**Technische Hochschule Deggendorf**

**Faculty of Applied Computer Science**

Study Program Master Artificial Intelligence and Data Science

# FRAUD DETECTION USING MACHINE LEARNING

Master thesis to obtain the academic degree:

*Master of Science (M.Sc)*

at the Technical University of Deggendorf

**Submitted by:**

**Muhammad Saad Uddin**

**Matriculation number: 12100732**

**On: 29 October 2023**

**Supervisor:**

**Prof. Dr. A. Fischer**

**Additional Supervisor:**

**Simon Kußmann**

# Abstract:

This research is mainly focuses on the detection of fraud and anomalies in SAP ERP transactions using machine learning and artificial intelligence techniques. The objective is to understand the transactional data of SAP ERP systems, including the relationship between different tables, how the tables are connected, and the underlying distributions of the data. Based on this analysis, appropriate machine learning and deep learning techniques are applied to understand and learn the behaviour of transactions and spot anomalous and fraudulent transactions.

The methodology involves starting with exploratory data analysis (EDA) and visualization to explore the relationship between different features, as no labelled data is available. Decisions are made for high missing value features, and appropriate standardization and normalization techniques are used for continuous and categorical features. Finally, machine learning and deep learning techniques like anomaly detection algorithms, clustering algorithms, and autoencoders are applied and fine-tuned.

Key findings show that traditional ML algorithms like anomaly detectors and clustering fail mainly due to engineered features being taken into consideration for detecting outliers or making clusters instead of original features. The variety of features (dimensions) also compromises the performance/processing time of these algorithms as they fail to understand proper clusters or anomalous observations even without engineered features. Several features from different SAP tables have very high correlation. Autoencoders were able to successfully understand the transactional behaviour and distributions of features in thousands of dimensions and were able to detect anomalies by reconstructing possible correct values for anomalous fields.

In conclusion, deep learning autoencoder solutions worked best for data where there are hundreds of features. They can learn underlying high dimensional data distributions to detect anomaly and fraud in a much better way than traditional ML algorithms when labelled data is not available.

# Table of Contents:

# List of Figures and Tables:

# 1. Introduction

In this section the foundation for the study's framework and objectives are discussed. The motivation for the research and problem definition is explained.

## 1.1 Research Relevance and Motivation

Detecting fraud and anomalies in transaction data is crucial to prevent financial loss due to deliberate or accidental irregularities, such as invoice fraud, incorrect vendors, occupational fraud in companies, or ordering and accounting errors. Anomaly detection is a major topic in transaction data analysis and can be defined as identifying deviations from the normal or expected behavior. In simpler terms, Anomalies and Frauds are patterns in data that do not fit well to a well-defined distribution of normal behavior [22]. With the increasing digitization of our world, data-driven approaches, including machine learning and AI, can help detect anomalies with less manual effort. Various machine learning-based methods achieve this by learning a model of normality and distinguishing anomalies from it. However, accurately modelling normality in transactional data requires capturing distributions and dependencies within the data, with particular attention to numerical dependencies such as quantities, prices, or amounts.

In general, a transaction can also refer to an atomic action that changes data within a database and encompasses various domains. In our work, the focus is being made on transactional data as logs of business processes, including SAP data and invoices. However, our methods can also be applied to a broader understanding of transactions. For the purpose of our research, this definition will be kept. Since transaction data often contains combination of both categorical and numerical features, many machine learning methods require numerical representations to process the data. To address this, various representation learning techniques to transform categorical transaction data into dense numerical vectors or one-hot encoding were explored. This allows us to incorporate both numerical and categorical attributes into the computation of embeddings, later to investigate latent spaces and evaluate the quantitative performance of our methods for anomaly and fraud detection.

Anomaly and/or fraud detection is a crucial tool for many industries, including banks, insurance companies, sales, and businesses. The Association of Certified Fraud Examiners (ACFE) defines estimates that companies currently lose 5% of their revenue to this type of fraud [11]. By quickly identifying and addressing outlier behavior, companies can prevent financial and client loss. As such, automatic anomaly detection is essential for medium and large enterprises. If a method could identify anomalies in large volumes of data and provide feedback on their root cause, many companies would benefit significantly. Despite the importance of this task, it remains unsolved. There are many algorithms available, but they all have their drawbacks, such as poor scalability to larger data sets, the need for prior knowledge about the input data or pre-defined hyperparameters [28] that are often chosen heuristically. While it may be unlikely to find a universal method that can solve any clustering or classification task, continued research into algorithms and their combinations can bring us closer to finding multipurpose solutions.

Anomaly detection and its application present several challenges that need to be addressed. One of the most significant challenges is the rarity of anomalies in practice. While not explicitly required, anomalies are generally expected to be less frequent than non-anomalies, as otherwise, they would not comply with the definition of normality or normal behavior. In practice, this class imbalance is often present in anomaly detection, particularly in transaction data. For example, the European Central Bank estimated that out of 100.75 billion card transactions in 2019, only 24.16 million were fraudulent, corresponding to a rare 0.024 percent of transactions [29]. This rarity highlights the fact that anomalies cannot be precisely defined for many applications, especially as the complexity of systems increases. The number of aspects that can contribute to non-normality grows with the number of dimensions and anomalies resulting from malicious intent are often being tried to intentionally be obscured or concealed [12]

## 1.2 Problem Definition and Objectives

One of the greatest challenges for academic research in anomaly detection is the availability of labeled data. This challenge can be partially addressed through unsupervised or one-class approaches that characterize normal or unknown data with an acceptable low contamination rate. Another approach, known as modeling normality or modeling normal of data, involves training a deep learning model to learn these characteristics from a large set of high dimensional data, possibly sparse and potentially unlabeled data [7]. This implicitly circumvents the challenges of rarity and the lack of universal definitions of anomalies. Modeling and learning data distribution without any available labels is a promising approach for the SAP transaction data domain and is studied extensively in this thesis.

To reduce revenue loss due to occupational fraud, researchers have suggested using data from SAP Enterprise Resource Planning (ERP) systems to detect fraudulent activity [9, 10]. SAP ERP systems are essential for managing the flow of cash, materials, production, and other resources within companies. They represent a large market and support the daily operations of most medium-sized and large companies. Previous research on SAP ERP system fraud detection can be divided into approaches that rely on entirely private data and frauds, private data with synthetically injected frauds, or entirely synthetic data and frauds. While some studies have used real SAP ERP system data to develop and evaluate fraud detection systems [7, 9, 10], details about the data and the data itself are kept confidential to protect company trade secrets and privacy information.

The increasing volumes of transaction data pose a challenge for people responsible for detecting anomalies or fraud. Due to the large amount of data, only a small percentage (less than one percent) can be randomly sampled for manual assessment. This creates a sampling risk, as financial misstatements may go undetected in the larger, non-sampled data group [13]. Humans have limited ability to evaluate each case of misstatement, especially when evolving fraudulent patterns are unknown. This can lead to failures in detecting intentional or unintentional financial misstatements. Missing the detection of financial misstatements, particularly fraudulent ones, carries significant risks. As such, the importance of accurately detecting data deviations cannot be underestimated [14].

Much current research is working on the question to find balance between use of data sampling, bagging, boosting or cost-sensitive analysis and they arrive at different conclusion

mainly due to the diverse data as each case requires a different approach then another one. Keeping this in mind, our goal is to find a solution for anomaly and fraud detection within industrial processes of SAP ERP. The question pertains to how this problem would differ from others in the industry. It is evident that in financial transactions, the amount of fraud is relatively low, but understanding its uniqueness and, more importantly, the training of a model to detect it, is the challenge. For problem-solving, several steps will be ensured, from exploratory data analysis, data distributions and relationships, and sampling to understanding model outcomes to reveal the fraud's statistical nature in this industry. A novel approach will be taken in this regard. Formulation will be based on a set of rules defined within the SAP ERP environment and domain knowledge from expert users. A useful dataset will be formulated from several databases and different document type distributions, making relations to different transactions. Then, unsupervised machine learning techniques, namely anomaly, clustering, and deep learning (autoencoders), will be employed to annotate the dataset for fraudulent behaviour of an SAP-based transaction.

## 1.3 Structure of this Thesis

The thesis is structured into several chapters, with each serving a distinct purpose and contributing to a holistic understanding of the research topic. In Chapter 1, the Introduction Is set by presenting the research's significance, motivation, problem definition, and objectives. Chapter 2, the Literature Review, delves into existing knowledge and relevant studies that form the foundation of this work. Chapter 3, Theoretical Background, introduces the essential theory, including SAP ERP, relevant tables, and various approaches such as rule-based methods, unsupervised learning, and neural networks. The Methodology, detailed in Chapter 4, explains the practical aspects, covering data creation, pipeline development, exploratory data analysis, feature engineering, imputation, selection, standardization, model development, and parameter tuning. Chapter 5, Implementation, discusses the hands-on aspects of data collection, organization, analysis, data wrangling, processing, feature extraction, modelling, and fine-tuning. Chapter 6, Experiment Results, presents the outcomes and insights gained from the experiments, followed by Chapter 7, Discussion, where the results are analysed and conclusions are drawn. In Chapter 8, Conclusion, key findings, and contributions are summarized, and avenues for future research are suggested. Lastly, Chapter 9, References, lists all the sources and references that have been cited.

# 2. Literature Review

Fraud and anomaly detection in financial data is a challenging problem that has been addressed by various research employing machine learning and AI techniques. However, most of the existing research has some limitations that make them less relevant or applicable to our work.

Some studies rely solely on labelled data, which is scarce and expensive to obtain in our context. Other research focuses only on rule-based methods [4] that only cover a specific subset of SAP transactions i.e., not working to detect anomaly on whole transaction cycle and use synthetic data instead of using real world dataset, so the scope was quite limited compared to volume and variety of real word data. Some other research explored Invoice data employing Feature engineering and selection with unsupervised anomaly detection approaches [3] but they also have access to labelled data from the finance ministry to validate their results, which deviates from our scenario. A similar study on fraud and anomaly detection in accounting data using cluster-based multivariate and histogram-based outlier detection methods. They applied their methods to a real-world dataset. They performed K-means clustering on different transaction types separately to obtain better clusters, and they used histograms to identify outliers. They also added synthetic anomalies to the data to test their methods. They set a specific threshold for the anomaly score to filter out the anomalies. However, some of the detected anomalies were not relevant for financial audit purposes, which shows the limitations of their methods. They concluded that clustering was a suitable technique for this kind of analysis [15].

Moreover, LIC Tree-LSTM (Local Intention Calibrated Tree-LSTM) or commonly called Behavior Tree was proposed, this LIC Tree-LSTM leverages the utilization of behavior trees to effectively detect fraudulent transactions. By integrating the behavior tree structure into the LSTM architecture, this method can capture complex temporal patterns inherent in fraud activities [16]. Similarly, another tree based approach namely Density Estimation Trees (DETs) was also introduced [1][17]. DETs offer a notable advantage of remarkably fast prediction times. DETs exhibit an interpretable nature, facilitating the generation of a set of decision rules that contribute to higher anomalousness scores. The combination of flexibility, efficiency, and interpretability in DETs presents a promising direction for improving the accuracy and understanding of anomaly detection models. Some light was also shed on one class classification and their use in learning deep features and possible application in big data [18][19] which could be possibly utilized for detecting anomalous data as well.

Another research leveraged the existing literature on SAP ERP systems to identify and formulate rules (red flags) that indicate potential fraud or anomaly [5]. They then applied a process mining technique to extract the relevant information from the data and compare it with the rules. However, this approach has two major drawbacks: first, the data they used was synthetically generated, which means that it may not represent the real-world situations and challenges that is faced in data used in this research. Second, the rules they defined were limited to a narrow range of SAP transactions, which ignores the possibility of fraud or anomaly occurring in other parts of the system or across different transactions.

Similarly, another research explored the use of deep neural networks (DNN) for fraud and anomaly detection in SAP ERP data [2]. They designed a shallow DNN architecture and trained it on one table of the SAP ERP system. They also introduced synthetic anomalies in the data to

evaluate their model. However, this approach suffers from several limitations: first, the shallow DNN architecture may not be able to capture the complex and nonlinear relationships among the features and the anomalies. Second, the focus on only one table of the SAP ERP system may not account for the interactions and dependencies among different tables and transactions that may affect the occurrence and detection of fraud and anomaly. Third, synthetic anomalies may not reflect the true nature and distribution of fraud and anomaly in real-world data.

One promising approach that was found in the literature was the use of deep autoencoder networks [7] for fraud and anomaly detection in SAP ERP data. They employed a deep autoencoder network and trained it on two tables of the SAP ERP system. They used real-world data to train and test their model and achieved good results. However, this approach also has some limitations: first, they restricted their scope to only two tables of the SAP ERP system, which may not cover all the possible sources and types of fraud and anomaly in the data. Second, they selected only a few features from each table to feed into their model, which may not capture all the relevant information and variations in the data. Another research focuses on deep autoencoder neural network [6] to find anomalies in the real-world dataset taken from the SAP ERP system for one legal entity and one fiscal year. They focused on the per-account level and selected only three types of accounts: Revenue Domestic, Revenue Foreign and Expenses. The authors concluded that reconstruction errors can be a very important metric in finding anomalies in the real-world dataset. Similarly, [8] used a variational autoencoder to detect anomalies in the data taken from the Synesis ERP system. They only used categorical features to train the model. They did not have any labeled data, so they assessed the model performance based only on the reconstruction error features of the model same as [6].

