of applying the clustering classification method on a dataset that has not undergone either of those two preprocessing methods. Because spectral clustering [68] is too slow, no further experiments were performed on it.

6.8.1 Results and analysis on Delivery truck trip data

Results of pure clustering methods on delivery truck trip data are shown in Table 42 and Table 43. As expected, the performance of pure clustering models for anomaly detection is far inferior to supervised learning methods. The situation is similar in the following models, so we will not compare them with supervised learning methods here.
In comparing with one-class SVM and one-class autoencoder, The descending order of the performance of the methods is shown in the table 40.

one class svm	clustering methods	one class autoencoder

Table 40: method performances in descending order

As mentioned earlier, one-class SVM can hardly play a role in anomaly recognition, while one-class autoencoder outperforms all supervised learning methods that are not based on decision trees on this dataset, so this performance ranking is not surprising. The comparative picture of the performance of these three unsupervised learning on this dataset is the same regardless of the preprocessing. For another dataset, one class autoencoder also fails. Therefore, in this subsection and the next two subsections, these three unsupervised methods will not be compared, but the focus will turn to comparing models based on clustering methods.
Based on the value of accuracy, precision, and recall, the performance ordering of the clustering learning methods here is shown in table 41.

one class svm	clustering methods	one class autoencoder

Table 41: method performances in descending order

The two DBSCAN variants of the clustering method perform the worst, and as expected, they perform relatively similarly.BIRCH and Hierarchical clustering are the best, and their performance is surprisingly identical, which means that with high probability their clustering is the same for each data record, which is probably going to be because of the following two reasons:

- The core of Birch is also a hierarchical structure (i.e. a CF tree). Therefore, it is similar to the merge part of hierarchical clustering. Both are conceptually very similar.
- They both use the same metric, which is Euclidean distances between points.

Although the performances of the Birch and hierarchical clustering methods, especially recall scores, are quite good, the values of the rand index, mutual information-based scores, homogeneity, completeness, and v-measure still don't look promising.

6.8.2 Results and analysis on Brazilian E-commerce public Dataset

Results of pure clustering methods on the Brazilian E-commerce public Dataset are shown in Table 44 and Table 45. On the table here, it's shown that both the "clustering method" and the "clustering method balanced" model names. The model with balanced in its name is the experiment on a balanced dataset (50% normal records + 50% anomaly data records). And the ones without are the experiments on the unbalanced dataset (93% normal records + 7% anomaly data records). It can be seen that for the balanced dataset, each metric is above

algorithm	confusion matrix	accuracy	precision	recall
KMeans	[[2209 2021] [1038 1482]]	0.547	0.423	0.588
hierarchical clustering	[[2358 1872] [401 2119]]	0.663	0.531	0.841
BIRCH	[[2358 1872] [401 2119]]	0.663	0.531	0.841
HDBSCAN	[[3673 557] [2297 223]]	0.577	0.286	0.088
OPTICS	[[3045 1185] [1805 715]]	0.557	0.376	0.284
random		0.5	0.37	0.5

Table 42: Performances of unsupervised methods on Delivery truck trip data

algorithm	rand index	mibs	homogeneity	completeness	v-measure
KMeans	0.504	0.008	0.009	0.008	0.008
hierarchical clustering	0.553	0.123	0.125	0.122	0.124
BIRCH	0.553	0.123	0.125	0.122	0.124
HDBSCAN	0.512	0.004	0.003	0.006	0.004
OPTICS	0.506	-0.000	0.000	0.000	0.000

Table 43: Performances of clustering methods on Delivery truck trip data

and below 50%; for the unbalanced dataset, each metric is above and below 50% except for precision, which has a value of around 7%. This indicates that the clustering method is hardly effective in identifying anomalies by categorization.

6.8.3 Summary of findings of results using pure clustering models

In this subsection, the following important findings can be drawn.

1. BIRCH and Hierarchical clustering work well on the delivery truck dataset.
2. As unsupervised learning methods, pure clustering methods fail again on the dataset provided by olist.

algorithm	confusion matrix	accuracy	precision	recall
KMeans	[[44247 52529] [1982 3522]]	0.467	0.063	0.640
KMeans balanced	[[2505 2999] [1982 3522]]	0.548	0.540	0.640
hierarchical clustering balanced	[[2314 3190] [1872 3632]]	0.540	0.532	0.660
BIRCH	[[37827 58949] [1750 3754]]	0.407	0.060	0.682
BIRCH balanced	[[2207 3297] [1773 3731]]	0.539	0.531	0.678
HDBSCAN	[[76221 20555] [4639 865]]	0.754	0.040	0.157
HDBSCAN balanced	[[3809 1695] [4317 1187]]	0.454	0.412	0.216
OPTICS	[[54700 42076] [3025 2479]]	0.559	0.056	0.450
OPTICS balanced	[[2569 2935] [2987 2517]]	0.462	0.462	0.457
random		0.5	0.07	0.5
random balanced		0.5	0.5	0.5

Table 44: Performances of clustering methods on the olist data

algorithm	rand index	mibs	homogeneity	completeness	v-measure
KMeans	0.502	0.002	0.005	0.001	0.002
KMeans balanced	0.504	0.007	0.007	0.007	0.007
hierarchical clustering balanced	0.503	0.005	0.005	0.005	0.005
BIRCH	0.517	0.001	0.003	0.001	0.001
BIRCH balanced	0.503	0.005	0.005	0.005	0.005
HDBSCAN	0.629	0.001	0.002	0.001	0.001
HDBSCAN balanced	0.504	0.009	0.008	0.010	0.009
OPTICS	0.507	0.000	0.000	0.000	0.000
OPTICS balanced	0.503	0.004	0.004	0.004	0.004

Table 45: Performances of clustering methods on the olist data

Because both unsupervised classification methods of one-class autoencoder and clustering methods have not been able to outperform random assignment on the public dataset of Brazilian e-commerce. So the estimation is that the proposed method of combining autoencoder and clustering methods also fails to improve the performance of the algorithm on this dataset. The hope that unsupervised learning methods can outperform random categorization on this dataset lies in the manual selection of data preprocessing coupled with clustering methods.
It is also important to note that on the three metrics of accuracy, precision and recall, different clustering methods are calculated differently. They can be categorized into group 1 and group 2. The methods included in group 1 and group 2 are shown in the table 46.

- Group 1: This type of clustering method allows you to specify the number of clusters. The number of clusters can be specified as 2 and use the method described in Chapter 5 to label two clusters as anomalous and normal.
- Group 2: This type of method cannot specify the number of clusters. There will be more than 2 clusters. They label outliers (data points that are not in any of the clusters) as -1. Data points labeled -1 are considered to be anomalies. The problem with this is that, for example, outliers make up about half of the Delivery truck trip data. These anomalies are also clustered into clusters, and the data points in these clusters cannot be considered anomalies.

group 1	**group 2**
KMeans	HDBSCAN
hierarchical clustering	OPTICS
BIRCH	

Table 46: 2 groups of clustering methods

It is the way the metrics for the methods in group 2 are calculated that leads to the very low recall values for HDBSCAN and OPTICS. This characteristic may also make the method inherently impossible to apply in practice. However, it would be worthwhile to experiment and see if there is any improvement in their metrics by manually selecting features and autoencoder preprocessing.

algorithm	confusion matrix	accuracy	precision	recall
KMeans	[[2209 2021] [1038 1482]]	0.547	0.423	0.588
hierarchical clustering	[[2358 1872] [401 2119]]	0.663	0.531	0.841
BIRCH	[[2209 2021] [1038 1482]]	0.547	0.423	0.588
HDBSCAN	[[3791 439] [2047 473]]	0.632	0.519	0.188
OPTICS	[[3404 826] [2004 516]]	0.581	0.385	0.205
random		0.5	0.37	0.5

Table 47: Performances of manually selected features + clustering methods on Delivery truck trip data

algorithm	rand index	mibs	homogeneity	completeness	v-measure
KMeans	0.504	0.008	0.009	0.008	0.008
hierarchical clustering	0.553	0.123	0.125	0.122	0.124
BIRCH	0.504	0.008	0.009	0.008	0.008
HDBSCAN	0.535	0.013	0.010	0.017	0.013
OPTICS	0.513	-0.000	0.000	0.000	0.000

Table 48: Performances of manually selected features + clustering methods on Delivery truck trip data

Among the three methods in group 1, it's found that KMeans performs the worst. Although it still outperforms the completely randomized classification. It is expected that the following two preprocessing methods will improve its performance on each metric.

6.9 Manually selected features + Clustering methods

6.9.1 Results and analysis on Delivery truck trip data

Surprisingly, the cluttering methods failed to perform better after preprocessing with manually selected features. As shown in Table 47 and Table 48, the hierarchical clustering, which was performing better before, is now performing at the same level as before, but BIRCH is now performing worse. The KMeans method likewise performs at the same level as before. And OPTICS and BIRCH performed a little better. But considering the extremely low recall values of these two methods, this makes no difference. Taken together, hierarchical clustering is still the best method for anomaly identification on this dataset, but the absence of manually selected features hardly has any impact on the performance of clustering methods.
However, it is worth noting that due to the slower clustering methods. The model with manually selected preprocessing (10 features in total) is slightly faster than the one without such preprocessing (19 features in total).

6.9.2 Results and analysis on Brazilian E-commerce public Dataset

As shown in Table 49 and Table 50, the clustering methods also fail after manual selection of features preprocessing on this dataset. The clustering models are also slightly faster.

algorithm	confusion matrix	accuracy	precision	recall
KMeans	[[44247 52529] [1982 3522]]	0.467	0.063	0.640
KMeans balanced	[[2923 2581] [3522 1982]]	0.446	0.434	0.360
hierarchical clustering balanced	[[2463 3041] [1910 3594]]	0.550	0.542	0.653
BIRCH	[[42166 54610] [1903 3601]]	0.447	0.062	0.654
BIRCH balanced	[[2463 3041] [1910 3594]]	0.550	0.542	0.653
HDBSCAN	[[66326 30450] [4251 1253]]	0.661	0.040	0.228
HDBSCAN balanced	[[3450 2054] [4131 1373]]	0.438	0.401	0.249
OPTICS	[[59131 37645] [3280 2224]]	0.600	0.056	0.404
OPTICS balanced	[[2950 2554] [3357 2147]]	0.463	0.457	0.390
random		0.5	0.07	0.5
random balanced		0.5	0.5	0.5

Table 49: Performances of manually selected features + clustering methods on the olist data

algorithm	rand index	mibs	homogeneity	completeness	v-measure
KMeans	0.502	0.002	0.005	0.001	0.002
KMeans balanced	0.506	0.009	0.009	0.009	0.009
hierarchical clustering balanced	0.505	0.008	0.008	0.008	0.008
BIRCH	0.506	0.002	0.004	0.001	0.002
BIRCH balanced	0.505	0.008	0.008	0.008	0.008
HDBSCAN	0.552	0.002	0.005	0.002	0.002
HDBSCAN balanced	0.508	0.014	0.013	0.014	0.014
OPTICS	0.520	0.000	0.000	0.000	0.000
OPTICS balanced	0.503	0.004	0.004	0.004	0.004

Table 50: Performances of manually selected features + clustering methods on the olist data

6.9.3 Summary of findings from results using manually selected features + clustering methods

The results obtained by this method did not exactly match the expectations.

1. The "cheating" preprocessing method of manually selecting features based on the feature importance of the supervised methods did not lead to a better performance of the model.
2. Although the number of features was nearly halved on both datasets by manually selecting features, there was only a small increase in the speed of the model. It was expected to be half (algorithmic time complexity O(N)) or more.
3. Once again, unsupervised learning methods fail again on the Brazilian E-commerce public dataset by list.

Let's focus again on the results of the Delivery truck trip data. It is interesting to note that the performance of KMeans and hierarchical clustering remains the same across metrics. The other three methods, however, show different degrees of degradation. This suggests that the alleviation of the curse of dimension does not lead to better performance. However, since the performance of the first two methods remains the same, it is difficult to say that important information is lost by manually selecting features. Why the performance of the last three methods declined after the curse of dimension was mitigated would be an interesting (though probably not meaningful) topic for further research.

6.10 Autoencoder + Clustering methods

After getting the results of the last subsection, the expectation for the effectiveness of autoencoders is low. It was before this that applying an autoencoder would give the following two advantages.

- Because autoencoders reduce the dimensionality of the input data to less than half of what it was, I expect the model to have a large speedup.
- After preprocessing with autoencoders clustering models will perform better as supervised learning methods do after preprocessing.

But here it was found that after autoencoder preprocessing, the clustering model surprisingly outperforms both the pure clustering model and the clustering model with manually selected features preprocessing. Next, the detailed results and analysis will be shown.

6.10.1 Results and analysis on Delivery truck trip data

The results of using autoencoder preprocessing + clustering methods on Delivery datasets are shown in Table 52 and Table 53. First of all, it is worth mentioning that even after the autoencoder preprocessing, where the original input dataset is downsized by 50%, the speed of the final model does not differ much from the previous one, and is only slightly accelerated. This suggests that the speed of the model does not depend much on the dimensionality of the input data.

BIRCH	hierarchical clustering	KMeans	HDBSCAN	OPTICS

Table 51: method performances in descending order

algorithm	confusion matrix	accuracy	precision	recall
KMeans	[[2366 1864] [406 2114]]	0.664	0.531	0.839
hierarchical clustering	[[2353 1877] [391 2129]]	0.664	0.531	0.845
BIRCH	[[2351 1879] [385 2135]]	0.665	0.532	0.847
HDBSCAN	[[3926 304] [2261 259]]	0.620	0.460	0.103
OPTICS	[[3152 1078] [1908 612]]	0.558	0.362	0.243
random		0.5	0.37	0.5

Table 52: Performances of autoencoders + clustering methods on Delivery truck trip data

Surprisingly, after preprocessing with autoencoders, the performance of most of the models was largely improved in all metrics. Based on the values of the individual indicators, the performance is shown in table 51.

The accuracy of KMeans, hierarchical clustering, and BIRCH all reached more than 66%, and the recall for anomaly reached 84%! This is higher than one-class autoencoders and supervised learning methods that are not based on decision trees. At the same time, the accuracy of the HDBSCAN model has been greatly improved to 62%. After such preprocessing, the methods that got the most improvement were KMeans and HDBSCAN, but almost all other metrics were improved.

Compared to the one-class autoencoder which only uses the autoencoder architecture, the accuracy is still lower though, and additional clustering methods need to be utilized. but here, there is no need to manually set the threshold, which is a metaphysical thing to set. The whole process is automated this way. Determining the threshold by observing the distribution of reconstruction errors also takes extra time, and a human being can't observe, analyze, and come to a conclusion as fast as a machine can execute an additional algorithm. So the method of autoencoder + clustering methods saves a lot of time in anomaly detection, even though it adds an extra step.

6.10.2 Results and analysis on Brazilian E-commerce public Dataset

After studying the results on the Brazilian E-commerce public dataset Table 54 and 55. it's found that even the most powerful unsupervised models tried so far, namely autoencoder + clustering methods, have not been able to make a meaningful discovery of anomalies on this

algorithm	rand index	mibs	homogeneity	completeness	v-measure
KMeans	0.554	0.123	0.125	0.122	0.123
hierarchical clustering	0.554	0.126	0.127	0.124	0.126
BIRCH	0.554	0.127	0.129	0.126	0.127
HDBSCAN	0.529	0.003	0.002	0.005	0.003
OPTICS	0.507	0.000	0.000	0.000	0.000

Table 53: Performances of autoencoders + clustering methods on Delivery truck trip data

algorithm	confusion matrix	accuracy	precision	recall
KMeans	[[44205 52571] [1980 3524]]	0.467	0.063	0.640
KMeans balanced	[[2489 3015] [1980 3524]]	0.546	0.539	0.640
hierarchical clustering balanced	[[2495 3009] [1983 3521]]	0.547	0.539	0.640
BIRCH	[[43910 52866] [1987 3517]]	0.464	0.062	0.639
BIRCH balanced	[[1648 3856] [1526 3978]]	0.511	0.508	0.723
HDBSCAN	[[55654 41122] [3603 1901]]	0.563	0.044	0.345
HDBSCAN balanced	[[2775 2729] [3265 2239]]	0.455	0.451	0.407
OPTICS	[[63715 33061] [3571 1933]]	0.642	0.055	0.351
OPTICS balanced	[[3288 2216] [3592 1912]]	0.472	0.463	0.347
random		0.5	0.07	0.5
random balanced		0.5	0.5	0.5

Table 54: Performances of autoencoder + clustering methods on the olist data

algorithm	rand index	mibs	homogeneity	completeness	v-measure
KMeans	0.502	0.002	0.005	0.001	0.002
KMeans balanced	0.504	0.006	0.006	0.007	0.006
hierarchical clustering balanced	0.504	0.006	0.006	0.007	0.007
BIRCH	0.503	0.002	0.004	0.001	0.002
BIRCH balanced	0.500	0.000	0.000	0.000	0.000
HDBSCAN	0.508	0.002	0.003	0.001	0.002
HDBSCAN balanced	0.504	0.006	0.006	0.006	0.006
OPTICS	0.540	0.000	0.000	0.000	0.000
OPTICS balanced	0.501	0.002	0.002	0.002	0.002

Table 55: Performances of autoencoder + clustering methods on the olist data

dataset.

6.10.3 Summary of findings about this model

The discovery of this subsection was the most important, and the autoencoder + clustering methods yielded good results that is not expected.

1. After preprocessing with autoencoders, clustering methods perform significantly better than without preprocessing on simpler logistics datasets like delivery truck trip data. Among them, the KMeans method shows the most obvious improvement, while the performance of hierarchy clustering and BIRCH, which are already very good, is further improved. It is also slightly faster after preprocessing.

2. The model of autoencoder + clustering has even better recall metrics than one-class autoencoder on a simple logistics dataset, and in practice, although there is one more step, it will still take less time than one-class autoencoder.

3. Even though the model of autoencoder + clustering is very powerful, it is still not possible to recognize anomalies in the Brazilian E-commerce dataset by olist without labels. This

suggests that it may be very difficult to perform anomaly recognition without labels on a complex logistic dataset such as the Brazilian E-commerce dataset by olist.

Using autoencoder for preprocessing versus the previous preprocessing of manually selecting features, it's found that autoencoder got less input data for the later model than just the number of features, thus making the later model faster. At the same time, it performs better than the model with manually selected features or without any dimensionality reduction preprocessing in all indicators, which fully demonstrates that the effectiveness of this method in the field of logistics anomaly recognition is not trivial.
Recalling all the unsupervised experiments, it can also be argued that the reason the same approach at the time of the previous unsupervised experiments did not perform well on the Brazilian E-commerce dataset is because the anomalies in it are inherently more difficult to recognize. The fact that the training data is a 50/50 split between anomalous and normal data records and the test data has only a few anomalies does not have a significant impact.
Let's focus again on the results of the Delivery truck trip data, comparing the pure clustering approach model (Table 42 and Table 43) with the encoder + clustering approach model (Table 52 and Table 53).
It's found that each method improved its performance over every metric except OPTICS, which had only a negligible slight decrease. Combined with the conclusion in the subsection of autoencoder + supervised methods (autoencoder can largely alleviate the curse of dimensionality, but with a slight loss of anomaly information), it can be concluded that anomaly detection in logistics may be bacause the following algorithms are indeed the best in terms of performance. It can be concluded that the performance of clustering methods in the field of logistics anomaly detection may be greatly improved due to the following reasons.

- The problem of curse of dimensionality is alleviated.
- Reduces the correlation between the features of the data.
- Remove the noise from the data.

For the two clustering methods in group 2, we find that even with the improvement of autoencoders, they both yield recall values below 30% (methods in group 1 all have recalls higher than 80%), and are therefore of little practical use. This means that the large number of anomaly records in the dataset are also clusters, so they cannot be considered as outliers. This improves our understanding of the distribution of data in an n-dimensional space in cases where we cannot effectively visualize the dataset.
For the clustering methods in group 1, after the enhancement of autoencoder, their recall value reaches about 84%, and the accuracy reaches more than 66%, which fully demonstrates the power of the autoencoder + clustering model. Compared with the one-class autoencoder, the recall value of the methods in group 1 is larger than that of the one-class autoencoder (80.6%). Another advantage it has over the one-class autoencoder is that we do not need to set a threshold manually, which avoids the problems mentioned in the one-class autoencoder section.

6.10.4 Our recommended approachs and a ensemble method

Since there is little difference in the performance of this model for the three methods in group 1, all three methods are recommended. In practice, I think it is also possible to create an ensemble method to enhance the stability and performance of the algorithm. This is shown in the figure 22. In the first step, the latent space obtained by the encoder can be shared by the next three clustering methods, so it does not consume more time and computational resources. In the second step, the three clustering methods can run in parallel. This makes the time spent in the second step equal to the longest time spent by the three clustering methods. The time complexity of the last step of the voting algorithm is O(N). The extra time consumed is not much. The extra computational resources are mainly consumed in the second step, but with the increasing power of parallel computing chips and tools, these do not consume much resources. This means that we get a more stable and reliable unsupervised logistics inspection system without spending much extra time and computational resources.

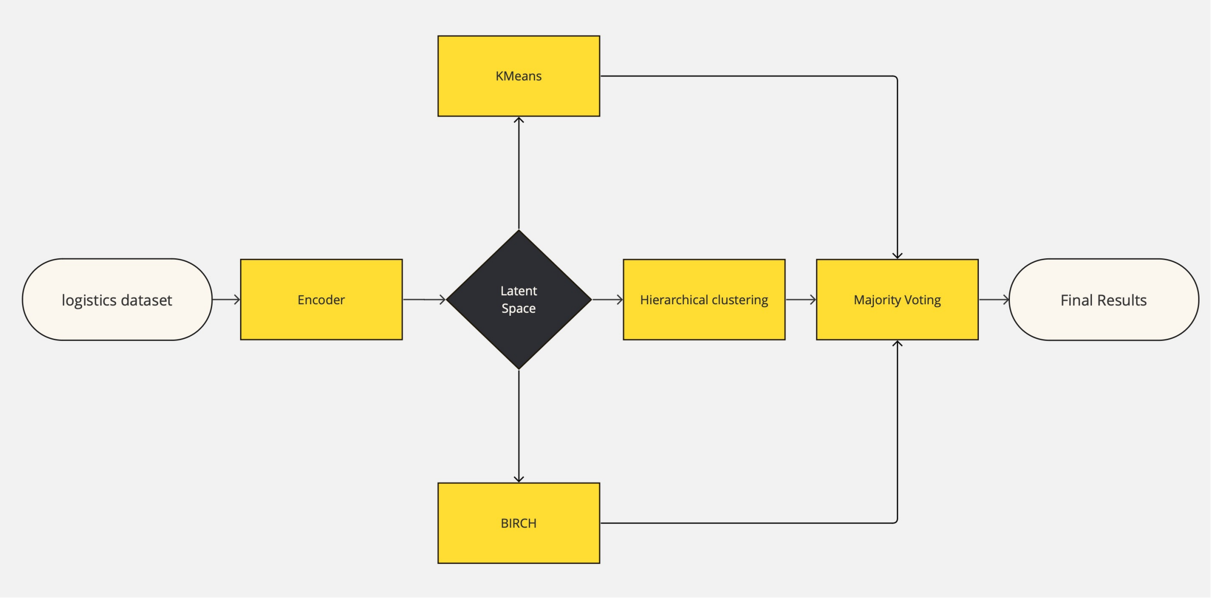

Figure 22: A ensemble methods of autoencoder + clustering methods

The application of this model to anomaly detection in logistics will not be explored in this final project. Further research on this model would be an interesting direction for future works.