In contrast, our approach aims to improve upon the previous work by using a deep autoencoder network with hyperparameter tuning and expanding the scope to the complete SAP transactional cycle. Reliance is not on any labelled data or predefined rules, but rather on an unsupervised learning method that can learn from the data itself. Real-world data is utilized for the training and testing of the model, and the complete transaction cycle of the SAP ERP system is included in the analysis. It is believed that this approach can overcome the limitations of the existing methods and achieve better performance and generalization for fraud and anomaly detection in SAP ERP data.

# 3. Theoretical Background

This chapter highlights the foundational theories and concepts that form the basis of our research on anomaly and fraud detection within SAP transactional data. The theoretical frameworks of machine learning algorithms, unsupervised learning, one-class classification, and deep learning are explored, as they play a pivotal role in developing effective strategies for anomaly and fraud detection. Through a comprehensive examination of these theoretical underpinnings, readers will acquire a robust understanding of the methodologies and principles driving the advancement of our research.

## 3.1 SAP ERP

SAP ERP is a software that belongs to the category of Enterprise Resource Planning (ERP). SAP ERP connects all the core functions needed to run a business, such as finance, human resources, manufacturing, logistics, services, procurement, and more. It helps coordinate all these functions within a unified system.

An ERP system can be referred to as the “central nervous system of a business” because it provides the automation, integration, and intelligence necessary for effectively managing all daily business activities. Most or all business data should be in the ERP system to have a single data source for the whole business. Therefore, an ERP system is crucial for both large and small and medium-sized businesses (SMEs).

## 3.2 SAP tables used

A range of tables from SAP ERP environment are employed to create a complete end to end transactional cycle, the details of each table are as follows:

### 3.2.1 BSEG

BSEG is a table that stores the line items for accounting documents in the SAP ERP system. Accounting documents are records of the financial activities and transactions of your organization, such as sales, purchases, payments, etc. Table BKPF is another table that holds the header lines for accounting documents, which contain information such as document number, document type, posting date, etc.

### 3.2.2 RBKP

RBKP is a table in the SAP application that belongs to the Invoice Verification module. It contains the document header data for invoice receipts, which are documents that confirm the receipt of goods or services and the payment details.

### 3.2.3 RSEG

RSEG is a table in the SAP application that is part of the Invoice Verification module. It stores detailed data for document items for incoming invoices, which are documents that record the goods or services received from a vendor and the payment terms.

### 3.2.4 EKKO

EKKO is a table in the SAP application that belongs to the Purchasing module. It contains the purchasing document header data, which is the data that describes the general information and conditions of a purchasing document, such as purchase order, contract, etc.

### 3.2.5 EKPO

EKPO is a table in the SAP application that is part of the Purchasing module. It contains the detailed purchasing document item data, which is the data that describes the details and specifications of each item in a purchasing document, such as material number, quantity, price, delivery date, etc.

### 3.2.6 BKPF

BKPF is a table in SAP R/3 ERP systems that is part of the accounting module. It holds the header lines for accounting documents, which are the lines that contain the general and administrative information of an accounting document, such as company code, document number, fiscal year, etc. These fields are also the key fields that uniquely identify an accounting document in the table.

## 3.3 Rule based approach:

The rule-based approach is a method for fraud and anomaly detection that involves creating rules or processes based on previously known cases of fraud and anomaly. The rules can also be derived from the experience and knowledge of domain experts who have insights into the patterns and behaviors of fraudsters and anomalies. The rule-based approach has some benefits, like the rules are easy to understand and implement, as they have clear and explicit criteria for identifying fraud and anomaly. The rules can be updated or added as new patterns emerge, which allows for flexibility and adaptability to changing situations. The rules can be applied quickly and efficiently to the data, as they have low computational requirements compared to some machine learning models that may need more time and resources to train and test.

One of the main reasons to have rule-based approach here is to collect data that can be used as ground truth to validate unsupervised learning techniques.

## 3.4 Un-Supervised Learning

Unsupervised learning is a branch of machine learning that does not depend on human supervision or labels to learn from data. Instead, it uses artificial intelligence algorithms to uncover hidden patterns, structures, or features in the data by itself. Unsupervised learning can be regarded as a form of self-learning, where the machine tries to understand the data without any prior knowledge or instructions.

In unsupervised learning, the data is unlabeled, which means that there is no output or answer associated with each data point. For example, if given to the machine a collection of images without telling it what they are, the machine has no clue what to look for or what to predict. Instead, the machine has to find its own way of organizing or grouping the data based on some criteria or similarity. For example, the machine might cluster the images based on their colors, shapes, textures, or other features that it can detect. The machine does not know what these clusters mean or represent, but it can create them based on the data itself.

### 3.4.1 Clustering

Clustering is a technique of grouping data points based on their similarity or distance. For example, imagine you have a lot of colored brushes, and you want to sort them into different baskets. How would you do it? You might use some criteria such as color, shape, or size, to decide which brush belongs to which basket. For example, you might put all the red brushes in one basket, all the green brushes in another basket, and so on. This is essentially what clustering does: it finds a way to divide the data into meaningful groups or clusters.

Clustering can be useful for many purposes, such as: Clustering can help you discover the structure and distribution of your data and identify the main characteristics and differences of each cluster. It can help you detect patterns or trends in your data that may not be obvious or visible otherwise or Clustering can help you reduce the complexity and dimensionality of your data by representing it with a smaller number of clusters.

#### 3.4.1.1 K-means

K-means is a popular and simple algorithm for clustering data into groups based on their similarity or distance. The main idea of k-means is to find the best way to partition the data into k clusters, where k is a number chosen by the user. The algorithm works as follows: The user decides how many clusters they want to have, this can be based on some prior knowledge, domain expertise, or trial and error. The algorithm randomly picks k data points from the data set and assigns them as the initial centroids of the clusters. The algorithm calculates the distance between each data point and each centroid using some distance measure, such as Euclidean distance and assigns each data point to the closest centroid based on some distance measure. In the next iteration, it recalculates the centroids as the mean of all the data points in each cluster. These steps are repeated until the centroids do not change significantly, or a maximum number of iterations is reached.

#### 3.4.1.2 DBSCAN

DBSCAN is a clustering algorithm that groups data points based on their density. Density refers to how close the data points are to each other. DBSCAN can also detect outliers or noise points that do not fit into any group. DBSCAN works by setting two parameters: epsilon and minimum points. Epsilon is the maximum distance between two data points to consider them as neighbors. Minimum points are the minimum number of neighbors a data point needs to be a core point.

### 3.4.2 Anomaly Detection

Anomaly detection is a process that uses machine learning to find data points, events, or observations that are different from the normal or expected behavior of the data. For example, if you have a sensor that measures the temperature of a machine, and it suddenly shows a very high or low value, that could be an anomaly. Anomaly detection can help you identify problems, errors, defects, frauds, or other unusual situations in your data.

#### 3.4.2.1 Isolation Forest

Isolation forest is a method of finding outliers or abnormal data points in a dataset. Outliers are data points that are very different from the rest of the data and may indicate some problems or errors. For example, if you have a dataset of people's heights, and you find someone who is 3 meters tall, that would be an outlier. The idea behind isolation forest is that outliers are easier to separate from the rest of the data than normal data points. This is because outliers are usually far away from the majority of the data and have different values or characteristics. Therefore, it takes fewer random splits to isolate an outlier than a normal data point [27].

#### 3.4.2.2 Local Outlier Factor (LOF)

Local outlier factor (LOF) is another method of finding outliers or abnormal data points in a dataset. LOF works by comparing the density of each data point with its neighbors. Density refers to how close the data points are to each other. LOF uses two concepts to measure density: reachability distance and local reachability density.

#### 3.4.2.3 Elliptic Envelope

Elliptic envelope is a method of finding outliers or abnormal data points by fitting an ellipse around the data points that are considered normal. An ellipse is a shape that resembles a circle that has been stretched. It has two axes: a major axis and a minor axis. The major axis is the longest distance across the ellipse, and the minor axis is the shortest distance across the ellipse. The center of the ellipse is the point where the major axis crosses.

Elliptic envelope tries to find the smallest ellipse that covers most of the data points, while excluding the outliers. It does this by using a parameter called contamination, which is the fraction of outliers in the dataset. The user must specify the contamination value, which can range from 0 to 0.5. For example, if contamination is 0.15, it means that 15% of the data points are outliers.

### 3.4.3 Neural Networks

Neural networks are a type of unsupervised learning that can learn from data and perform various tasks. Neural networks are modeled after and resemble the structure and function of the human brain. Neural networks consist of layers of artificial neurons that are connected by weights. Each neuron receives some input from the previous layer, performs some computation, and produces some output to the next layer. The weights determine how much each input influences the output of each neuron. The first layer of a neural network is the input layer, which takes the data as input. The last layer of a neural network is the output layer, which produces the outcome or prediction. The layers in between are the hidden layers, which process and transform the data.

#### 3.4.3.1 Autoencoders

Autoencoders are a type of neural network that can learn to copy or replicate input data. They consist of two main parts: an encoder and a decoder.

Encoder takes the input data and compresses it into a smaller representation, also called a latent space. It tries to capture the most important features of the input data. The decoder takes the compressed representation (latent space) from the encoder and tries to reconstruct the original input data from it[25].

The idea behind autoencoders is to reduce the input data into a more compact and meaningful representation and then recreate the input data as closely as possible using that representation. This process is called "autoencoding.". Autoencoders can be used for various types of data, not just images [23]. They are widely used in unsupervised learning tasks, anomaly detection, and other data-related applications.

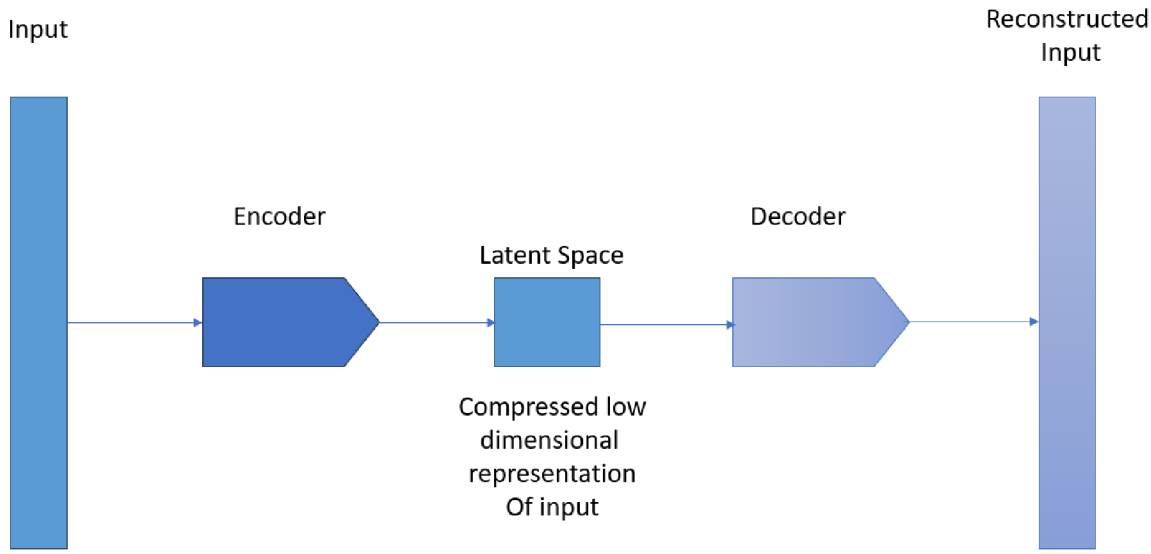



Figure 3.1: Concept of Autoencoders

### 3.4.3.2 Adam Optimizer

Adam (Adaptive Moment Estimation) is an optimization algorithm used in training neural networks. It combines the concepts of both momentum and adaptive learning rates to efficiently update the model's parameters during the training process. The adaptive learning rate in Adam helps handle different scales of gradients for different parameters, making it well-suited for optimizing models with sparse gradients or noisy data.

### 3.4.3.3 AdamW Optimizer

AdamW is a variation of the Adam optimizer, which addresses a potential limitation of the original Adam algorithm related to weight decay regularization [21]. Weight decay is a common regularization technique used to prevent overfitting in neural networks by adding a penalty term to the loss function based on the magnitude of the model's weights. The main difference between Adam and AdamW lies in the way they handle weight decay.