7 Important logistical factors

The results in this section come mainly from the study of the values of feature importances in experiments on both datasets using supervised methods. For each method, see the Appendix for specific experimental results for feature importances. It is important to note that a lot of features is removed in the preprocessing that are believed to be completely irrelevant to whether a package arrives on time or not, and only those features that common sense suggests are relatively relevant to whether a package arrives on time or not are discussed, but the experiments have shown that the relationship is not very strong.
The two main methods used to get these important features are as follows.

- Guess what are the possible important and unimportant features by looking at the feature importances.

- Experiment based on observation. Remove those features that are thought to be not important and see if the performance of the supervised learning method declines. See how the performance of the supervised learning method changes when some features (e.g., latitude/longitude pairs) are removed or retained at the same time.

Understanding which features have an impact on the accuracy of a supervised learning model can help to reverse-engineer important factors in the logistics process, and by focusing on these factors, it is possible to reduce the anomaly rate of the logistics network, improve efficiency, and reduce the potential for additional overhead.
First, let's take another look at the features summarized on both datasets that are more relevant and relatively irrelevant to whether logistics arrive accurately or not.

7.1 Relevant features

Features that are relatively relevant are in Table 56 and Table 57:

originLocation_code	vehicle_no	transportation_distance_in_km	vehicleType	supplierID
trip_time	day	minimum_kms_to_be_covered_in_a_day	month	year

Table 56: Relatively relevant features from delivery truck trip data

time_estimate_delivery	year	customer_geolocation_lng	day
seller_zip_code_prefix	freight_value	seller_geolocation_lat	seller_geolocation_lng
customer_zip_code_prefix	month	customer_geolocation_lat	distance

Table 57: Relatively relevant features from Brazilian E-commerce dataset

7.2 Irrelevant Faeatures

Features that are relatively irrelevant aren in Tabke 58 and Table 59:

Market/Regular	destinationlocation_code	Org_lat	Org_lon	dayofweek
customerID	Material Shipped	Des_lat	Des_lon	

Table 58: Relatively irrelevant features from delivery truck trip data

price	product_category_name	product_weight_g	product_length_cm
product_height_cm	product_width_cm		

Table 59: Relatively irrelevant features from Brazilian E-commerce dataset

7.3 Relevant factors and analysis

Looking at the features in each of the above tables, the following conclusions can be drawn:

1. The exact day, month, and year of shipment have a relatively large impact on whether or not it is an anomaly, and it's believed that unexpected events such as wars, epidemics, etc. may have affected the delivery of the product at the expected time. But the specific day of the week (Monday, Tuesday, ... Sunday) has little effect. Because there is no strong correlation between the occurrence of a sudden event and the day of the week.

2. There is a strong positive correlation between the two metrics, distance and expected time. Both metrics have a strong impact on whether or not a delivery is made on time.

3. In addition, features such as zip code, longitude, and dimension, which show the exact location of the sender and receiver, are also critical. It may be because different places have different infrastructure development, traffic congestion, and ease of reaching other places, which leads to different probabilities of anomalies in different places.

4. Also, the identity of the sender has a greater impact on whether or not it is delivered on time, but the identity of the receiver has little impact. This is because the expected delivery time is calculated when the order is created. And the sender has a different habit of sending the package to the carrier. Some have the habit of sending the package to the carrier soon after the order is generated, while some have the habit of delaying it. So the probability of an exception is different for different suppliers.

5. minimum_kms_to_be_covered_in_a_day also has an impact on service. When it is requested, it may indicate that the order is more urgent and more likely to be delayed thus increasing the probability of an exception occurring; it may also cause the carrier to prioritize its shipment, thus decreasing the probability of an exception occurring.

6. Meanwhile, specific information about the vehicles used for transportation has a significant impact. Such an impact could come from two sources. The first aspect is related to the vehicle and the driver, different drivers' driving skills and habits and other factors affect whether the order can be delivered on time; on the other hand, it is also related to the vehicle, the reliability of the vehicle, the range of the vehicle all affect the transportation process.

7. In addition, it's found that the price, size, and material information of the goods being shipped itself had very little effect on whether or not the goods arrived on time. This is something that is not expected. This is probably because the process of transportation

itself is the same no matter what the goods are, so it has little effect on whether or not an exception to the order occurs.

8. Last but not least, whether or not the sender of the order has a standing order with the logistics company (Market/Regular) has very little influence on whether or not it arrives on time, probably also because it has very little to do with the logistics process itself.

It should be noted that the above analysis of the causes of each category is only conjectural. Further experiments are needed to verify the accuracy of these conjectures.

The author hopes that this section's observation and discussion of the features that influence anomalies will be useful in further discovering what factors influence whether or not an anomaly occurs. More resources should be allocated to the study of these factors in logistics planning to reduce the occurrence of anomalies in the logistics network.

8 Conclusion

In this thesis, the problem of how to perform anomaly detection in the field of logistics is studied. Two logistics datasets obtained from real-life logistic operations are used to test the speed and performance (accuracy, recall, etc.) of various standard supervised and unsupervised learning methods in different situations. Both additional preprocessings are performed on the raw input data by manually filtering the features and utilizing the encoder part of the autoencoder. They are compared and analyzed with the speed and performance of various standard methods. The features importances obtained from supervised learning and combinations of features are tested to find out which features help the model to detect anomalies, and thus to work out which factors in the logistics domain determine whether a delay in the arrival of a shipment occurs or not, based on the experimental data. Here are the important conclusions drawn.

8.1 Relationship between methods and complexity of the logistics datasets

In this graduation project, Delivery truck trip data is a simpler dataset, and Brazilian Ecommerce public dataset by list is a more complex dataset. Reviewing the preprocessing of the complex dataset, it's needed to merge multiple data tables and remove a lot of irrelevant features, and it's found that all supervised and unsupervised learning methods can significantly outperform the randomized classification on the logistics dataset for the simple dataset, and the supervised learning methods still outperform the randomized classification on the dataset by a wide margin but the classification results of the unsupervised learning are almost as good as the randomized classification on the logistic dataset. learning results are close to random classification. The lesson learned here is that supervised and unsupervised learning methods can be used on simple datasets, but must be required to be labeled for meaningful training on complex datasets. In the following summary, when talking about supervised learning methods, the common patterns obtained on complex and simple datasets are discussed. When summarizing the speed and performance of unsupervised learning methods, only the laws of unsupervised methods on simple datasets are discussed. There is no point in discussing the performance of unsupervised methods on complex datasets.

In practice, predicting the outcome of complex datasets will not be difficult as long as we have enough labels to be able to train the model. Many articles have also applied supervised learning models in the field of anomaly discovery with positive results [69] [70] [71].

8.2 Analysis of the results of methods

8.2.1 Analysis of the results of supervised methods

The first thing to say is that on each dataset, combined across each metric, supervised learning methods outperform unsupervised learning methods. Supervised learning methods can be categorized into decision-tree-based supervised learning methods and other supervised learning methods. Decision-tree-based supervised learning methods are decision trees, random forest, gradient boosted trees, and XGBoost; other supervised learning methods are logistic regression and support vector machine. All the methods based on the decision tree method perform better than the other methods. The underlying reason could be that some assumptions of logistic regression and support vector machine classification are not fully valid on the logistics dataset. Among the decision-tree-based methods, Gradient-boosted trees and XGBoost, an implementation of the former, perform better than the other two decision-tree-based methods. Especially XGBoost performs best on both datasets. Supervised learning methods, on the other hand, take less than a second to train and test together, so their performance metrics such as accuracy and recall are the main concern here. Summarize the performance rankings for supervised learning methods as follows: $Gradient\ boosted\ trees\ and\ its\ implementations > other\ decision\ treesbased\ methods > non\ decision\ trees\ based\ models$. So to summarize, the conclusions can be drawn:

1. Supervised learning for anomaly detection in logistics works well, not only for relatively simple and easily recognizable datasets but also for complex datasets.

2. Among the supervised learning methods, the decision tree-based methods obtain far better results in the field of logistics anomaly detection.

3. XGBoost performs best on both simple and complex datasets and is therefore my recommended method for logistics anomaly detection.

It's also found that after preprocessing with autoencoders, the performance of supervised learning methods not based on decision trees gained a lot of improvement, but the performance of supervised learning methods based on decision trees did not improve much or even decreased 60. Since the problem can be viewed as a classification problem, It's found from reading the literature [32] [72] that autoencoder can also improve the performance of supervised learning methods that are not based on decision trees, among general classification problems. Thus the findings here are consistent with the previous paper, but the validity of the same approach for anomaly detection in the logistics domain is tested here. Previous articles [73] [74] have also found that preprocessing by autoencoder improves the classification accuracy of decision tree-based algorithms, but I have not been able to validate this phenomenon in the logistics domain.

It is also understandable that the results of the DBSCAN-based HDBSCAN and OPTICS are unsatisfactory, since in both methods the number of classes after the algorithm clusters can not be specified. It's only doable to consider outliers as anomalies. However, since there are half abnormal and half normal data records in the dataset, the abnormal data records will also form clusters, and we can't mark such clusters as abnormal, so understandably, the recall values of these two methods are low.

Table 60: Models that autoencoder improve and worsen

Improve	Worsen
support vector machine	XGBoost
logistic regression	random forest
naive bayes	gradient-boosted trees
	decision trees

8.2.2 Analysis of the results of unsupervised methods

Among all standard unsupervised learning methods, the best performers are one-class autoencoders. At least on the dataset used it outperforms even all supervised learning methods that

are not based on a decision tree. This shows that the encoder part of autoencoders can accurately understand the features of the input data and summarize them. The worst performer is one-class SVM, which has a recall value of only about 20%. It is also much worse than the support vector machine in supervised learning, which reinforces the important positive impact of labeling on training logistics anomaly discovery models. Among the clustering methods, hierarchical clustering and BIRCH perform the best with an accuracy of 66%, while recall reaches 84%. K-Means results are also significantly better than random classification. However, OPTICS and HDBSCAN, which are variants of DBSCAN, perform similarly, with recall below 20%. At the same time, it was found that using manual selection of important features did not improve the accuracy of clustering methods and only slightly improved the speed of classification. Also, regardless of the additional preprocessing used, the spectral clustering originally planned to be used took more than 1 day to obtain the results, and it's not realistic to perform the computation for such a long period in a real-world application, so experiments using it are suspended.
It is also interesting to note that the improved performance of the K-Means method is similar to that of the BIRCH method and the Hierarchical clustering method, which are already better. This is likely an indication that all three of the previously mentioned methods reached the limits of comprehension that the data allowed without the guidance of the labels. This boosts confidence in the approach of combining autoencoder and clustering methods.

8.2.3 Analysis of the results of autoencoders + clustering methods

The proposed method of combining autoencoders with clustering methods has yielded excellent results, improving the vast majority of the metrics of the results of all clustering methods. However, even though it was possible to downscale the data to about 50% of its original size, there was still only a small increase in model speed. The clustering method that autoencoders improve the most is KMeans, which improves KMeans to about the same performance as the two best methods, BIRCH and hierarchical clustering. I think the main reason may be that it solves the problem of the curse of dimensionality of this model by dimensionality reduction. Also, it's found that autoencoders can improve the accuracy of support vector machine and logistic regression, but will reduce the accuracy of decision-tree-based methods. This is because the first two supervised learning methods are a bit more affected by the curse of dimensionality. Such a result has been consistent with the previous finding [75] [76] [77] that the accuracy of clustering methods clustering can be improved by autoencoder. Such conclusions are extended here to the field of logistics anomaly detection.

8.2.4 Manual selection of features vs. preprocessing via autoencoder

Before arriving at the results, it's expected that mitigating the curse of dimensionality by manually selecting important features for dimensionality reduction will improve the accuracy, precision, and recall of clustering methods while speeding them up while solving the curse of dimensionality problem by preprocessing with an autoencoder will improve the accuracy, precision, and recall of the clustering methods. The curse of dimensionality approach mitigates the curse of dimensionality problem, but loses important information for finding logistic anomalies, and therefore performs better than the method without preprocessing, but worse than the method with manual feature selection.
However, after some experimentation, it's found that although the speed of clustering is slightly improved after selecting features manually, the accuracy, recall, and rand index, which are the

indicators we care more about, are maintained at the original level or slightly decreased. This shows that although manually selecting features can reduce the dimensionality of the original data, it does not improve the performances of the clustering algorithms. However, as mentioned earlier, the data obtained through the autoencoder preprocessing is of lower dimensionality and provides a greater (though still not significant) speedup to the clustering method. At the same time, it improves the algorithm's key metrics compared to the clustering algorithm without preprocessing. This further demonstrates the effectiveness of autoencoder to solve curse of dimensionality by downscaling the data while maintaining important information. It also suggests that the autoencoder is likely to reduce noise and mitigate correlation between features as well, compared to manual methods that also reduce dimensionality but do not improve algorithm performance. Anthropomorphically, it "understands" the original input data better.

8.3 Relevant logistics factors

The major difference between this thesis and previous research [78] [79] [80] about factors that influence logistical anomalies occurrance rate is that more of a quantitative rather than a qualitative methodology is used to examine this issue.

It's found that the exact day a shipment is sent out has a greater impact on whether or not the shipment arrives on time, but the day of the week that the shipment is sent out has less of an impact. Information about the exact location of the receiver and the sender and the distance between them to the shipment affects whether the shipment delivery time is delayed or not. The identity of the sender also has a significant effect, but the identity of the receiver has little effect. Whether or not a shipment order has a minimum daily distance requirement affects whether or not a shipment record is delayed, but whether or not the order is from vendors with long-term contracts has little effect. Delivery vehicles and delivery personnel are likely to have a significant impact on whether or not the delivery is on schedule.

Information about the goods themselves is not a significant factor, which is a surprising finding for me. This is because from the point of view of the logistics process if the same logistics service is chosen, the size and weight of the goods are only affected in the process of the sender handing over the goods to the carrier and in the process of the deliveryman sending the goods to the receiver. It does not affect the process in between. However, the time spent on these two processes is a very small percentage of the overall logistics process time, so size and weight do not affect whether the goods are delivered on time or not.

8.4 Future work

Future research could delve deeper into the reasons for the positive impact of autoencoders on clustering methods and supervised learning methods that are not based on decision trees in the logistics anomaly detection domain. Verify if it is due to the curse of dimensionality, which would require more available datasets to realize.

Regarding the study of the causes affecting the logistics process and thus the delays in the arrival of goods, previous researchers have focused in a different direction than the direction in this thesis. Many of the papers think in terms of the builders and planners of logistics systems, for example, [81] [82]. Papers focusing on improving processes from the perspective of a logistics company, on the other hand, are studied more qualitatively [78] [79] [80].

The factors found in this paper are associated with greater influence on whether logistics records become anomalies. It's possible can go deeper to discover the reasons behind their phenomenon and use experimental methods to observe such as and can reduce the anomaly rate of the logistics network by controlling these factors. And it might be interesting to study how to consider these factors more in the ETA prediction model to make the prediction more accurate.

The Ensemble approach, which is proposed in 6.10.4, is also an interesting direction for further research. Similar approaches have been proposed by authors in previous work and applied to areas such as intrusion detection [83] [84] [85]. What is proposed here can be considered as an extension in the field of logistics. With the development of graphics cards and parallel computing, the extra computational resources and time consumed by the proposed approach of using KMeans, BIRCH, and Hierarchical clustering in parallel are almost negligible. However, such an approach will greatly improve the stability of the algorithm and may also correct the otherwise misclassification errors thereby improving the performance in recognizing anomalies. The following three areas will be worth looking into.

- Does the ensemble method improve stability and anomaly detection performance over the three methods that comprise it?
- How much more time did it take, and did it take a similar amount of time compared to the clustering algorithm that took the longest of the three methods that comprised it?
- What computational resources (CPU, GPU...) should be used to perform this task?

It might also be interesting to look at KMeans, BIRCH, and Hierarchical clustering at a finer level of granularity, where it's possible to examine exactly what data records diverge from each other as anomalies and how to account for these divergences. In the experiments, BIRCH and Hierarchical clustering perform almost identically in classification, and it's now possible to investigate whether this means that they have almost the same opinion on whether each data record is anomalous or not. If the categorization of each data record is the same in the vast majority of cases for both methods. That may leave KMeans with a weak voice in the final voting system, so the weights of each method can be adjusted appropriately. As for the KMeans clustering method, it's possible to see exactly where it diverges from the other two methods. The analysis of specific data records may have the following possibilities.

1. The vast majority of the data records which it disagrees with the other two methods are in which the other two methods are right.
2. Of the data records in which it disagrees with the other two methods, there are many cases in which it is the only one that is right, although it is right for fewer records than the other two methods.

The first thing worth stating is that since the ensemble method's determination of whether each data record is an anomaly or not depends mainly on the voting system, KMeans will not have any negative impact on the ensemble model even if the case in 1 occurs. But if it is the case in 1, it may not make much sense to include it in the ensemble method. In the case of 2, KMeans is a good addition to the overall ensemble model. Even when it makes a mistake, the voting system behind it prevents the error from being transmitted to the final anomaly recognition result.

It's a pity to not experiment with the spectral clustering method because it takes too long to run and does not make sense in practical applications. In future research, the problem of accelerating the performance of spectral clustering for anomaly detection in practical applications will be worth investigating, and if it can achieve far better results than other methods, it may be of practical significance in the case of anomaly detection in logistics that is not time sensitive. Also how to accelerate it is an interesting area of research. In the article [86], the authors proposed the RESKM framework to accelerate spectral clustering. It is possible to explore whether this framework applies to anomaly detection in the logistics field, whether it can speed up spectral cluttering to a level that is valuable in practical applications, and the performance of anomaly detection. It would also be interesting to integrate it into the previously mentioned ensemble method and explore its contribution to the ensemble method. If it is integrated into the system, the time spent by the other algorithms is almost negligible. But if the other algorithms can help improve the performance and stability of the algorithm, it would be a very worthwhile endeavor.

In addition, it's found that in the process of logistics delay or even supply chain research, many of them start from experience or surveys of logistics practitioners, and draw phenomena and conclusions through certain analysis. More research is done through qualitative methods. Introducing more quantitative ideas and experiments may bring more novel findings. For example, when executing this graduation project, it was previously hypothesized that factors such as the weight, volume, and price of goods would greatly influence whether or not anomalies occur in the flow of goods, but it's found that these factors did not have much impact on the prediction of anomalies, and thinking backward it means that the information of the goods itself does not have much impact on the occurrence of logistics anomalies. During the author's search for an internship last year (2023), he also found that there were very few data scientist positions studying supply chain issues, far less than the supply chain's share of the overall economy (10%). In reading papers, it's also found far fewer papers studying anomaly discovery in logistics than those studying anomaly discovery in areas such as interconnection networks, imagery, and clearing networks. This may be because staff and researchers working on the Internet, image recognition, and clearing networks have more background in information technology. However, logistics does not have less impact on society than image recognition or clearing networks [87]. Therefore, combining data science and logistics management issues can be an area of opportunity for both individuals and society.

References

[1] Bert Hofman. "Performance and Prospects of Global Logistics: Keynote speech at the CaiNiao Global Smart Logistics Conference". In: *CaiNiao Global Smart Logistics Conference* (May 2017).

[2] Varun Chandola, Arindam Banerjee, and Vipin Kumar. "Anomaly detection: A survey". In: *ACM Comput. Surv.* 41.3 (July 2009). ISSN: 0360-0300. DOI: 10.1145/1541880.1541882. URL: https://doi.org/10.1145/1541880.1541882.

[3] Victoria Hodge and Jim Austin. "A Survey of Outlier Detection Methodologies". In: *Artificial Intelligence Review* 22 (Oct. 2004), pp. 85–126. DOI: 10.1023/B:AIRE.0000045502.10941.a9.

[4] Malik Agyemang, Ken Barker, and Reda Alhajj. "A comprehensive survey of numeric and symbolic outlier mining techniques". In: *Intell. Data Anal.* 10 (Nov. 2006), pp. 521–538. DOI: 10.3233/IDA-2006-10604.

[5] Markos Markou and Sameer Singh. "Novelty detection: a review—part 1: statistical approaches". In: *Signal Processing* 83.12 (2003), pp. 2481–2497. ISSN: 0165-1684. DOI: https://doi.org/10.1016/j.sigpro.2003.07.018. URL: https://www.sciencedirect.com/science/article/pii/S0165168403002020.

[6] Animesh Patcha and Jung-Min Park. "An overview of anomaly detection techniques: Existing solutions and latest technological trends". In: *Computer Networks* 51.12 (2007), pp. 3448–3470. ISSN: 1389-1286. DOI: https://doi.org/10.1016/j.comnet.2007.02.001. URL: https://www.sciencedirect.com/science/article/pii/S138912860700062X.

[7] Raghavendra Chalapathy and Sanjay Chawla. *Deep Learning for Anomaly Detection: A Survey.* 2019. arXiv: 1901.03407 [cs.LG].

[8] C. De Stefano, C. Sansone, and M. Vento. "To reject or not to reject: that is the question-an answer in case of neural classifiers". In: *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 30.1 (2000), pp. 84–94. DOI: 10.1109/5326.827457.

[9] Daniel Barbará, Ningning Wu, and Sushil Jajodia. "Detecting Novel Network Intrusions Using Bayes Estimators". In: *Proceedings of the 2001 SIAM International Conference on Data Mining (SDM)*, pp. 1–17. DOI: 10.1137/1.9781611972719.28. eprint: https://epubs.siam.org/doi/pdf/10.1137/1.9781611972719.28. URL: https://epubs.siam.org/doi/abs/10.1137/1.9781611972719.28.

[10] Bernhard Schölkopf et al. "Estimating Support of a High-Dimensional Distribution". In: *Neural Computation* 13 (July 2001), pp. 1443–1471. DOI: 10.1162/089976601750264965.

[11] Volker Roth. "Outlier Detection with One-class Kernel Fisher Discriminants". In: *Advances in Neural Information Processing Systems.* Ed. by L. Saul, Y. Weiss, and L. Bottou. Vol. 17. MIT Press, 2004. URL: https://proceedings.neurips.cc/paper_files/paper/2004/file/1680e9fa7b4dd5d62ece800239bb53bd-Paper.pdf.

[12] Volker Roth. "Kernel Fisher Discriminants for Outlier Detection". In: *Neural computation* 18 (May 2006), pp. 942–60. DOI: 10.1162/089976606775774679.

[13] Karlton Sequeira and Mohammed Zaki. "ADMIT: anomaly-based data mining for intrusions". In: *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '02. Edmonton, Alberta, Canada: Association for Computing Machinery, 2002, pp. 386–395. ISBN: 158113567X. DOI: 10.1145/775047.775103. URL: https://doi.org/10.1145/775047.775103.