### 3.4.3.4 SGD Optimizer

Stochastic Gradient Descent is an optimization algorithm commonly used in training neural networks. SGD with momentum which is a variation of the standard SGD optimizer will be used, that helps accelerate the learning process and improve the efficiency of training neural networks. It addresses some of the limitations of the basic SGD by incorporating the concept of momentum.

### 3.4.3.5 Dropout

Dropout layer is a regularization technique used to prevent overfitting. Dropout is designed to improve the generalization ability of neural networks by randomly dropping out a fraction of neurons during training. By randomly deactivating neurons during training, dropout helps prevent complex co-adaptations of neurons. This encourages the network to learn more robust and generalized features that are less sensitive to specific inputs.

### 3.4.3.6 Leaky ReLU

is a variant of the standard ReLU activation function and is designed to address some of the issues that ReLU may have when a function becomes negative i.e., the output is set to zero, resulting in deactivating the neuron. The problem with this is that once a neuron becomes inactive (outputting zero), it may stay that way during training and never recover.

Leaky ReLU solves this issue by allowing a small, non-zero gradient for negative inputs, which means that the neuron remains active even for negative input values [20]. The Leaky ReLU function is defined as:

$$f(x) = \max(ax, x)$$

where **f(x)** is the output of the activation, function **x** is the input to the function, and **a** is a small positive constant that determines the slope of the function for negative inputs.

The introduction of the positive constant ensures that the neuron is not fully deactivated for negative inputs, making Leaky ReLU more robust and less prone to dying neurons during training. It allows the neuron to continue learning from negative input values and helps prevent the saturation of neurons, especially in deep neural networks.

## 3.5 One class classifier

One-class classifiers are a type of machine learning model that learns to recognize and identify instances of a specific class of data, where the data mostly belongs to that single class. These classifiers are trained on only one class of data and are designed to detect anomalies or outliers, which are data points that differ significantly from most of the training data. One-class classifiers are helpful in situations where you have a scarcity of data for anomalies, or you want to focus on detecting specific types of outliers. They are commonly used in anomaly detection, fraud detection, intrusion detection in cybersecurity, fault detection in manufacturing processes, and many other applications where identifying rare and unusual events is crucial.

### 3.5.1 One Class Support Vector Machine

One-class SVM (Support Vector Machine) is a type of machine learning algorithm used for one-class classification, particularly in anomaly detection. It is a variation of the traditional SVM, which is mainly used for binary classification tasks. The goal of a one-class SVM is to learn and create a boundary around the normal or majority class data points in such a way that it includes as many of those normal data points as possible while excluding anomalies or outliers. It tries to find the best boundary that encapsulates the majority of the data, considering it as the "normal" region, and anything outside that boundary is considered an anomaly.

One-class SVM is well-suited for anomaly detection because it can handle imbalance data effectively, it only requires a normal class data for training, and it works well with high dimensional data [26].

## 3.6 Curse of Dimensionality

"Curse of dimensionality" is a term used in machine learning to describe the challenges that arise when working with data in high-dimensional spaces. It refers to the fact that as the number of dimensions (features) in a dataset increases, the data points become sparser and more spread out. This sparsity and increased distance between data points can cause problems in data analysis and modeling. As the dimensionality increases, the volume of the space grows exponentially, resulting in an enormous number of data points required to adequately cover the space. This sparsity leads to several issues like increased computational time or more importantly distance lose meaning in higher dimension due to sparsity.

## 3.7 Evaluation metric

Reconstruction loss serves as the primary evaluation metric in our study to assess the performance of the autoencoder model. The reconstruction loss measures the dissimilarity between the original input data and the data reconstructed by the autoencoder. Specifically, as the autoencoder aims to encode the input data into a lower-dimensional representation and then decode it back to its original form, the reconstruction loss quantifies the discrepancy between the reconstructed data and the original data. A lower reconstruction loss indicates that the autoencoder is more successful in accurately replicating the input data, implying that it has effectively captured the essential features and patterns present in the data. Moreover, the use of reconstruction loss facilitates model comparison and hyperparameter tuning, as it provides a quantitative measure of the autoencoder's performance across various configurations.

For validating the performance of the autoencoder model, adoption is made for a data validation approach that involves querying data points identified as known anomalies based on pre-defined rules. Specifically, a set of predefined rules or criteria to determine anomalous instances in the dataset is established. These rules are derived from prior domain knowledge and expert. By adhering to these rules, a subset of data samples that are pre-labeled as anomalous will be obtained. The use of known anomalies for validation purposes allows us to conduct a rigorous evaluation of the autoencoder's anomaly detection performance. It

provides a controlled and objective assessment of the model's capability to discern between normal and anomalous instances, thus contributing to the robustness and reliability of the evaluation process.

# 4. Methodology

This chapter outlines the methodology employed to execute our research on anomaly and fraud detection within SAP transactional data. The step-by-step approach to data preprocessing, feature engineering, model selection, and evaluation is detailed. The utilization of various machine learning techniques, including autoencoders, clustering algorithms, and one-class SVM, is elucidated within the context of the defined problem. By navigating through this chapter, the systematic process is explained through which our research objectives are translated into actionable methodologies for effective solution.

## 4.1 Creation of transactional data

This research approach involves comprehensively exploring the data available in the SAP system, which ensures that no potential source of information is overlooked through collaboration with domain experts from financial department of Endress+Hauser who possess extensive knowledge of SAP landscape and specializes in understanding of financial transaction and invoicing in SAP. This collaborative effort allows us to gain valuable insights into the underlying structure and content of the data including how to define a complete end to end SAP transaction by utilizing multiple tables and which SAP tables and field are relevant from financial transactions. Specifically, the investigation of various tables such as BSEG, BKPF, RSEG, RBKP, EKPO, EKKO and CDPOS are being made, which collectively form a fundamental part of the SAP database. By understanding the purpose and significance of each table, it is learned how they interrelate and can be effectively combined to construct a steady transaction flow that encompasses the entire end-to-end cycle.

### SQL connection

Subsequently, our focus shifts to establishing a SQL connector that enables seamless access to the SAP database through remote connections. This SQL connector serves as a vital link between our analytical environment and the SAP system, facilitating the retrieval and interaction with the underlying data. By implementing this connector, the data retrieval process was streamlined, paving the way for efficient querying, processing, analysis, and exportation of information. The SQL connector is designed to offer a robust and secure means of establishing communication with the SAP database, standard OBDC connection can also be established but the organization doesn't allow direct query to the sensitive data that's why custom developed connector is utilized. Leveraging remote access capabilities, it ensures that our analytical platform can connect to the SAP system from an external environment. This remote connectivity is carefully configured and governed by security protocols such as SSL encryption for connections to SAP databases, ensuring data integrity and privacy are upheld throughout the data interaction process.

## 4.2 Data pipeline creation

In the next phase creation of data pipeline is established, which serves as a systematic and organized framework for data processing, transformation, and analysis.

### Data extraction and cleaning

The data extraction process involves retrieving relevant data from SAP database landscape. A data extraction pipeline is established which will be discussed in detail in next chapter. This pipeline was imperative because it work as foundation for providing in format which was clearer and concise for next step in this research. Once the data is extracted, it goes through the cleaning process. Data cleaning is crucial as it ensures the quality of the data used in analysis. This involves identifying and correcting or removing errors, inconsistencies, and inaccuracies that occur in the data. It may include tasks like removing duplicates, handling missing values, and correcting inconsistent data entries.

The cleaned data is then prepared for the next stages of the pipeline, which may include data analysis, model building, and interpretation of results. By ensuring the data is clean and well-structured, we can improve the accuracy and reliability of our findings in the subsequent stages of the data pipeline.

### Rule based data extraction.

Here, rules are defined based on the invaluable expertise and domain knowledge provided by domain experts. These rules include finding anomalies by combining a set of SAP tables in process like verifying invoice numbers systematically, multiple invoices for same order as shown in figure 4.1 where it was systematically taken into account SAP tables namely BSEG, BKPF. Similarly, other rules-based data extraction process was also performed which includes creation of rules as payment to invoices which were blocked and invoices not matching posting orders, unintentional financial changes which is originated out of system, changes in invoice during payment runs, incomplete documents where purchase orders, sales orders or mandatory information are empty and vendors which are not qualified are providing new invoices.

By leveraging the insights and knowledge of these domain experts, access is gained to invaluable information that helps shape the rules and criteria for data extraction. Through this collaborative process, different scenarios that could potentially indicate fraud or anomalies within the data are carefully identified. The expertise of domain experts ensures that the rules encompass a comprehensive range of potential fraudulent or anomalous scenarios, capturing both common and rare occurrences that may be present in the data.

- Rule 2: Double invoice check (same/similar)

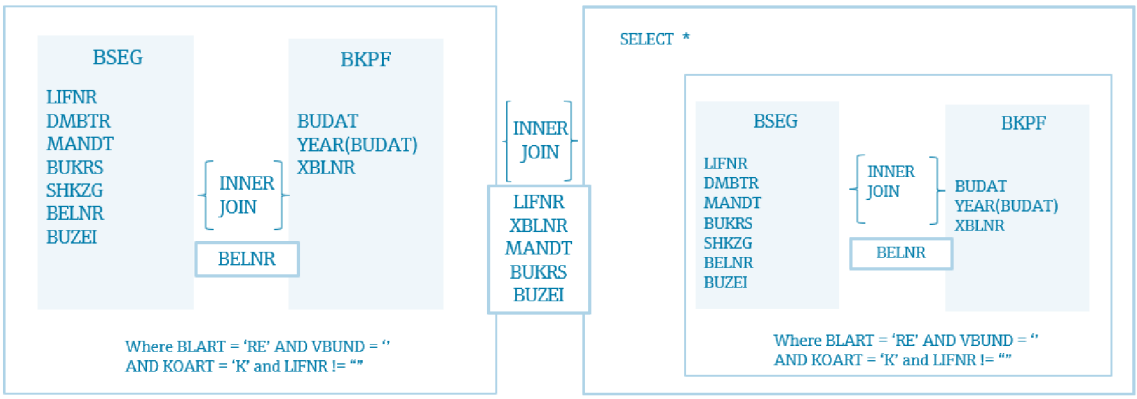



Figure 4.1: Example of a rule-based data extraction

### Data quality

Following the data extraction, significant emphasis is placed on examining the data quality before proceeding with any analyses. This involves checking if there are any missing essential information, comparing the data with reliable reference tailored for the organization and accuracy where knowledge about possible valid values for given field were leveraged, and examining the data's coherence and uniformity for consistency. This crucial step aims to assess the reliability, consistency, and completeness of the extracted data to ensure that subsequent analyses are based on accurate and dependable information. The completeness of the data is assessed, ensuring that essential information is not missing. This is achieved by checking and verifying all the necessary fields including invoice numbering format and sequence, vendor numbers, addresses, product type, plant, currency, and release date. The accuracy of the data is evaluated by comparing it with reliable reference sources and expert knowledge. The sources include datasets and transaction format already defined and used as principal guideline for any entry in SAP. The consistency of the data is examined, checking for coherence and uniformity across various attributes and data points. Lastly, duplicate entries within the dataset are checked for to eliminate redundancy and ensure the uniqueness of data points.

## 4.3 EDA

Before diving into the exciting world of AI modelling, a closer look at the data is needed to explore its characteristics and patterns. This is what Exploratory Data Analysis (EDA) is all about. It helps uncover the hidden patterns in the data and make smart choices about how to use them for modelling. This includes catering missing values, understanding numeric and categorical features, visualizing, and interpreting statistical significance like correlations and covariance and understanding the underlying data distributions.

## 4.4 Feature Engineering

Feature engineering is undertaken, a crucial step involving the creation of new features from the original ones to enhance the dataset's predictive power and uncover deeper insights. Feature engineering is a critical aspect of data preprocessing, aimed at extracting meaningful information from existing features while eliminating irrelevant or redundant ones. Thirty two features are created from existing ones. For instance, new features are derived from date-related columns by extracting useful information such as the month, day, or day of the week from the original dates. Summary statistics (e.g., mean, standard deviation, maximum, minimum) are computed for original features, providing a more compact representation of the data while retaining important information.

## 4.5 Imputation

In the next step to cater the issue of missing values, imputation was performed. imputation was applied because it plays a pivotal role in our data analysis and preprocessing efforts. Imputation, in essence, is an effective statistical methodology employed to address the issue of missing or incomplete data, a common challenge encountered in data analysis. This technique operates by substituting the absent values within a dataset with estimated or predicted values, thoughtfully derived from the existing data points. In simpler terms, imputation functions as a data enhancement process, effectively filling in the gaps to ensure that our dataset remains reliable and conducive to meaningful analysis and machine learning tasks.

## 4.6 Feature selection

Feature selection was performed, a critical step utilized across various methods, with a specific focus on autoencoders. Especially for autoencoders, which learn from the data distribution, feature selection plays a substantial role in improving the model's performance and interpretability.