[14] Eleazar Eskin et al. "A Geometric Framework for Unsupervised Anomaly Detection: Detecting Intrusions in Unlabeled Data". In: *Applications of Data Mining in Computer Security* 6 (Feb. 2002). DOI: 10.1007/978-1-4615-0953-0_4.

[15] Ningning Wu and Jing Zhang. "Factor-analysis based anomaly detection and clustering". In: *Decision Support Systems* 42.1 (2006), pp. 375–389. ISSN: 0167-9236. DOI: https://doi.org/10.1016/j.dss.2005.01.005. URL: https://www.sciencedirect.com/science/article/pii/S0167923605000096.

[16] M. Otey et al. "Towards NIC-based intrusion detection". In: *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '03. Washington, D.C.: Association for Computing Machinery, 2003, pp. 723–728. ISBN: 1581137370. DOI: 10.1145/956750.956847. URL: https://doi.org/10.1145/956750.956847.

[17] Kalliopi Tsolaki et al. "Utilizing machine learning on freight transportation and logistics applications: A review". In: *ICT Express* 9 (Feb. 2022). DOI: 10.1016/j.icte.2022.02.001.

[18] Maria Riveiro, Giuliana Pallotta, and Michele Vespe. "Maritime anomaly detection: A review". In: *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 8 (May 2018), e1266. DOI: 10.1002/widm.1266.

[19] Brian Donovan and Dan Work. *New York City Taxi Trip Data (2010-2013)*. 2016. DOI: 10.13012/J8PN93H8. URL: https://doi.org/10.13012/J8PN93H8.

[20] Adam Kiersztyn et al. "The Concept of Detecting and Classifying Anomalies in Large Data Sets on a Basis of Information Granules". In: *2020 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*. 2020, pp. 1–7. DOI: 10.1109/FUZZ48607.2020.9177668.

[21] Paweł Karczmarek et al. "K-Means-based isolation forest". In: *Knowledge-Based Systems* 195 (2020), p. 105659. ISSN: 0950-7051. DOI: https://doi.org/10.1016/j.knosys.2020.105659. URL: https://www.sciencedirect.com/science/article/pii/S0950705120301064.

[22] Po-Ruey Lei. "A framework for anomaly detection in maritime trajectory behavior". In: *Knowledge and Information Systems* 47 (May 2015). DOI: 10.1007/s10115-015-0845-4.

[23] Claudio Di Ciccio et al. "Detecting Flight Trajectory Anomalies and Predicting Diversions in Freight Transportation (Extended Abstract)". In: Jan. 2016.

[24] *flightstates*. URL: https://www.flightstates.com (visited on 10/13/2023).

[25] *flightradar24*. URL: https://www.flightradar24.com (visited on 10/13/2023).

[26] Chung-Hao Lee and Yen-Fu Chen. "Anomaly Detection in Driving by Cluster Analysis Twice". In: *ArXiv* abs/2212.07691 (2022). URL: https://api.semanticscholar.org/CorpusID:254685909.

[27] Umang Bhatt et al. *Intelligent Pothole Detection and Road Condition Assessment.* 2017. arXiv: `1710.02595 [cs.CY]`.

[28] Adam Kiersztyn et al. “Detection and Classification of Anomalies in Large Data Sets on the Basis of Information Granules”. In: *IEEE Transactions on Fuzzy Systems* PP (Apr. 2021), pp. 1–1. DOI: `10.1109/TFUZZ.2021.3076265`.

[29] Sjoerd Van der Spoel, Chintan Amrit, and Jos Hillegersberg. “Predictive Analytics for Truck Arrival Time Estimation: A Field Study at a European Distribution Center”. In: *International Journal of Production Research* In Press (Jan. 2016), pp. 1–21. DOI: `10.1080/00207543.2015.1064183`.

[30] Rafael Duarte Alcoba and Kenneth W. Ohlund. “Predicting On-time Delivery in the Trucking Industry”. In: ().

[31] Annie George. “Anomaly Detection based on Machine Learning Dimensionality Reduction using PCA and Classification using SVM”. In: *International Journal of Computer Applications* 47 (June 2012), pp. 5–8. DOI: `10.5120/7470-0475`.

[32] Yao Ju, Jun Guo, and Shuchun Liu. “A Deep Learning Method Combined Sparse Autoencoder with SVM”. In: *2015 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery.* 2015, pp. 257–260. DOI: `10.1109/CyberC.2015.39`.

[33] Yuji Yamamoto. *Logistics service analysis.* 2022. URL: `https://www.kaggle.com/code/yujiyamamoto/logistics-service-analysis` (visited on 10/05/2023).

[34] Ram Thiagu. *Identifying effective parameters and optimal trip.* 2021. URL: `https://www.kaggle.com/code/ramakrishnanthiyagu/identifying-effective-parameters-and-optimal-trip` (visited on 10/05/2023).

[35] Mohammad Masdari. “Improving Security Using SVM-based Anomaly Detection: Issues and Challenges”. In: *Soft Computing* (Nov. 2020).

[36] Sarah M. Erfani et al. “High-dimensional and large-scale anomaly detection using a linear one-class SVM with deep learning”. In: *Pattern Recognition* 58 (2016), pp. 121–134. ISSN: 0031-3203. DOI: `https://doi.org/10.1016/j.patcog.2016.03.028`. URL: `https://www.sciencedirect.com/science/article/pii/S0031320316300267`.

[37] Abdallah Sebyala, Temitope Olukemi, and Dr Sacks. “Active Platform Security through Intrusion Detection Using Naive Bayesian Network For Anomaly Detection”. In: (Aug. 2002).

[38] Mayank Swarnkar and Neminath Hubballi. “OCPAD: One class Naive Bayes classifier for payload based anomaly detection”. In: *Expert Systems with Applications* 64 (2016), pp. 330–339. ISSN: 0957-4174. DOI: `https://doi.org/10.1016/j.eswa.2016.07.036`. URL: `https://www.sciencedirect.com/science/article/pii/S0957417416303839`.

[39] Rong-En Fan et al. “LIBLINEAR: A Library for Large Linear Classification”. In: *Journal of Machine Learning Research* 9.61 (2008), pp. 1871–1874. URL: `http://jmlr.org/papers/v9/fan08a.html`.

[40] Ciyou Zhu et al. “Algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound-constrained optimization”. In: *ACM Trans. Math. Softw.* 23.4 (Dec. 1997), pp. 550–560. ISSN: 0098-3500. DOI: 10.1145/279232.279236. URL: https://doi.org/10.1145/279232.279236.

[41] Sunil Aryal and Jonathan Wells. “Ensemble of Local Decision Trees for Anomaly Detection in Mixed Data”. In: Sept. 2021, pp. 687–702. ISBN: 978-3-030-86485-9. DOI: 10.1007/978-3-030-86486-6_42.

[42] Zhiruo Zhao, Kishan Mehrotra, and Chilukuri Mohan. “Online Anomaly Detection Using Random Forest”. In: Jan. 2018, pp. 135–147. ISBN: 978-3-319-92057-3. DOI: 10.1007/978-3-319-92058-0_13.

[43] Maryam Douiba et al. “Anomaly detection model based on gradient boosting and decision tree for IoT environments security”. In: *Journal of Reliable Intelligent Environments* 9 (July 2022). DOI: 10.1007/s40860-022-00184-3.

[44] Sumaiya Ikram et al. “Anomaly Detection Using XGBoost Ensemble of Deep Neural Network Models”. In: *Cybernetics and Information Technologies* 21 (Sept. 2021), pp. 175–188. DOI: 10.2478/cait-2021-0037.

[45] Michael S. Steinbach, George Karypis, and Vipin Kumar. “A Comparison of Document Clustering Techniques”. In: 2000. URL: https://api.semanticscholar.org/CorpusID:12808608.

[46] *Clustering Methods*. URL: https://scikit-learn.org/stable/modules/clustering.html.

[47] D. Sculley. “Web-scale k-means clustering”. In: *Proceedings of the 19th International Conference on World Wide Web*. WWW ’10. Raleigh, North Carolina, USA: Association for Computing Machinery, 2010, pp. 1177–1178. ISBN: 9781605587998. DOI: 10.1145/1772690.1772862. URL: https://doi.org/10.1145/1772690.1772862.

[48] Umberto Michelucci. “An Introduction to Autoencoders”. In: *CoRR* abs/2201.03898 (2022). arXiv: 2201.03898. URL: https://arxiv.org/abs/2201.03898.

[49] Xinyi Wang and Lang Tong. *Innovations Autoencoder and its Application in One-class Anomalous Sequence Detection*. 2021. arXiv: 2106.12382 [stat.ML].

[50] Kun Yang, Samory Kpotufe, and Nick Feamster. *An Efficient One-Class SVM for Anomaly Detection in the Internet of Things*. 2021. arXiv: 2104.11146 [cs.NI].

[51] Zhaomin Chen et al. “Autoencoder-based network anomaly detection”. In: *2018 Wireless Telecommunications Symposium (WTS)*. 2018, pp. 1–5. DOI: 10.1109/WTS.2018.8363930.

[52] D. Comaniciu and P. Meer. “Mean shift: a robust approach toward feature space analysis”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24.5 (2002), pp. 603–619. DOI: 10.1109/34.1000236.

[53] A. Famili et al. “Data preprocessing and intelligent data analysis”. In: *Intelligent Data Analysis* 1.1 (1997), pp. 3–23. ISSN: 1088-467X. DOI: https://doi.org/10.1016/S1088-467X(98)00007-9. URL: https://www.sciencedirect.com/science/article/pii/S1088467X98000079.

[54] *Ordinal Encoder*. URL: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.OrdinalEncoder.html (visited on 11/02/2023).

[55] *pandas.Dataframe.merge.* URL: https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.merge.html (visited on 10/02/2023).

[56] Mauricio Hernández and Salvatore Stolfo. "Real-world Data is Dirty: Data Cleansing and The Merge/Purge Problem". In: *Data Min. Knowl. Discov.* 2 (Jan. 1998), pp. 9–37. DOI: 10.1023/A:1009761603038.

[57] Qi Ding and Eric D. Kolaczyk. "A Compressed PCA Subspace Method for Anomaly Detection in High-Dimensional Data". In: *IEEE Transactions on Information Theory* 59.11 (2013), pp. 7419–7433. DOI: 10.1109/TIT.2013.2278017.

[58] Shimon Harush, Yair Meidan, and Asaf Shabtai. "DeepStream: Autoencoder-based stream temporal clustering and anomaly detection". In: *Computers Security* 106 (2021), p. 102276. ISSN: 0167-4048. DOI: https://doi.org/10.1016/j.cose.2021.102276. URL: https://www.sciencedirect.com/science/article/pii/S0167404821001000.

[59] Yunpeng Chang et al. "Clustering Driven Deep Autoencoder for Video Anomaly Detection". In: *Computer Vision – ECCV 2020.* Ed. by Andrea Vedaldi et al. Cham: Springer International Publishing, 2020, pp. 329–345. ISBN: 978-3-030-58555-6.

[60] Lawrence J. Hubert and Phipps Arabie. "Comparing partitions". In: *Journal of Classification* 2 (1985), pp. 193–218. URL: https://api.semanticscholar.org/CorpusID:189915041.

[61] Alexander Strehl and Joydeep Ghosh. "Cluster Ensembles - A Knowledge Reuse Framework for Combining Multiple Partitions". In: *Journal of Machine Learning Research* 3 (Jan. 2002), pp. 583–617. DOI: 10.1162/153244303321897735.

[62] Andrew Rosenberg and Julia Hirschberg. "V-Measure: A Conditional Entropy-Based External Cluster Evaluation Measure". In: *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL).* Ed. by Jason Eisner. Prague, Czech Republic: Association for Computational Linguistics, June 2007, pp. 410–420. URL: https://aclanthology.org/D07-1043.

[63] *Law of large numbers.* URL: https://https://en.wikipedia.org/wiki/Law_of_large_numbers (visited on 01/30/2024).

[64] Felipe L. Gewers et al. "Principal Component Analysis: A Natural Approach to Data Exploration". In: *ACM Computing Surveys* 54.4 (May 2021), pp. 1–34. ISSN: 1557-7341. DOI: 10.1145/3447755. URL: http://dx.doi.org/10.1145/3447755.

[65] Hongjing Zhang, Tianyang Zhan, and Ian Davidson. "A Self-Supervised Deep Learning Framework for Unsupervised Few-Shot Learning and Clustering". In: *Pattern Recognition Letters* 148 (2021), pp. 75–81. ISSN: 0167-8655. DOI: https://doi.org/10.1016/j.patrec.2021.05.004. URL: https://www.sciencedirect.com/science/article/pii/S0167865521001720.

[66] Junaid Haseeb et al. "Autoencoder-based feature construction for IoT attacks clustering". In: *Future Generation Computer Systems* 127 (2022), pp. 487–502. ISSN: 0167-739X. DOI: https://doi.org/10.1016/j.future.2021.09.025. URL: https://www.sciencedirect.com/science/article/pii/S0167739X2100371X.

[67] Pitoyo Hartono. "Mixing Autoencoder With Classifier: Conceptual Data Visualization". In: *IEEE Access* 8 (2020), pp. 105301–105310. DOI: 10.1109/ACCESS.2020.2999155.

[68] Yu and Shi. "Multiclass spectral clustering". In: *Proceedings Ninth IEEE International Conference on Computer Vision*. 2003, 313–319 vol.1. DOI: 10.1109/ICCV.2003.1238361.

[69] João Henriques et al. "Combining K-Means and XGBoost Models for Anomaly Detection Using Log Datasets". In: *Electronics* 9 (July 2020). DOI: 10.3390/electronics9071164.

[70] Bayu Adhi Tama and Kyung Hyune Rhee. "An in-depth experimental study of anomaly detection using gradient boosted machine". In: *Neural Computing and Applications* 31 (2017), pp. 955–965. URL: https://api.semanticscholar.org/CorpusID:254021853.

[71] Rifkie Primartha and Bayu Adhi Tama. "Anomaly detection using random forest: A performance revisited". In: *2017 International Conference on Data and Software Engineering (ICoDSE)*. 2017, pp. 1–6. DOI: 10.1109/ICODSE.2017.8285847.

[72] Mengjie Zhang. "The Application of Autoencoder in Classification of the Eye Movement Data". In: *PoS* ISCC2015 (2016), p. 001. DOI: 10.22323/1.264.0001.

[73] Max Gordon and Cranos Williams. "PVC Detection Using a Convolutional Autoencoder and Random Forest Classifier". In: Jan. 2019, pp. 42–53. DOI: 10.1142/9789813279827_0005.

[74] Baoan Zhang, Yanhua Yu, and Jie Li. "Network Intrusion Detection Based on Stacked Sparse Autoencoder and Binary Tree Ensemble Method". In: *2018 IEEE International Conference on Communications Workshops (ICC Workshops)*. 2018, pp. 1–6. DOI: 10.1109/ICCW.2018.8403759.

[75] Chunfeng Song et al. "Auto-encoder Based Data Clustering". In: *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*. Ed. by José Ruiz-Shulcloper and Gabriella Sanniti di Baja. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 117–124. ISBN: 978-3-642-41822-8.

[76] Chunfeng Song et al. "Deep auto-encoder based clustering". In: *Intelligent Data Analysis* 18 (Dec. 2014), S65–S76. DOI: 10.3233/IDA-140709.

[77] Hamid Hadipour et al. "Deep clustering of small molecules at large-scale via variational autoencoder embedding and K-means". In: *BMC Bioinformatics* 23 (Apr. 2022). DOI: 10.1186/s12859-022-04667-1.

[78] Mohamed Naim. "Evaluating the causes of uncertainty in logistics operations". In: *International Journal of Logistics Management, The* 21 (May 2010), pp. 45–64. DOI: 10.1108/09574091011042179.

[79] R. van Hoek, H. Commandeur, and G.C.J.M. Vos. "Reconfiguring logistics systems through postponement strategies". English. In: *Planning for virtual Response, Proceedings of the Annual Transportation and Logistics Educators Conference*. Unknown Publisher, 1996.

[80] Chia-Hsun Chang, Jingjing Xu, and Dong-Ping Song. *Risk analysis for container shipping: from a logistics perspective*. May 2015. DOI: 10.13140/RG.2.1.1717.9048.

[81] Ching-Chiao Yang and Yu-Kuo Chang. “Crucial factors influencing international logistics operations for African landlocked countries – A case study of Burkina Faso”. In: *Maritime Policy & Management* 46.8 (2019), pp. 939–956. DOI: 10.1080/03088839.2019.1606464. eprint: https://doi.org/10.1080/03088839.2019.1606464. URL: https://doi.org/10.1080/03088839.2019.1606464.

[82] Gerhard Münz, Sa Li, and Georg Carle. “Traffic anomaly detection using k-means clustering”. In: *Gi/itg workshop mmbnet*. Vol. 7. 9. 2007.

[83] Huan Niu et al. “An Ensemble of Locally Reliable Cluster Solutions”. In: *Applied Sciences* 10.5 (2020). ISSN: 2076-3417. DOI: 10.3390/app10051891. URL: https://www.mdpi.com/2076-3417/10/5/1891.

[84] Hamid Parvin and Behrouz Minaei. “A clustering ensemble framework based on selection of fuzzy weighted clusters in a locally adaptive clustering algorithm”. In: *Formal Pattern Analysis Applications* 18 (Feb. 2014). DOI: 10.1007/s10044-013-0364-4.

[85] “An ensemble clustering method for intrusion detection”. In: *Int. J. Intell. Eng. Inform.* 7.2–3 (Jan. 2019), pp. 112–140. ISSN: 1758-8715.

[86] Geping Yang et al. “RESKM: A General Framework to Accelerate Large-Scale Spectral Clustering”. In: *Pattern Recognition* 137 (2023), p. 109275. ISSN: 0031-3203. DOI: https://doi.org/10.1016/j.patcog.2022.109275. URL: https://www.sciencedirect.com/science/article/pii/S0031320322007543.

[87] Birol Erkan. “THE IMPORTANCE AND DETERMINANTS OF LOGISTICS PERFORMANCE OF SELECTED COUNTRIES”. In: *Journal of Emerging Issues in Economic, Finance and Banking* 3 (Jan. 2014), pp. 1237–1238.

A Feature importances on Delivery truck trip data

A.1 Before selecting Features

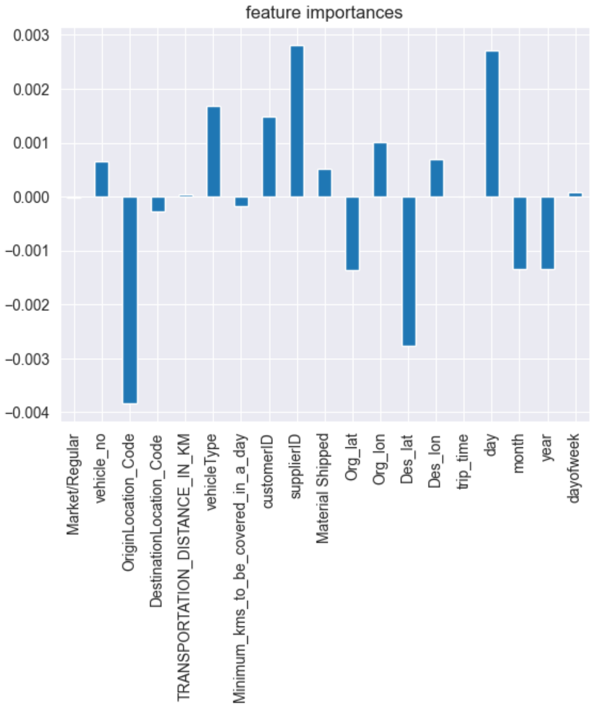

Figure 23: feature importances when using logistic regression before selecting important features

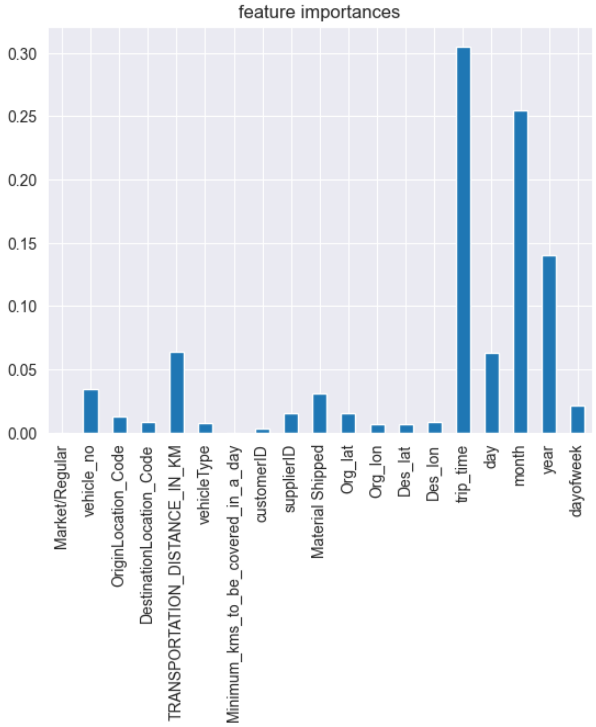

Figure 24: feature importances when using decision trees before selecting important features

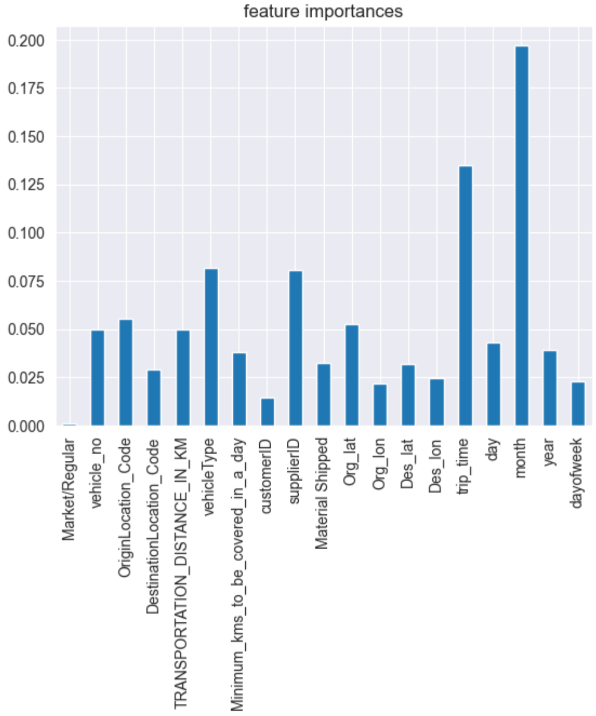

Figure 25: feature importances when using random forest before selecting important features