During the exploratory data analysis step, certain patterns and characteristics in the features were identified that guided the feature selection process. Specifically, focus was placed on three key aspects. Attention was paid to features that exhibited a high proportion of missing values and those that demonstrated a strong correlation with other features. High missing values could lead to biased or skewed model training, while high correlation between features might introduce multicollinearity, impacting the stability and interpretability of the autoencoder model. Features were identified that had an excessive number of unique values, making it impractical to derive meaningful data distribution or probabilities for each unique value. For instance, certain categorical features with thousands of unique values might lack significant distributional patterns, rendering them less informative for model learning.

## 4.7 Standardization

In this segment, the focal point of attention was the standardization of the dataset, a critical prerequisite to ready it for integration into the autoencoder model. This standardization process unfolded as a series of pivotal steps, carefully orchestrated to craft the data into a format that would seamlessly accommodate effective model training and facilitate comprehensible interpretations. To address the categorical features embedded within the dataset, the methodology of one-hot encoding was judiciously employed. One-hot encoding performs a remarkable transformation by rendering categorical variables as binary vectors. This transformation crystallizes the categorical nuances within the data, ensuring that no information is inadvertently omitted.

## 4.8 Select modelling technique.

After preparing the datasets, the next step was to select appropriate machine learning techniques for experimentation. Considering the problem's nature, which involves anomaly and fraud detection in the absence of labeled data, three distinct approaches were chosen, each offering its own advantages.

From the anomaly detection algorithms, the isolation forest algorithm was chosen due to its efficiency in detecting anomalies in large datasets. The Local Outlier Factor (LOF) was chosen for its ability to identify local deviations or density-based outliers within the dataset. The elliptic envelope algorithm was selected as it models the data distribution with an elliptical shape, making it effective in detecting multivariate outliers. From the clustering algorithms, the K-means algorithm was chosen as it groups data points into distinct clusters based on similarity. While K-means is not specifically designed for anomaly detection, its clustering capability can be leveraged to identify outliers as data points distant from the cluster centroids. The K-Nearest Neighbors (KNN) was also incorporated as a clustering technique, relying on distance-based similarity to group data points. Similar to K-means, KNN can also be used for anomaly detection by considering data points that are distant from their neighboring points. Autoencoders were selected as the deep learning approach for anomaly detection. Autoencoders are neural network architectures specifically designed for unsupervised learning tasks like anomaly detection.

The decision to use multiple approaches was made due to their unique strengths and adaptability to different types of anomalies. Additionally, due to the lack of labeled data, these unsupervised techniques proved valuable in identifying anomalies without requiring labeled examples.

## 4.9 Build models

While the anomaly and clustering models were prebuilt and used as is, there was a need to build the autoencoder model from the ground up to meet specific requirements. Autoencoders are a type of neural network architecture that learns to reconstruct input data in a compressed representation. For this, the encoder and decoder parts of the autoencoder were carefully designed.

The autoencoder architecture consists of two main components: the encoder and the decoder. The encoder is responsible for transforming the input data into a lower-dimensional latent space, while the decoder aims to reconstruct the original input data from this compressed representation. It is crucial to ensure that the encoder and decoder have the same number of dimensions to enable an accurate reconstruction of the input data.

## 4.10 Tuning Parameters

For the autoencoder model, an extensive hyperparameter tuning process was conducted. The process started by selecting different optimizers, including Adam, AdamW, and SGD with momentum. Each optimizer has its unique strengths and adapts differently to the data distribution, making it crucial to explore various options.

# 5. Implementation

This chapter provides an in-depth exploration of how the selected methods are applied in practice. Firstly, it includes how each method is set up and the particular parameters that are chosen for each one. This not only involves explaining the technical details, but also the reasons behind these choices, and how they contribute to the overall goals of the research.

## 5.1 Data Gathering and Organization

To collect and organize the right data, an active engagement with subject matter experts were made, those possessing extensive knowledge and expertise in SAP systems and business processes. This engagement involves weekly and bi-weekly status calls to discuss progress, address any challenges, and clarify uncertainties. These regular interactions serve as an opportunity to ensure alignment on goals, share insights, and strategize on the best ways to optimize the data collection process. Their valuable domain-specific insights enable us to unravel the underlying semantics of the data and its relevance to various aspects of the business workflow such as the meanings of specific SAP terms, the significance of different data fields, and the implications of certain data patterns. This understanding is crucial in accurately interpreting the data and making meaningful inferences like for instance, a particular data trend may signify a specific business process or an accounting practice, which could be misunderstood without the proper expertise. The collaborative effort ensures accurate interpretation of the data in the context of business operations, accounting practices, and relevant financial transactions. A comprehensive exploration of the tables is undertaken, analyzing their individual attributes, and understanding the relationships between them. This includes examining the primary keys, foreign keys, and other linkages that govern how the tables are interconnected, thus providing a clear understanding of data relationships. By integrating this knowledge, a coherent representation of the end-to-end transaction cycle that spans across different tables is established, effectively capturing the complete flow of business activities.

This process is critical in laying the foundation for subsequent analyses and insights generation. It serves as the bedrock upon which further data processing and modeling steps including data cleaning, feature engineering and model testing and evaluation were build, enabling this research to unlock valuable patterns, anomalies, and trends present within the SAP data. Moreover, this collaborative approach of domain expertise with data exploration, ensures that our subsequent data-driven analyses align closely with real-world business operations, bolstering the credibility and applicability of our research findings.

Next step was to align SQL pipeline for data organization. Within the SQL connector, SAP plugin were incorporated, which plays a pivotal role in creating virtual tables. These virtual tables function as an intermediary data layer, residing within the analytical environment while mirroring essential data from the SAP database. Having established these virtual tables, data can be seamlessly queried, processed, analyzed, and exported within the analytical environment without directly impacting the SAP database. This ensures that analytical activities, including complex queries, data manipulations, and advanced analytics, do not impose any undue strain on the SAP system's performance. Moreover, this isolation of data

through virtual tables provides a safeguard against inadvertent modifications or disruptions to the SAP database, safeguarding its integrity and stability.

Data extraction is proceeded from the SAP database utilizing the SQL connector. However, during this extraction process, certain data format issues within the database are encountered. Specifically, more than 20 numeric fields are marked as strings by default, this likely is part of database design, while other data fields lack standardization or have been designated as strings despite their numeric nature. Additionally, float fields are encountered that are excessively long, potentially leading to precision and storage challenges. To address these data format discrepancies, the focus intensifies on data extraction and cleaning. This critical step aims to rectify the encountered issues to ensure that the extracted data adheres to correct and consistent data formats, eliminating any potential for corrupt or erroneous data.

The extraction process involves converting the incorrectly labeled numeric fields, represented as strings, back to their original numeric formats. This process comprises the resolution of various issues. Firstly, the inconsistency of non-numeric values, which are converted to 'NaN'. Secondly, the standardization of numeric representations to address variations in comma and decimal point usage. Lastly, it involves addressing the issue of special characters or extraneous spaces that may be attached as prefixes or suffixes to these numeric values. This rectification ensures that these numeric values are interpreted and treated appropriately, allowing for accurate mathematical calculations and meaningful analyses. Moreover, standardization issues such as date formats or categorical variables are identified and resolved into a consistent format where were compatible with international Organization for Standardization (ISO). Standardization enhances the uniformity of the data and enables seamless data integration across different parts of the research workflow.

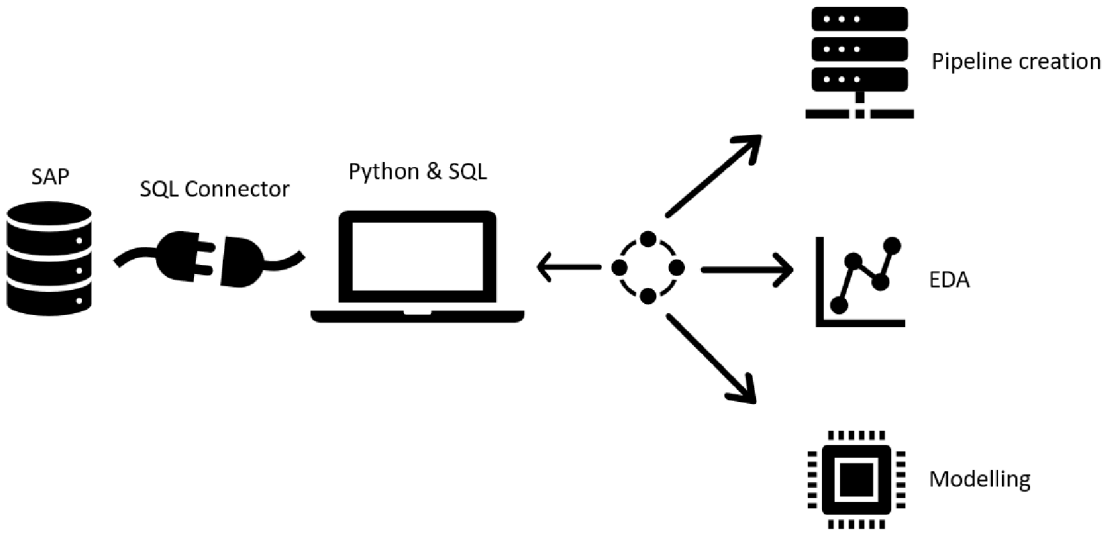



Figure 5.1: Overview of Pipeline creation

Figure 5.1 simplify the various stages in this research workflow implementation from data extraction to modeling, providing a streamlined and unified view of the various steps involved.

The process commences with data extraction from the SAP system using SQL connector which is then subjected to a data cleaning process. The subsequent step involves the creation of a pipeline, a sequence of data processing elements, to automate the data processing tasks. Exploratory Data Analysis is conducted to gain insights into the trends, patterns, and relationships in the data. This is followed by the development of models.

## 5.2 Data Analysis & Exploration

Data analysis and exploration is a critical step in any data pipeline. This phase involves scrutinizing the gathered data to identify patterns, trends, and insights that can inform decision-making. It's during this phase that we delve into the intricate details of the data and uncover its hidden truths.

### Missing Values

One of the things that requires attention when exploring data is the presence of missing values. Missing values can signal poor data quality or incomplete data collection. They can also impact the performance of AI models if not addressed appropriately. Thus, it is necessary to identify the features that have a significant number of missing values and decide whether to retain them or eliminate them from the data. In this case, the features with more than 10% missing values were identified and dropped, as imputation might result in a change of dimensions for the original data.

### Continuous Features

This section shed light about understanding the relationship between continuous features within the context of the research. The exploration of how these continuous features interact and influence the data's characteristics is thoroughly examined.

### Correlation and Covariance

A vital task that needs to be accomplished is understanding the relation between different features, how they are related to each other, and how they affect each other. The simplest way to achieve this is to measure the correlation and covariance of the features. Correlation tells how strongly two features are linked or related to each other, while covariance portrays how much they change or vary together.

### Grouping

By computing these metrics for each feature, they can be classified into different groups based on their values. For example, different features may have a strong positive correlation but a low positive covariance, meaning that they move in the same direction but not by much. Meanwhile, others may have a weak negative correlation but a high negative covariance, meaning that they move in opposite directions and by a large margin. This information can be presented in a table for easy reference.

In this case, features with strong correlations or covariance have been filtered out and grouped into different categories. These categories range from features with very high correlation combined with high, medium, or low covariance, to features with moderate correlation

combined with similar levels of high, medium, or low covariance. This helps us understand the nature of features in our dataset and helps us understand the driving force and relation behind certain features like for example, as shown in figure 5.2 there is a strong positive correlation between features bukrs_ekko and altkt_bseg with a correlation value of 0.803. This means that as the values of bukrs_ekko increase, the values of altkt_bseg also tend to increase. The covariance value of 177022.01 indicates that these two variables vary together to a large degree.

Similarly, again from figure 5.2 there is a perfect positive correlation between bukrs_ekko and kokrs_bseg with a correlation value of 1.0, indicating that these two variables have a strong linear relationship. The covariance value of 612.11 indicates that these two variables vary together, but to a lesser degree than bukrs_ekko and altkt_bseg.

| | Column 1 | Column 2 | Correlation | Covariance |
|---|---|---|---|---|
| 1 | bukrs_ekko | bukrs_ekpo | 1.000000 | 612.11 |
| 2 | bukrs_ekko | bukrs_rseg | 1.000000 | 612.11 |
| 3 | bukrs_ekko | bukrs_rbkp | 1.000000 | 612.11 |
| 4 | bukrs_ekko | bukrs_bkpf | 1.000000 | 612.11 |
| 5 | bukrs_ekko | bukrs_bseg | 1.000000 | 612.11 |
| ... | ... | ... | ... | ... |
| 264 | rebzj_bseg | stjah_rbkp | 0.705873 | 14120.07 |
| 266 | rebzj_bseg | rebzz_bseg | 1.000000 | 7.00 |
| 267 | rebzz_bseg | stjah_rbkp | 0.705873 | 6.98 |
| 268 | rebzz_bseg | rebzj_bseg | 1.000000 | 7.00 |
| 270 | qbshb_bseg | qsshb_bseg | 0.807047 | 56598615.85 |

Figure 5.2: Statistical relation between continuous features

It is also evident from figure 5.2 that there is also a very strong positive correlation between ekorg_ekko and werks_ekpo with a correlation value of 0.999. This means that as the values of ekorg_ekko increase, the values of werks_ekpo also tend to increase. The covariance value of 1915035.47 indicates that these two variables vary together to a very large degree.