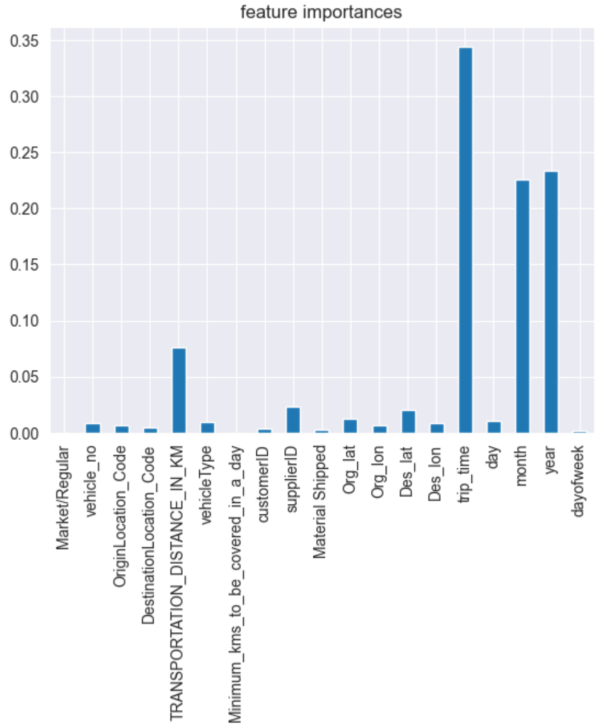

Figure 26: feature importances when using gradient boosted trees before selecting important features

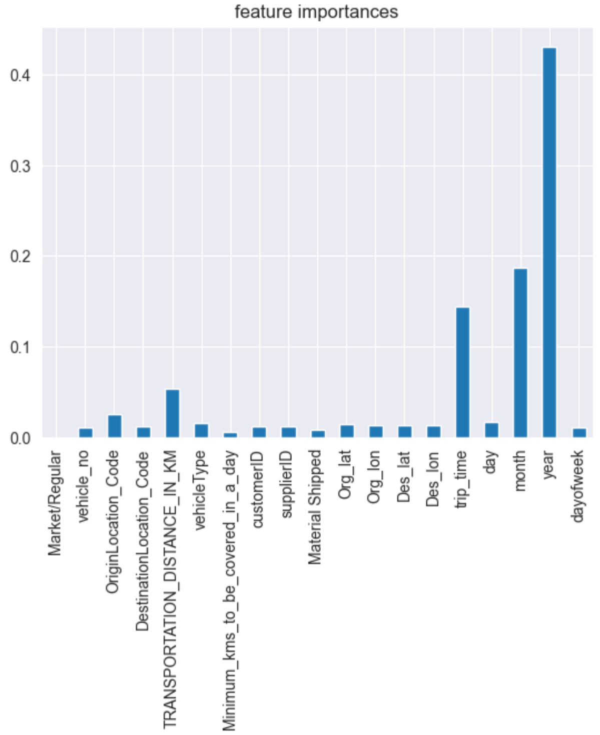

Figure 27: feature importances when using XGBoost before selecting important features

A.2 After selecting Features

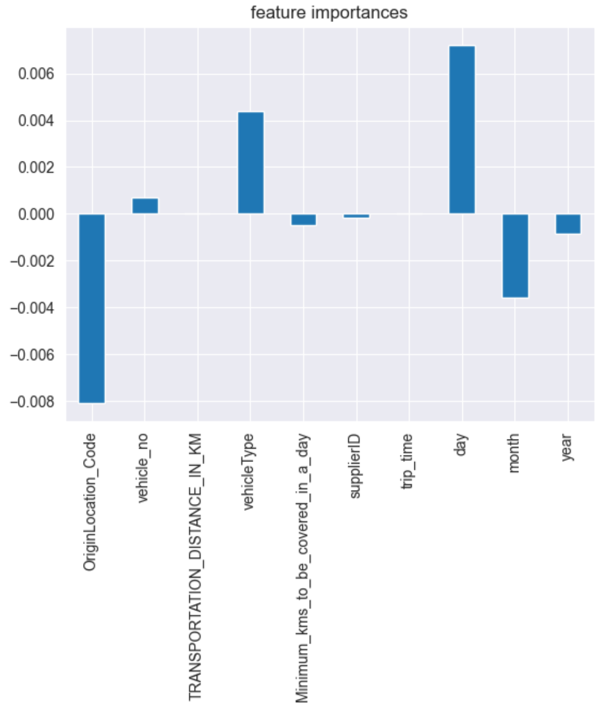

Figure 28: feature importances when using logistic regression after selecting important features

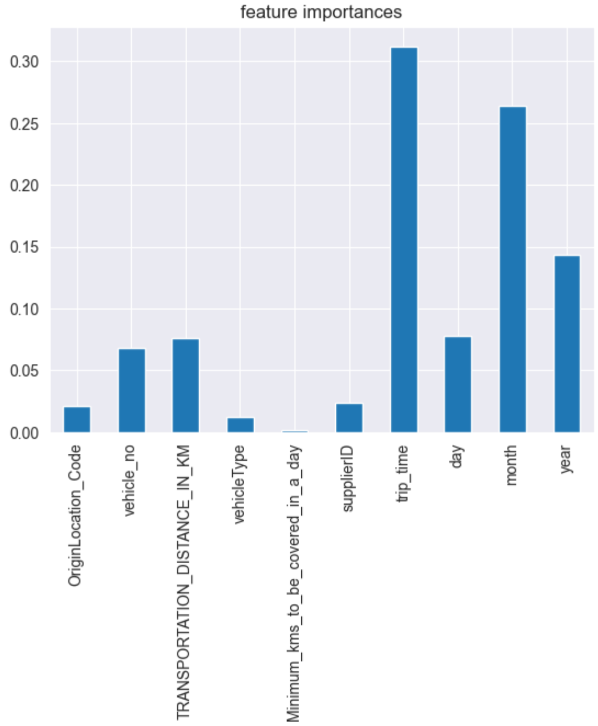

Figure 29: feature importances when using decision trees after selecting important features

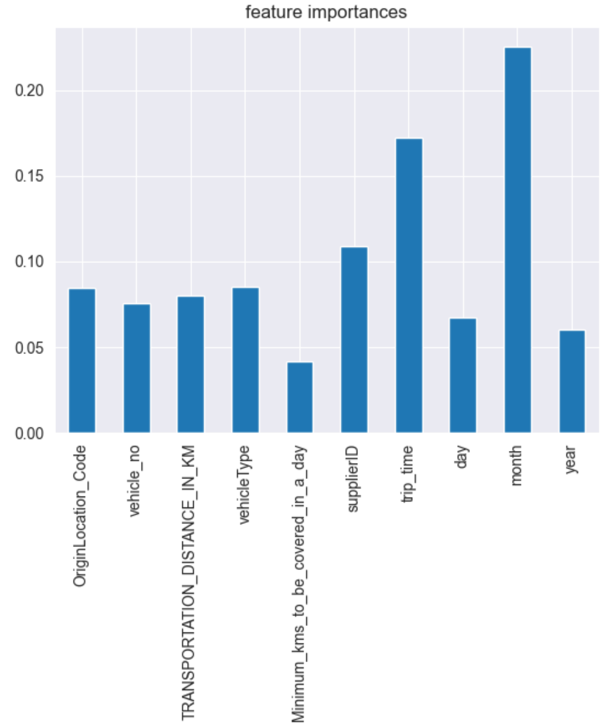

Figure 30: feature importances when using random forest after selecting important features

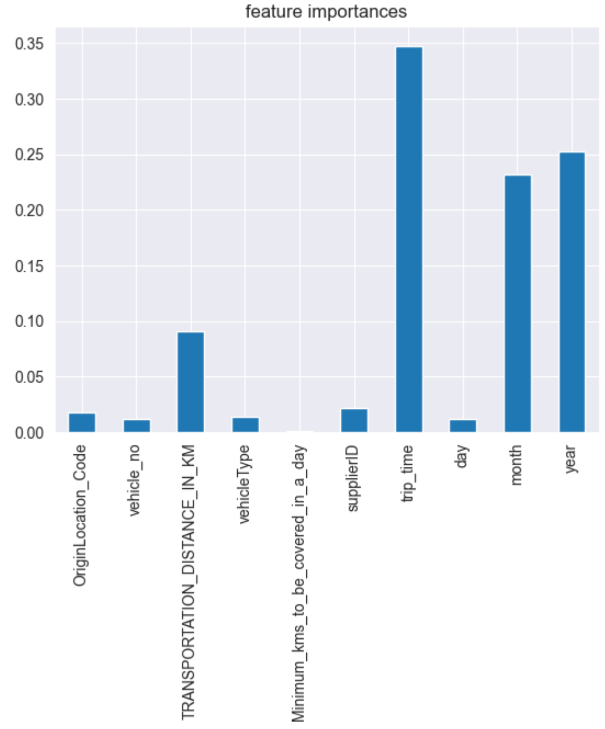

Figure 31: feature importances when using gradient boosted trees after selecting important features

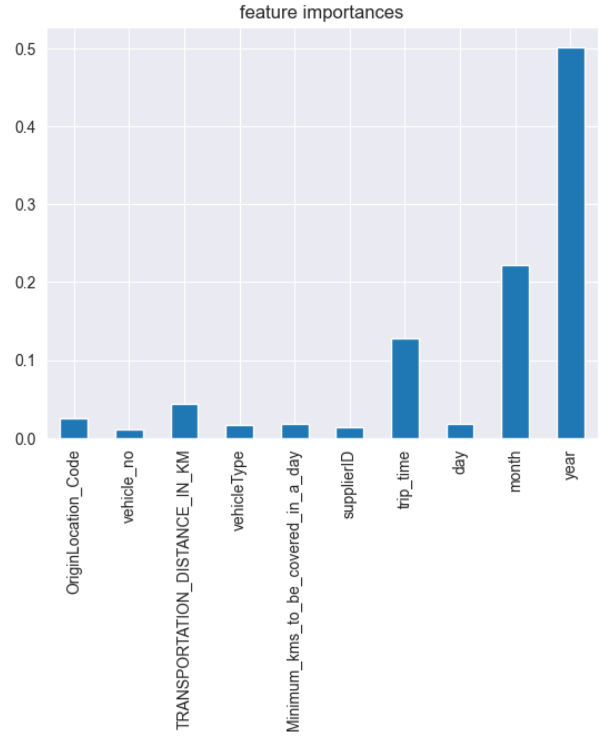

Figure 32: feature importances when using XGBoost after selecting important features

B Feature importances on Brazilian E-commerce public dataset

B.1 Before selecting Features

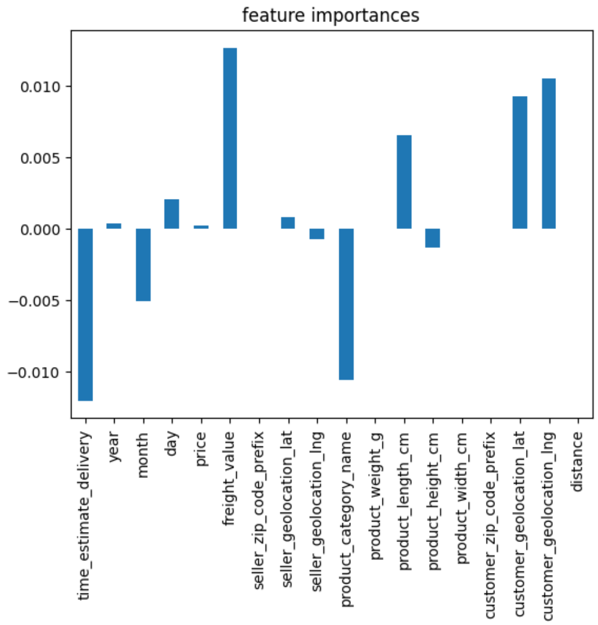

Figure 33: feature importances when using logistic regression before selecting important features