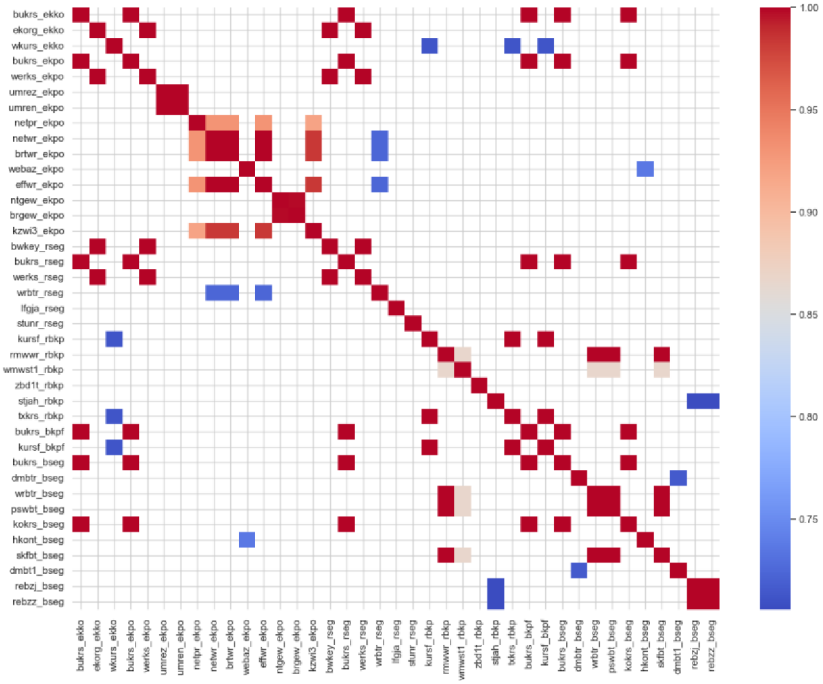



Figure 5.3: Correlation Heatmap for Highly correlated features

Correlation heatmap in figure 5.3 visually explains all the features having strong to very strong correlations between features which was essential for identifying potential multicollinearity issues in our data, guiding feature selection and aiding in the development of an effective predictive model.

## Distributions

In addition to measuring correlation and covariance, the distribution of the features also needs to be inspected. The distribution shows how the values of a feature are dispersed or concentrated. In this case, it was found that majority of the features have an exponential distribution, meaning that they have many low values and few high values. This can be a challenge for some AI models that assume a normal distribution. Gross order value and exchange rate plots have been shared in figure 5.4 to illustrate this distribution and provide a better understanding of the data. All the pricing features in this dataset follows gross order value distribution i.e., they right skewed with most values ranges at lowest prices and fewer transactions in higher ranges. Similarly numeric features other than likely follows exchange range plot distribution which is generally concentrated with high kurtosis. These distributions explain the patterns and trends within data and pave way to transformation and effectively binning which are key factors in the modeling step.

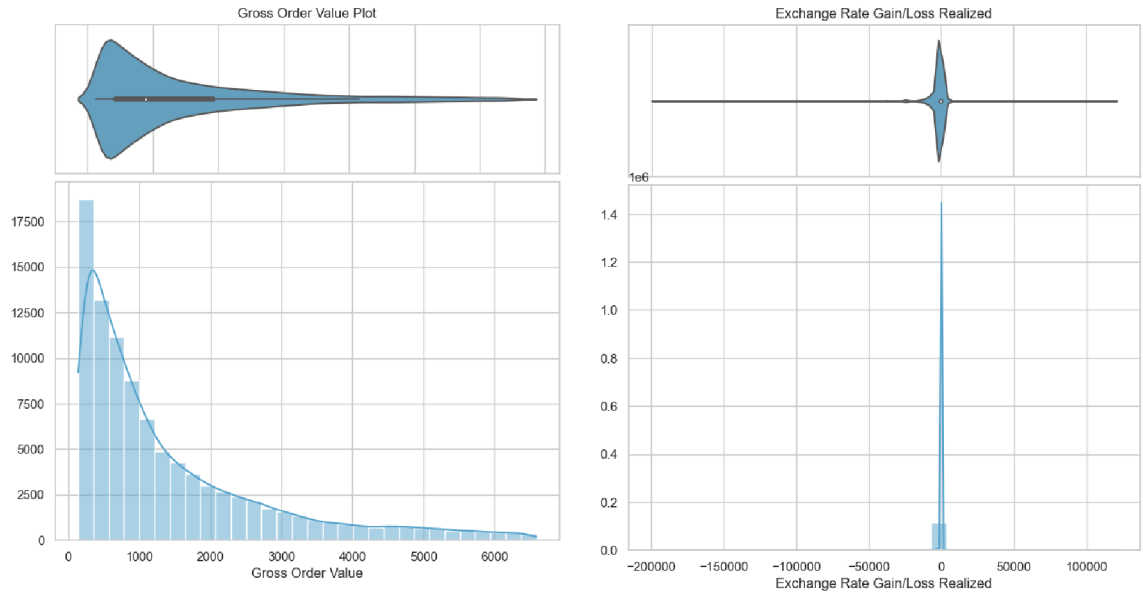



Figure 5.4: Distribution of different features

Next pair plot is also visualized in figure 5.5. varying degrees of correlation among features can be observed. Twenty Six features exhibit a strong correlation others display low correlation, then there are Ninety Four features that show no discernible correlation at all, with data points scattered randomly without any noticeable pattern. These insights from the pair plot are crucial as they can guide the feature selection process and help the research identify which features are most relevant for the model.

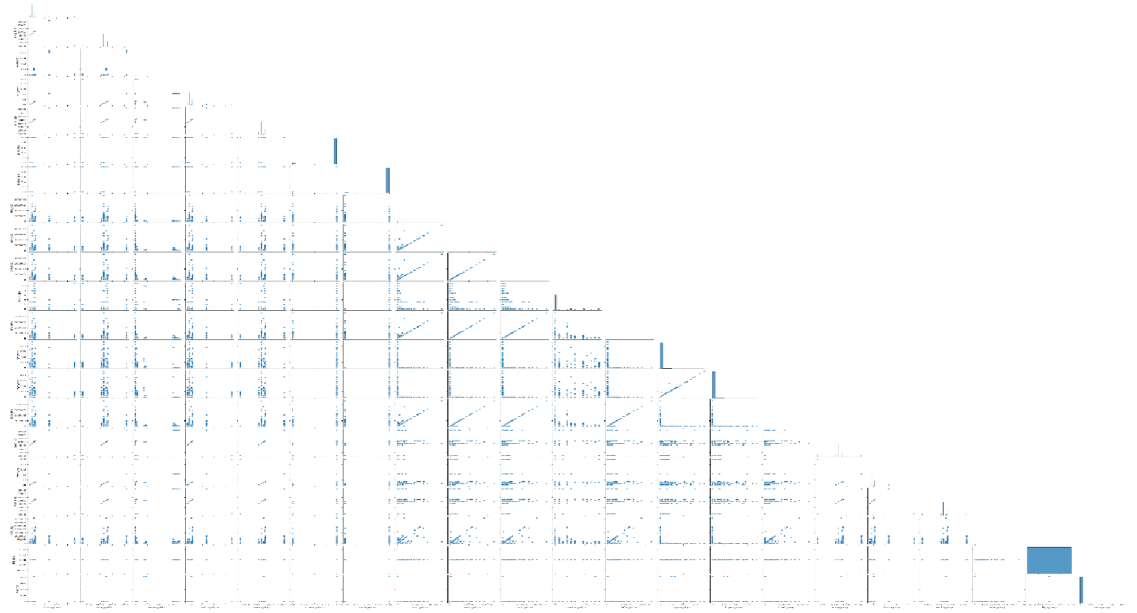


Figure 5.5: Pair plot for continuous features

## Discrete features

Apart from continuous feature, the dataset also consists of categorical features, these features include categories of document, type of invoices, different group of materials and more. For features that are not continuous but discrete, it is also imperative to check their distribution, percentiles to understand their dynamics and employ Cramer's V to calculate the correlation between nominal categorical variables. Nominal categorical variables are variables that have no inherent order or hierarchy. Cramer's V helps us to understand how these variables are related to each other and how they influence our modelling.

Cramer's V is a statistical measure that is used to determine the strength of the relationship between two categorical variables. It is based on the chi-squared statistic, which measures how much the observed frequencies in a contingency table differ from the expected frequencies if the two variables were independent. Cramer's V ranges from 0 to 1, with 0 indicating no association between the two variables and 1 indicating a perfect association. The value of Cramer's V can be interpreted as the strength of the relationship between the two variables, with larger values indicating a stronger relationship.

## Features statistics

Firstly, it's important to observe how different features have different mean, standard deviation, IQR, and percentile from other features, as shown in the Figure 5.6. These statistics can assist in identifying outliers and anomalies in the data. In this analysis, the feature total_chng_num_fields described as total change of fields in a document reveals unique statistics for different categories of documents. As demonstrated in figure 5.6, the document type VERKBELEG presents entirely different statistics compared to other document types. Similarly, EINBELEG also exhibits distinct statistical characteristics. This distinction underscores the diversity of the data within the dataset. The second part of figure 5.6 delves deeper into the stats for one particular document type, 'BELEG'. This in-depth analysis provides invaluable insight into the underlying behavior of different document types. It helps this research by revealing the specific patterns and trends within each document type, thus enhancing our understanding of the dataset and informing our predictive modeling strategy.

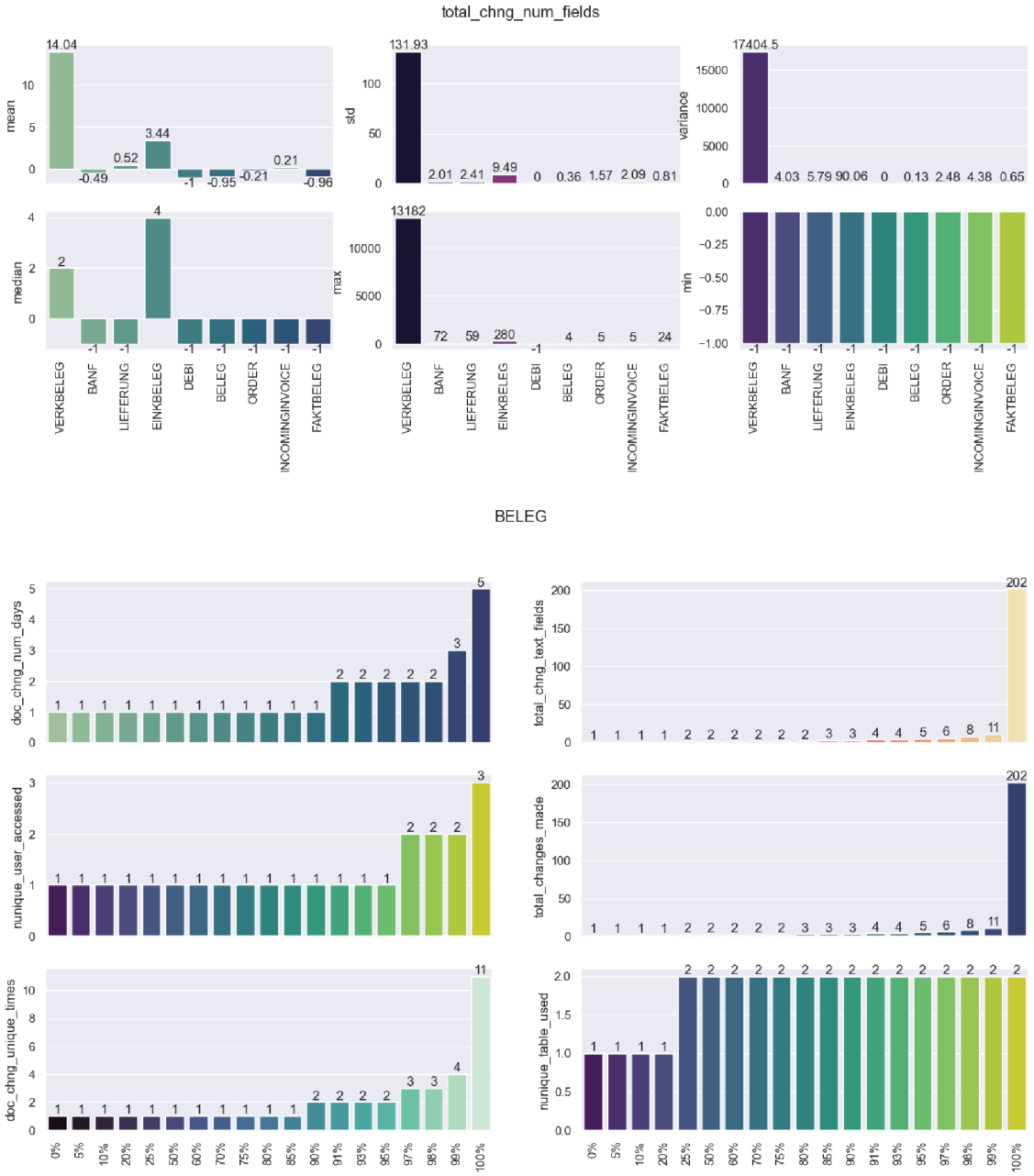



Figure 5.6: Statistics of discrete feature

## Relationship of categorical features

Cramer's V was used to compute the correlation between nominal categorical variables and the findings are shown in table below. This can help this research to identify redundant or irrelevant variables that do not contribute much to our modelling. As shown in figure 5.7 ekgrp_ekko has very high Cramer's V score with kalsm_ekko, lands_ekko, statu_ekko and stceg_l_ekko which tells that these variables are strongly associated with each other.