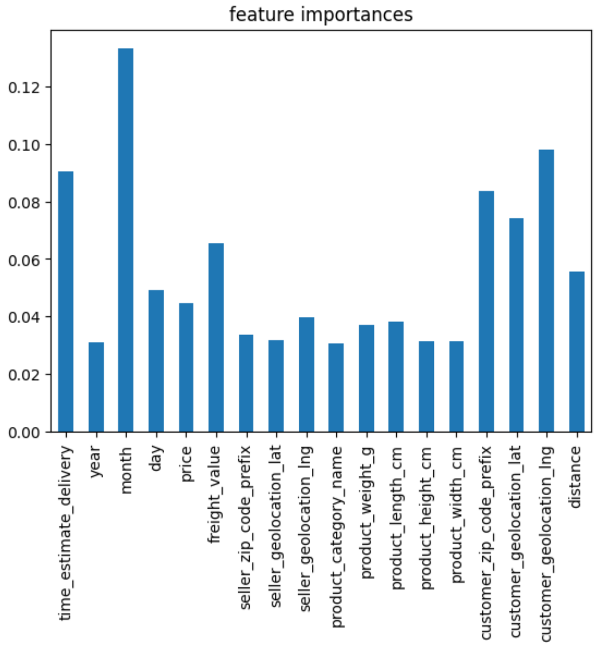

Figure 34: feature importances when using decision trees before selecting important features

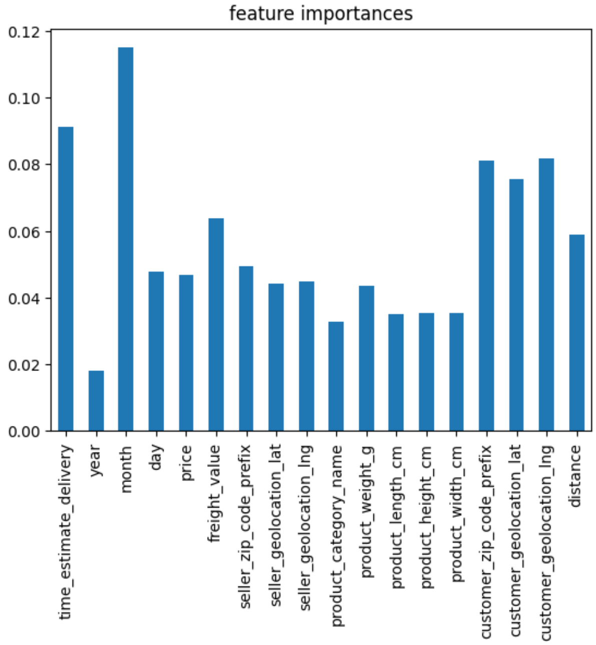

Figure 35: feature importances when using random forest before selecting important features

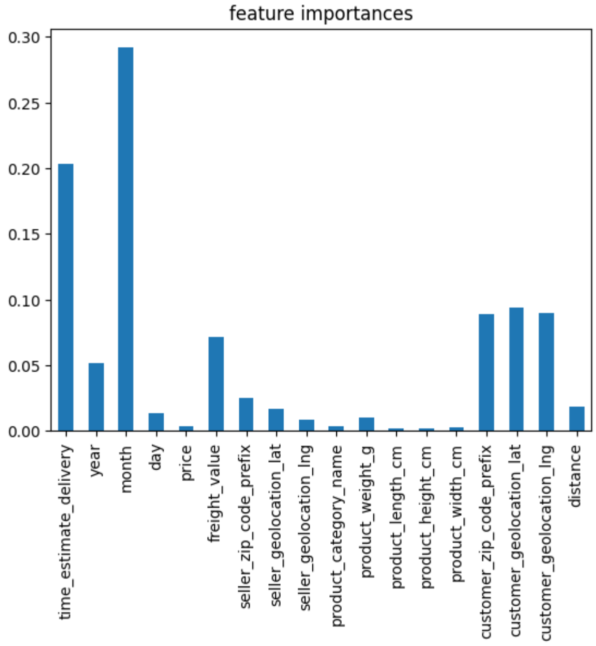

Figure 36: feature importances when using gradient boosted trees before selecting important features

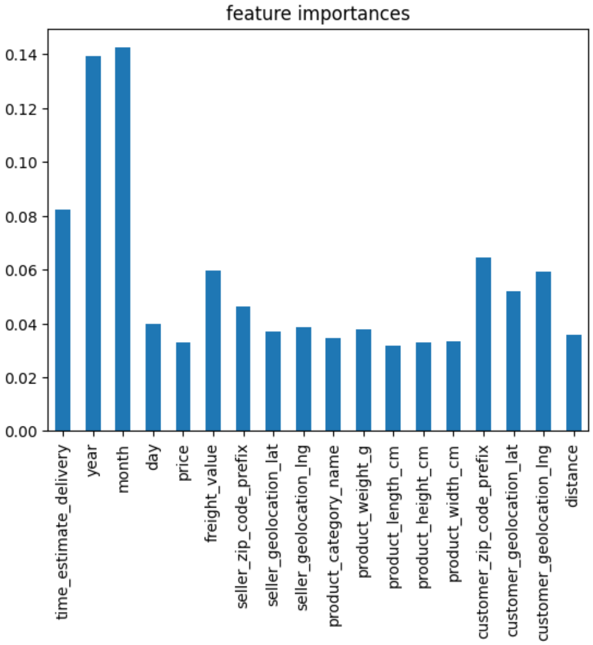

Figure 37: feature importances when using XGBoost before selecting important features

B.2 After selecting Features

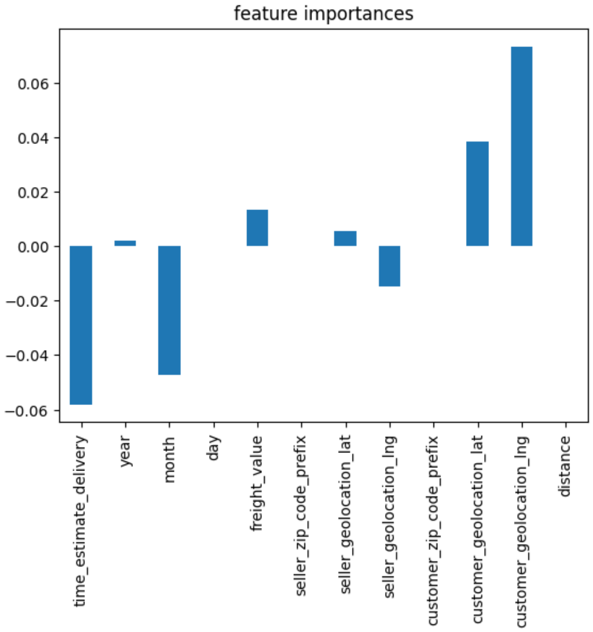

Figure 38: feature importances when using logistic regression after selecting important features

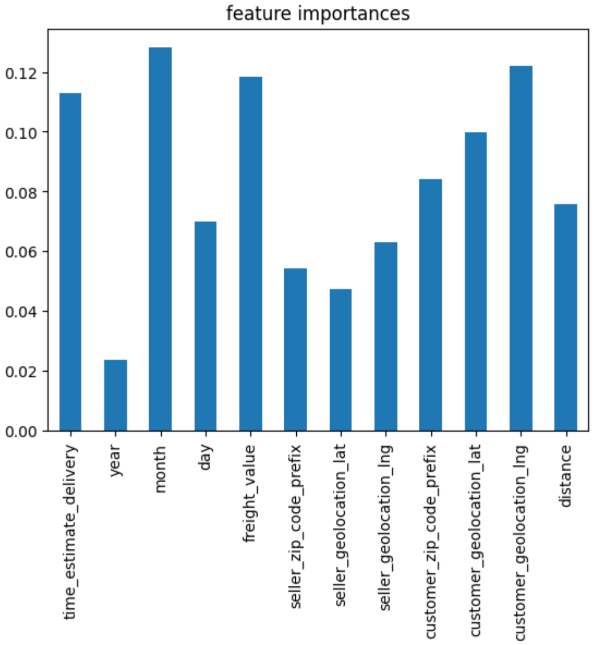

Figure 39: feature importances when using decision trees after selecting important features

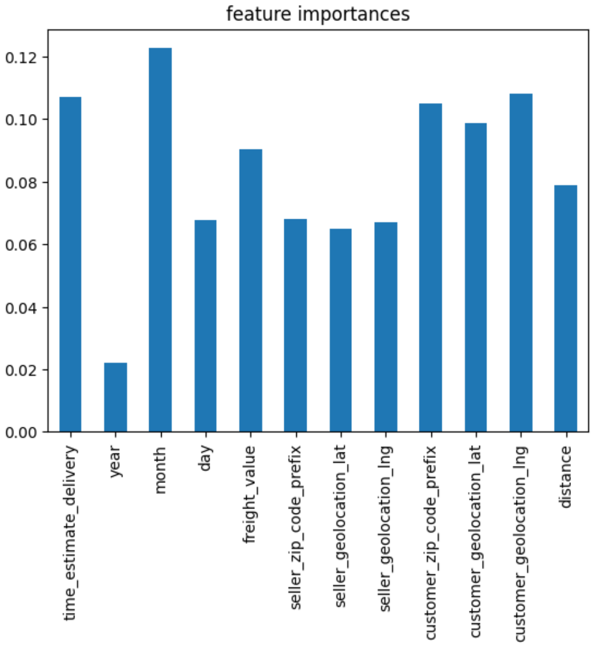

Figure 40: feature importances when using random forest after selecting important features

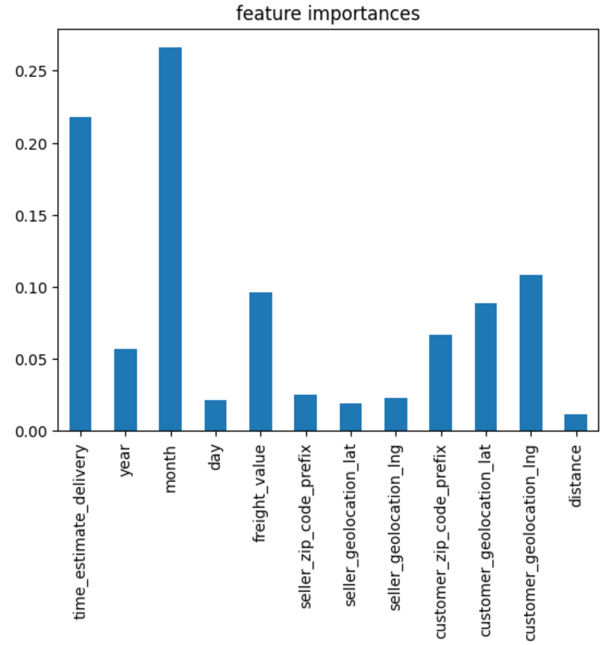

Figure 41: feature importances when using gradient boosted trees after selecting important features

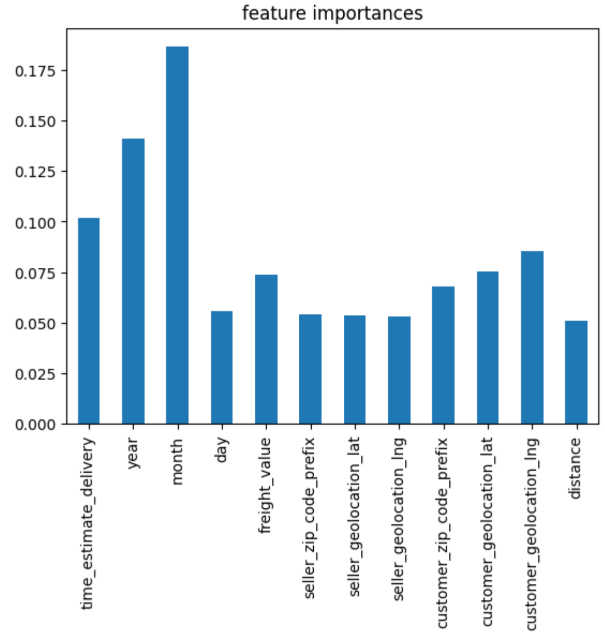

Figure 42: feature importances when using XGBoost after selecting important features