Cramer's V score, which ranges from 0 to 1, is a measure of association between two nominal variables, providing key insights into the interdependencies between different variables in the dataset. A high Cramer's V score between 'ekgrp_ekko' and 'kalsm_ekko', 'lands_ekko', 'statu_ekpo', and 'stceg_l_ekko' suggests a strong correlation. This means changes in 'ekgrp_ekko' could potentially have a significant impact on these associated variables. similarly, ekgrp_ekko has very low value with statu_ekko, zterm_ekko, meins_ekpo and lgort_ekpo which indicates that these variables are weakly associated or possibly independent of each other. A low Cramer's V score between 'ekgrp_ekko' and 'statu_ekko', 'zterm_ekko', 'meins_ekpo', and 'lgort_ekpo' suggests a weak correlation or potentially no correlation at all. This means that changes in 'ekgrp_ekko' are unlikely to significantly impact these variables.

| | bstyp_ekko | bsart_ekko | statu_ekko | ernam_ekko | spras_ekko | zterm_ekko | ekgrp_ekko | waers_ekko | kalsm_ekko | lands_ekko | stceg_l_ekko |
|---|---|---|---|---|---|---|---|---|---|---|---|
| bstyp_ekko | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN | NaN |
| bsart_ekko | NaN | 0.998872 | 0.015944 | 0.983589 | 0.016209 | 0.071543 | 0.448641 | 0.015980 | 0.019800 | 0.064540 | 0.067920 |
| statu_ekko | NaN | 0.015944 | 1.000000 | 0.716235 | 0.023273 | 0.031674 | 0.090577 | 0.025087 | 0.064655 | 0.075725 | 0.107054 |
| ernam_ekko | NaN | 0.983589 | 0.716235 | 1.000000 | 0.809238 | 0.525807 | 0.556759 | 0.666911 | 0.284460 | 0.367577 | 0.321722 |
| spras_ekko | NaN | 0.016209 | 0.023273 | 0.809238 | 1.000000 | 0.512589 | 0.704051 | 0.901159 | 0.086916 | 0.372446 | 0.371928 |
| zterm_ekko | NaN | 0.071543 | 0.031674 | 0.525807 | 0.512589 | 1.000000 | 0.400776 | 0.479012 | 0.318496 | 0.559515 | 0.557582 |
| ekgrp_ekko | NaN | 0.448641 | 0.090577 | 0.556759 | 0.704051 | 0.400776 | 1.000000 | 0.575701 | 0.999711 | 0.999641 | 0.999662 |
| waers_ekko | NaN | 0.015980 | 0.025087 | 0.666911 | 0.901159 | 0.479012 | 0.575701 | 1.000000 | 0.186856 | 0.297911 | 0.292320 |
| kalsm_ekko | NaN | 0.019800 | 0.064655 | 0.284460 | 0.086916 | 0.318496 | 0.999711 | 0.186856 | 1.000000 | 0.999983 | 0.999982 |
| lands_ekko | NaN | 0.064540 | 0.075725 | 0.367577 | 0.372446 | 0.559515 | 0.999641 | 0.297911 | 0.999983 | 1.000000 | 1.000000 |
| stceg_l_ekko | NaN | 0.067920 | 0.107054 | 0.321722 | 0.371928 | 0.557582 | 0.999662 | 0.292320 | 0.999982 | 1.000000 | 1.000000 |
| statu_ekpo | NaN | 0.364153 | 0.016360 | 0.984533 | 0.030127 | 0.710997 | 0.974885 | 0.034369 | 0.057263 | 0.175753 | 0.080612 |
| matnr_ekpo | NaN | NaN | 0.265596 | 0.602764 | 0.823461 | 0.713324 | 0.436328 | 0.848645 | 0.481983 | 0.477150 | 0.472356 |
| ematn_ekpo | NaN | NaN | 0.265596 | 0.602764 | 0.823461 | 0.713324 | 0.436328 | 0.848645 | 0.481983 | 0.477150 | 0.472356 |
| lgort_ekpo | NaN | 0.000000 | 0.105181 | 0.637605 | 0.010450 | 0.039336 | 0.408248 | 0.048890 | 0.059844 | 0.064630 | 0.051630 |
| meins_ekpo | NaN | 0.004848 | 0.000000 | 0.488611 | 0.044535 | 0.301499 | 0.312284 | 0.044710 | 0.034245 | 0.066185 | 0.026951 |
| bprme_ekpo | NaN | 0.004848 | 0.000000 | 0.489395 | 0.044535 | 0.301550 | 0.312284 | 0.044710 | 0.034245 | 0.066185 | 0.026951 |

Figure 5.7: Discrete feature relations via Cramer’s V

Apart from this, some more sophisticated unsupervised approach was also employed to further understand relation between data via clustering features, feature selection approaches and anomaly detection algorithm in combination with clustering for detecting outliers which were creating issues in clustering features.

## 5.3 Data Wrangling, Processing and Feature Extraction

To ensure data quality and integrity, the Random Forest algorithm was chosen for the imputation process. In this approach, each feature containing missing values was designated as the target variable, while all other features were considered as input variables. The process was executed each time based on 100 iterations, the top 20 features with the highest feature importance scores were identified. These features were selected with great care, as they played a critical role in imputing the missing values of the target variable. By prioritizing these high-importance features, imputations were derived to fill the data gaps while aligning with the underlying data patterns.

It is important to note that an alternative approach was explored, wherein Random Forest was directly used for imputation[24]. However, this technique did not consistently produce the

desired results across all instances. Hence, the strategy of utilizing Random Forest to identify influential features for imputation was deemed more reliable and effective in ensuring the dataset's worthiness and completeness for subsequent analysis.

Further analysis is performed on categorical features with high unique values to understand their contribution to data distribution. It is found that certain categorical features have little to no impact on the distribution of the data. It is determined that these can be dropped from the dataset to enhance the model's robustness and eliminate unnecessary uncertainties. For numeric features, possible data preprocessing options are decided upon. By carefully selecting the most informative features and discarding less relevant ones, the model's ability to generalize to unseen data and make robust reconstruction is enhanced.

A distinct subset of numeric fields exhibited an intriguing facet; they held categorical significance within the context of SAP transactions. To ensure the preservation of these categorical interpretations shown within domain of numeric features, a deliberate conversion to categorical variables was performed. Furthermore, a subset of numeric attributes presented a unique opportunity for enhancement through binning. This process involved the strategic partitioning of numeric values into predefined intervals or bins. Binning, a resourceful technique, essentially restructures continuous data into a more tractable and semantically meaningful format. Upon the completion of binning, the next logical step entailed the application of one-hot encoding to the newly minted bins. This multifaceted approach synergized these strategies, generating an expansive feature set that eclipsed the 1500 mark. It is noteworthy that this feature expansion sprouted from the original 276 features emblematic of a complete transaction.

The resulting larger group of features contained a wide range of insights, capturing both the details of categorized data and grouped versions of numerical data. This wide range of insights allowed the autoencoder model to improve its abilities by understanding complex data patterns and connections. In simple terms, the expanded feature set made the model better at carefully rebuilding input data. This, in turn, allowed the model to not just identify regular transactions, but also find possible unusual activities hidden within the complex landscape of transactions.

## 5.4 Modelling

For the autoencoder model, it was decided to start with 1043 features in the input layer. The subsequent layers in the encoder were designed to have gradually decreasing numbers of neurons. This progressive reduction in the number of neurons aids in capturing the essential

patterns and features in the data, effectively compressing the information into a lower-dimensional latent space. Following the latent space, the decoder architecture was constructed to mirror the encoder's design, with an increase in the number of neurons in each subsequent layer. The decoder's goal is to reconstruct the data from the compressed latent space back to its original representation. To mitigate the risk of overfitting, dropout regularization was incorporated with a dropout rate of 0.2 for each layer in the autoencoder. This is illustrated in figure 5.8.

```
Model: "model_3"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_1 (InputLayer)        [(None, 1043)]            0

 model (Functional)          (None, 3)                 1769131

 model_1 (Functional)        (None, 1043)              1770171

=================================================================
Total params: 3,539,302
Trainable params: 3,539,302
Non-trainable params: 0
_________________________________________________________________
```

Figure 5.8: model overview

## 5.5 Fine Tuning

For the SGD with momentum, a grid search was set up to identify the optimal values for the learning rate and momentum as shown in figure 5.9. The learning rate determines the step size in the optimization process, while momentum introduces a memory-like behavior to accelerate convergence and escape local minima. For AdamW, a variant of the Adam optimizer with weight decay, a similar grid search was performed to determine the optimal values for the learning rate and weight decay. Weight decay is a regularization term that penalizes large weight values, helping to control overfitting. Similarly, for the Adam optimizer, which combines adaptive learning rates with momentum, a grid search was conducted to identify the best learning rate for the model. The adaptive learning rate helps the model to converge quickly by dynamically adjusting the learning rate for each parameter. Additionally, different numbers of epochs were experimented with during hyperparameter tuning. The number of epochs determines the number of times the model goes through the entire dataset during training. Various epoch values were tested to determine the optimal balance between model training time and convergence to the optimal solution.

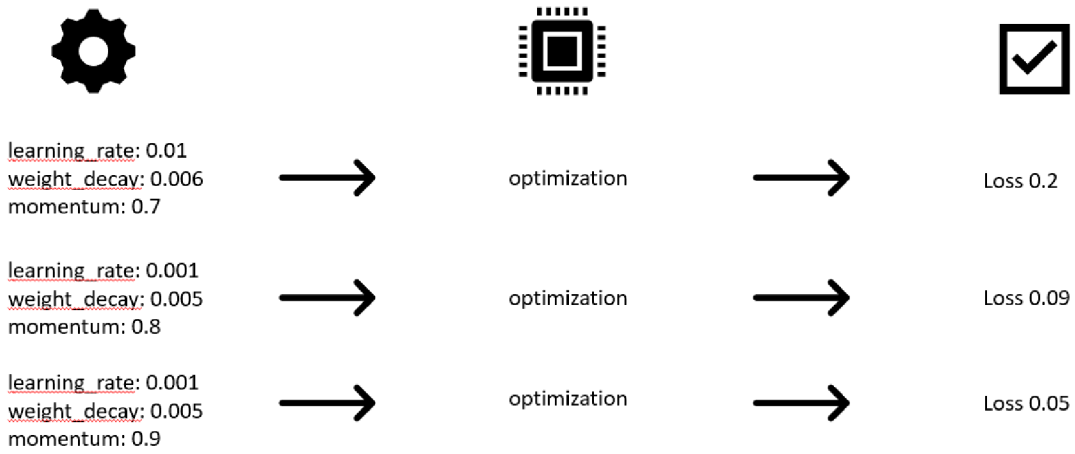



Figure 5.9: illustration of parameters tuning for model

For the anomaly detection and clustering models, there was also a plan to employ grid search with cross-validation (GridSearchCV) to select optimal hyperparameters. However, due to performance issues that will be discussed in the next chapter, hyperparameter tuning was not applied to these models.

# 6. Experimental Results

The performance assessment of the anomaly and clustering algorithms took place within the framework of our anomaly detection task. A complete pipeline of tasks has been executed before this stage involving data collection, data preprocessing, feature engineering, effective imputations to finally data conversion to numeric. Unfortunately, significant challenges were encountered by these algorithms, leading to outcomes that did not meet our expectations. A thorough findings were performed to find the underlying reasons and the core reason behind this underwhelming performance can be traced back to their heavy reliance on the engineered features, which were formulated during the feature engineering phase. In contrast, the original features, which might better capture genuine anomalies or fraudulent behavior, received limited attention. These engineered features, such as anomalies in number of times a document is being changed or numbers of users changing a single document or alterations in multiple fields, might indeed stand as statistical outliers within the dataset. However, their presence doesn't necessarily guarantee the identification of actual instances of anomalies or fraudulent actions. Consequently, the algorithms labeled instances as outliers based on these engineered features, even though such instances might not genuinely represent anomalies in the broader context of the entire transactional cycle, i.e., if a document is being edited after typical working hours or if a document is changed more than 10 times or a document has many currency conversions is surely a statistical outlier, but it cannot be labelled as a fraud.

Similarly, the clustering algorithms encountered similar challenges, primarily revolving around an excessive focus on the engineered features. Their attempt to structure clusters primarily centered around these engineered outliers resulted in outcomes that might not have been optimal. Traditional techniques like the elbow method and silhouette score, commonly employed to determine the ideal number of clusters, could have suggested a limited number of clusters due to the dominance of these outliers in the dataset. Like the elbow method, which is used to calculate the optimal clusters in a dataset result in suggesting 2 clusters as most optimal for our case as shown in figure 6.1 despite the dataset is visualized via compression in 2d space using t-SNE and even after going against optimal cluster selection of 2 the results were not up to the par as shown in figure 6.2. This dominance of outliers further made the clustering algorithms problems even worse. In simple words, this analysis highlights how tricky it can be for these algorithms to sometime cope up with engineered features. They seem to give too much importance to the numbers being created by feature engineering, which can sometimes lead to them thinking something is strange when it might not be. This makes it harder for them to accurately find legitimate anomalies or meaningful data patterns within the transactional data context.

In a pursuit to address the challenges encountered in the previous phases of the research, a final attempt was made by eliminating the engineered features from consideration. Instead, a decision was made to exclusively input the original transactional data into the clustering algorithm, with the anticipation of potentially rectifying the unfavorable outcomes observed earlier. Regrettably, even in this effort, the performance of the clustering algorithm did not exhibit the desired level of improvement as shown in figure 6.3. This approach was hoped to reduce the potential influence of any distortions or misinterpretations that could have stemmed from the incorporation of engineered features. However, the results obtained indicated that the limitations of the clustering algorithm were not solely attributed to the presence of engineered features. Rather, there seemed to be fundamental issues within the

algorithm's capacity to handle the complexities and intricacies of the transactional data of such high dimensions.

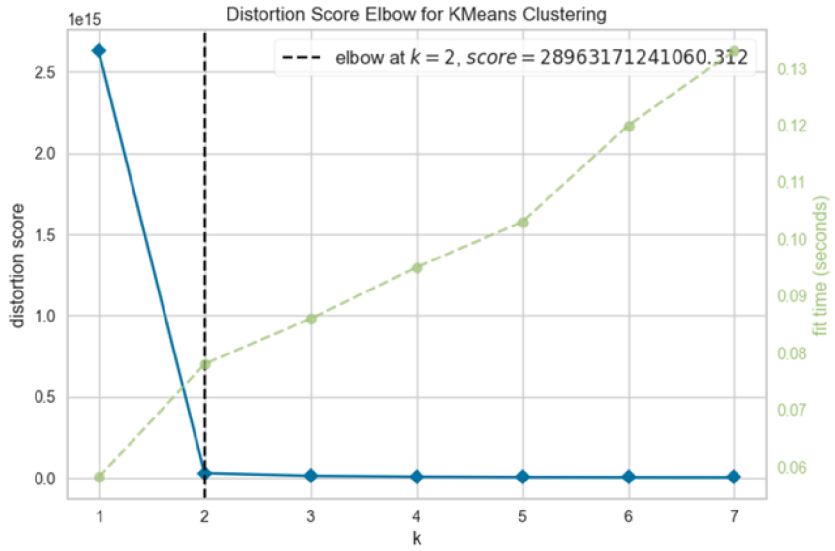



Figure 6.1: Elbow method suggestion for clusters

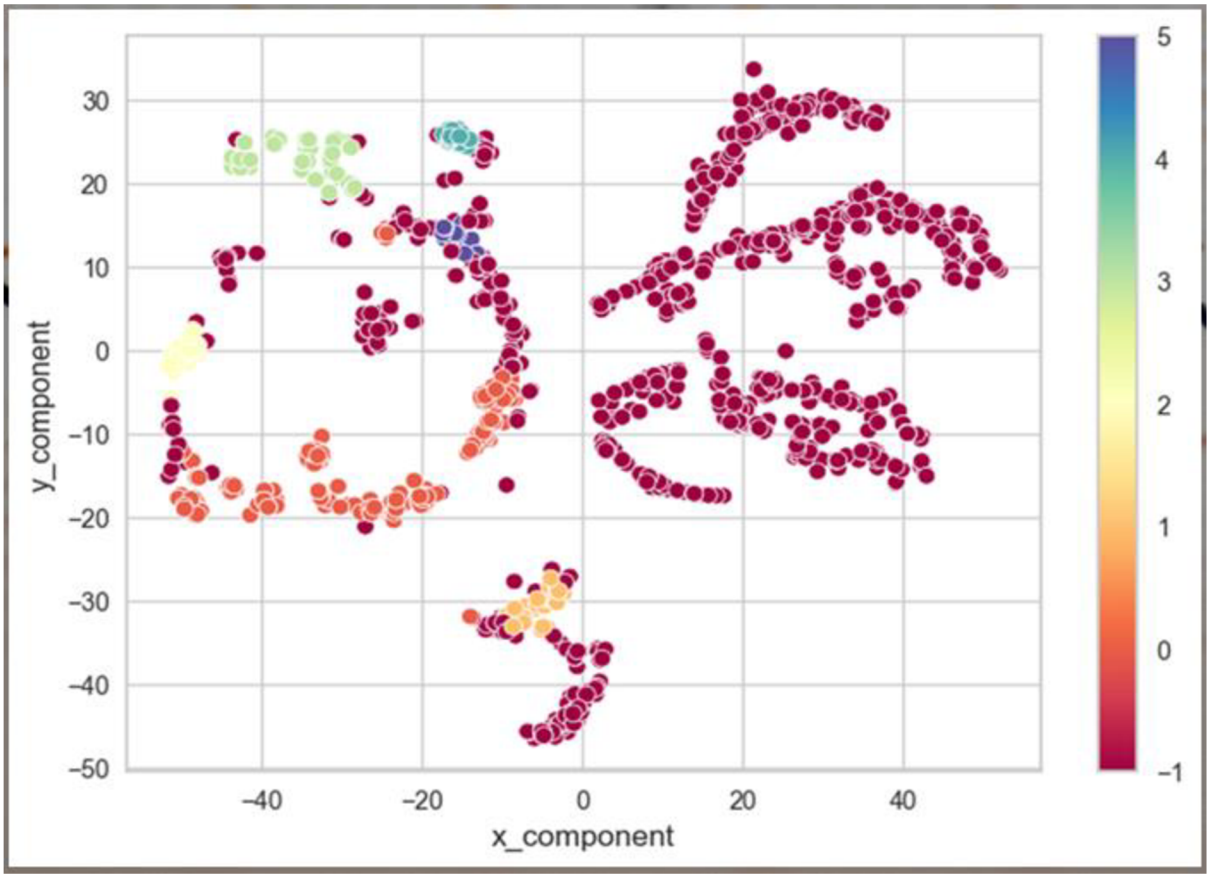



Figure 6.2: Clustering outcome in 2 dimensions with engineered features

In view of the relatively low performance exhibited by these algorithms within the context of our specific utilization, a deliberate and conscious decision not to invest further effort or additional resources in hyperparameter tuning these algorithms. The shortcomings displayed by these algorithms in effectively capturing authentic anomalies or discerning meaningful patterns within our data underscored their inherent limitations in handling the complexities of our high dimensional transactional cycle data. It is also imperative to note that these limitations and constraints do not diminish the value of these algorithms which can be impressively employed in other contexts or different datasets. Anomaly detection and clustering algorithms can be highly effective in various scenarios, especially when applied to datasets that align with their assumptions and requirements. While these algorithms may not have yielded the optimal outcomes that was sought for this transactional data, it remains crucial to recognize that our research, characterized by the high dimensionality of the end-to-end transactional cycle data and the introduction of engineered features, presented formidable challenges that these algorithms encountered and may not be able to comprehend correctly. Although these algorithms didn't give us the best results for our transactional data, their overall worth remains. This situation encourages us to keep looking for different methods and strategies to achieve strong anomaly detection specifically within our field. This persistent quest for attaining better solutions keeps the motivation alive and serves as driving force to showcases the dynamic nature of research in finding the right approaches to address and tackle complex challenges. This ongoing search for improvement and finding appropriate solutions keeps the sense of exploring alive. This is fueled by the strong drive to find new and creative ways to solve problems. This highlights how research is all about adapting and changing over time to keep getting better at what is being done.

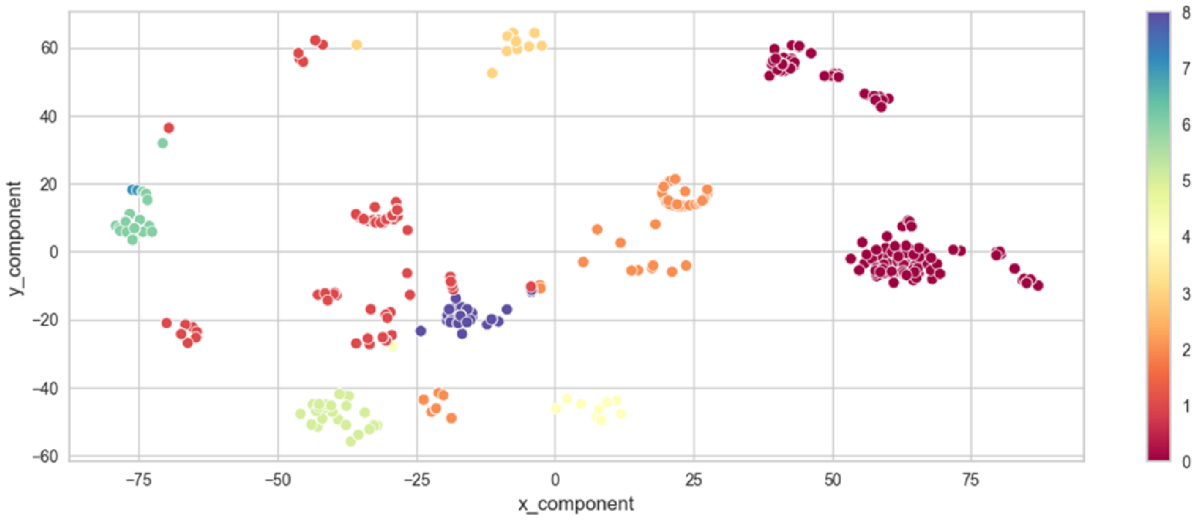



Figure 6.3: Clustering outcome in 2 dimensions without engineered features

Following the suboptimal outcomes generated from the anomaly detection and clustering algorithms, our attention naturally moved towards exploring alternative solutions, specifically focusing on neural networks as a potential solution. Within this exploratory scope, the research led to the realm of autoencoders, a class of neural networks that exhibited promise in tackling the strong challenges posed by our complex and high-dimensional transactional data

landscape. Autoencoders, characterized by their capacity to reconstruct input data, emerged as a viable candidate due to their inherent ability to learn meaningful patterns within the data. Given the complicated nature of transactional data, full of complexities and high-dimensional aspects, the application of autoencoders offers the prospect of unearthing latent structures that may evade conventional algorithms. These neural networks-based solutions might possess the potential to unravel subtle anomalies and trends, while simultaneously addressing the challenges of noise and dimensionality inherent within our dataset. In general, the research towards autoencoders signifies our aspiration to harness the power of neural networks in addressing the complexities unique to our domain data.

The structural configuration of the autoencoder is characterized by the construction of both the encoder and decoder segments in a similar manner. These segments collectively facilitate the transformation and subsequent reconstruction of the input data. Within this framework, the models exhibit a symmetrical progression, commencing with the intake of input data by the encoder. This initial phase is followed by a sequential passage through a series of intermediary layers. These layers are combined with the inclusion of dropout mechanisms and leaky_relu as activation function, which systematically decrease the number of neurons. This reduction transpires through a succession of stages, orchestrating a gradual transition from 1024 neurons to 512, further decreasing neurons to 256, and lastly to 4 neurons. This condensed ensemble of neurons functions as the gateway to the subsequent phase: the latent space, which consists of 3 neurons. On the other hand, the decoder module. Like its encoder counterpart have a comparable architecture. The journey of reconstruction of data starts within the latent space, where the data is then undergoing a symmetrical process of layers to a proper reconstruction. This architectural framework, consists of the encoder's encoding and the decoder's decoding, conveys the autoencoder's distinctive capacity to deconstruct and subsequently reconstruct the data.

The performance of autoencoders stood out notably, showcasing promising results when applied to the SAP transactional data that encompasses the entire process. In order to comprehensively assess their capabilities, a thorough experiment was carried out, refining the hyperparameters to achieve a comprehensive understanding. This involved a systematic exploration of different combinations of optimizers and epochs within the autoencoder framework. The experiment consists of a multi-step process. The initial phase centered around evaluating distinct optimizers, namely AdamW, Adam, and SGD with momentum. Each optimizer was subjected to perform 3 testing instances, employing 25, 50, and 100 epochs respectively. This deliberate variation in epochs allowed for a comprehensive examination of the model's behavior under differing timeframes. The reconstruction loss to gauge the effectiveness of the autoencoder's performance, a pivotal metric was employed. This metric gauge the dissimilarity between the original input data and the data that the autoencoder reconstructs. In general rule of thumb, a lower reconstruction loss signifies the model's proficiency in recreating the input data with precision. Remarkably, the outcomes across all instances consistently demonstrated a reconstruction loss that remained below 0.05. Interestingly, both AdamW and Adam optimizers yielded comparable outcomes as shown in figure 6.5 and table 6.1 respectively, especially evident in the 50 and 100 epoch scenarios when employing the validation dataset. This outcome suggests that both optimizers effectively captured the intrinsic data patterns and efficiently reconstructed the input data. In contrast, the SGD optimizer exhibited slightly higher reconstruction loss in comparison to its counterparts. This divergence in performance potentially indicates that the SGD optimizer

might have encountered relatively more challenges in reconstructing data within this specific context.

Additionally, our experimentation extended to encompass the evaluation of the models at higher epochs, specifically 200 and 300. However, the outcomes of these extended epochs training did not yield any substantial improvement in the model's performance. Consequently, a conclusion can be drawn, indicating that the optimal reconstruction loss for our specific use case is effectively achieved within 100 epochs. Beyond this training epoch's limit, any further training fails to yield or achieve improvements in the model's performance, as evidenced by the lack of substantial progress in data reconstruction as shown in figure 6.4. Through this observation and it can be concluded that for our specific use case model reaches optimal reconstruction loss within 100 epochs beyond which additional training does not significantly contribute to augmenting the model's proficiency within our transactional data landscape.

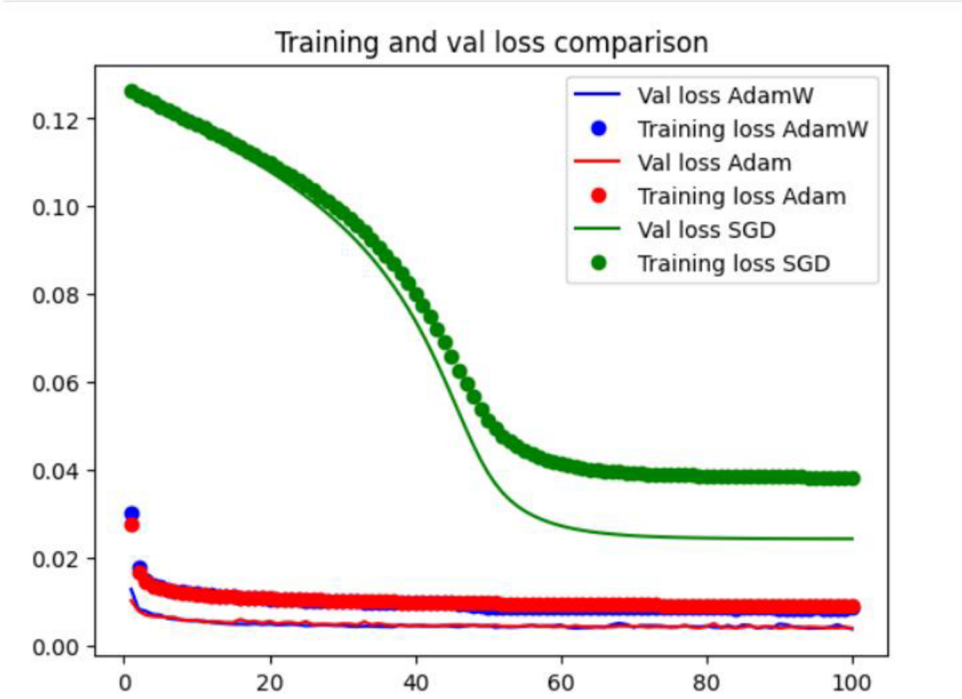



Figure 6.4: Training and Validation results of models

In the pursuit of further optimization and finetuning the autoencoder model, A series of plans were started for exploring various architectural modifications. These optimizations and finetuning plans involved experimenting with alterations in the structure of the model, encompassing both the reduction and augmentation of layers, along with the manipulation of neuron counts within each layer, as well as adjustments to the dropout rate – a technique aimed at mitigating overfitting. However, the outcomes generated from these additional experiments did not yield the desired optimal enhancements. Curiously, the autoencoder architecture initially deployed in the experiment consistently generated the most favorable results. This intriguing observation underscores the compatibility between the autoencoder's architecture and the inherent attributes of the comprehensive transactional cycle data. Furthermore, it implies that introducing additional architectural adjustments to autoencoder model was not the best approach and it did not yield substantial performance improvements in

reconstruction loss. Evidently, the first autoencoder's architecture was well-aligned with the complex nature of the SAP transaction data, rendering further changes unjustifiable. The robustness of the autoencoder's performance across different optimizers and epochs, combined with the architecture's stability, makes it a highly promising approach for anomaly detection in the given dataset. The consistently low reconstruction loss suggests that the autoencoder effectively captured the relevant data distributions and learned essential patterns in the SAP transaction data.

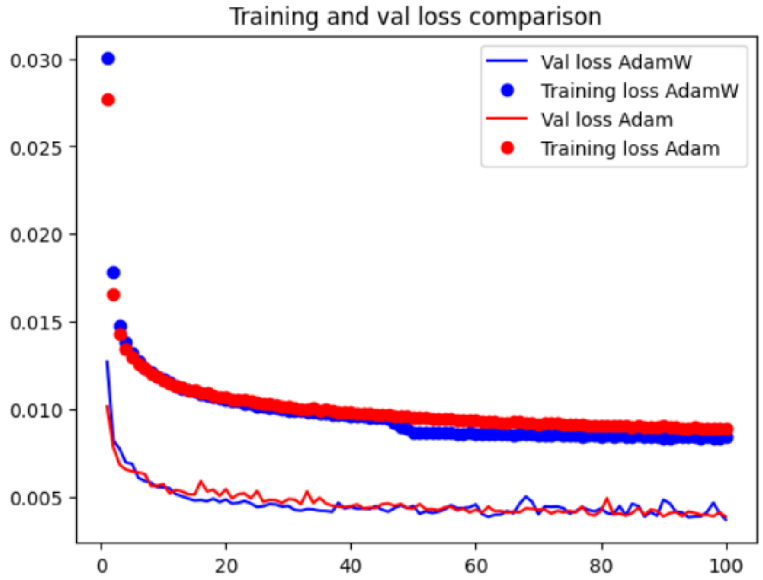



Figure 6.5: Comparison of results of Adam and AdamW optimizers

| | Autoencoder with optimized Adam | | Autoencoder with optimized AdamW | | Autoencoder with optimized SGD | |
|---|---|---|---|---|---|---|
| Epochs | Train loss | Validation loss | Train loss | Validation loss | Train loss | Validation loss |
| 25 | 0.01 | 0.004 | 0.0102 | 0.004 | 0.1059 | 0.104 |
| 50 | 0.0091 | 0.003 | 0.0087 | 0.003 | 0.0589 | 0.050 |
| 100 | 0.0089 | 0.003 | 0.0078 | 0.003 | 0.0382 | 0.024 |

Table 6.1: Comparison of models with different optimizers

In our pursuit of establishing the credibility of the autoencoder model performance, a comprehensive validation approach is adopted. This endeavor begins with the inclusion of anomalous data, carefully identified through a thorough application of predefined rules. These rules were thoughtfully designed, drawing upon a wealth of domain-specific insights and expertise, strategically woven to encapsulate plausible instances of anomalies or potentially fraudulent transactions within the context of SAP data. This process of crafting rules evolved into a precise orchestration, fueled by a deep understanding of the domain's complex details and an accumulation of specialized knowledge of SAP. The derived set of anomalous data, thus obtained through the application of these rules, assumed a pivotal role as the main source of

our validation process. This data acted as a ground truth against which the performance of the autoencoder models was gauged specifically. The process of validation hinged on this foundation of real-world anomalies, enabling a thorough assessment of the autoencoder models potential in identifying and addressing deviations from the normal distribution.

For the comprehensive evaluation of the autoencoder model, a critical preliminary step entailed the transformation of the identified anomalous data. This transformation was a deliberate attempt to harmonize the inherent characteristics of the anomalous instances with the autoencoder’s data processing mechanisms. To facilitate this alignment, the anomalous instances underwent a strategic transformation via the data pipeline step, evolving into a format optimally suited for the autoencoder’s input layer. This transformation process, while seemingly technical, played a crucial role in ensuring that the autoencoder could effectively be able to detect the complex and high dimensional part of the anomalous data. This rule-based data conversion ensured that the autoencoder's capabilities were precisely put to the test. It allowed the model to navigate through the nuances of the transformed data, skillfully identifying patterns that might signify anomalies. This validation procedure, anchored in real-world anomalies and supported by the transformation of data, further solidified the autoencoder models' standing as a potent tool in the sphere of anomaly detection within the landscape of SAP data.

Next, the performance of the different autoencoder models was gauged using the transformed anomalous data. A group of 500 sets of transactions were selected using different rules-based approaches. Each of these sets had 20 anomalies that were a bit different from the usual. Surprisingly, the autoencoder model did a great job as shown in table 6.2. It managed to find all 20 different anomalies in 465 out of the 500 sets of transactions. This is really impressive and shows that the model is good at understanding and handling these complex anomalies. However, the detection process wasn't the same for all cases. In 17 sets of transactions, the model found 19 out of the 20 different anomalies. This is a really good result and shows that the model can understand most of the changes. In 9 other cases, the model correctly spotted 18 out of the 20 different anomalies which is also quite accurate. The exploration of the model's performance continued. In 3 cases, the model correctly identified 17 out of the 20 different things, which is impressive. This trend continued in 6 more cases, where the model spotted 16 out of the 20 different things, showing its ability once again.

One important thing to note is that none of the transactions managed to escape the model's detection. Even though the model might have missed a few different anomalies in some cases, it didn't miss any entire set of transactions. This overall performance is noteworthy and impressive and shows that the model is reliable. This detailed analysis highlights the fact that the autoencoder model is quite impressive at finding unusual transactions. The results confirm that the model can handle complex data like SAP transactions. Thus making autoencoder the better choice to be deployed for future transactions.

| Anomalies | Detected by Model | Percentage |
|---|---|---|
| 500 (20 per transaction) | 465 (all 20 detected) | 93% |
| | 17 (19 out of 20 detected) | 3.4% |
| | 9 (18 out of 20 detected) | 1.8% |
| | 3 (17 out of 20 detected) | 0.6% |
| | 6 (16 out of 20 detected) | 1.2% |

Table 6.2: Model results with known anomalies

In the pursuit of greater transparency and comprehensibility of the autoencoder model, a decision was made to extract the output at the latent space of the SAP data, which includes anomalies, and represent it visually. This process was primarily aimed at enhancing the model's explainability, and providing a clear illustration of how it interprets data that inherently deviates from the distribution.

The latent space, composed of three neurons, offered three dimensions for plotting. This multi-dimensional representation, as depicted in Figure 6.6, was leveraged to scrutinize how the model comprehends the data which is inherently out of distribution. The visualization in the three-dimensional latent space essentially allowed for a more tangible understanding of how the model processes anomalies in the SAP data.

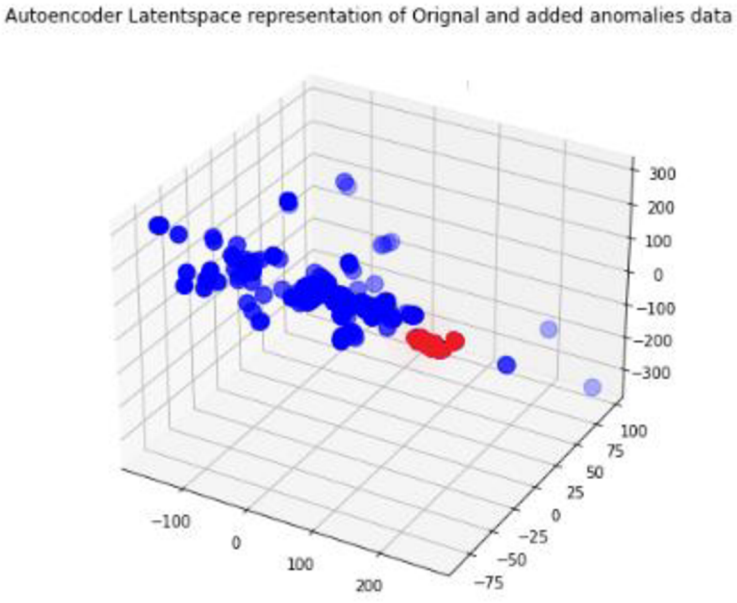
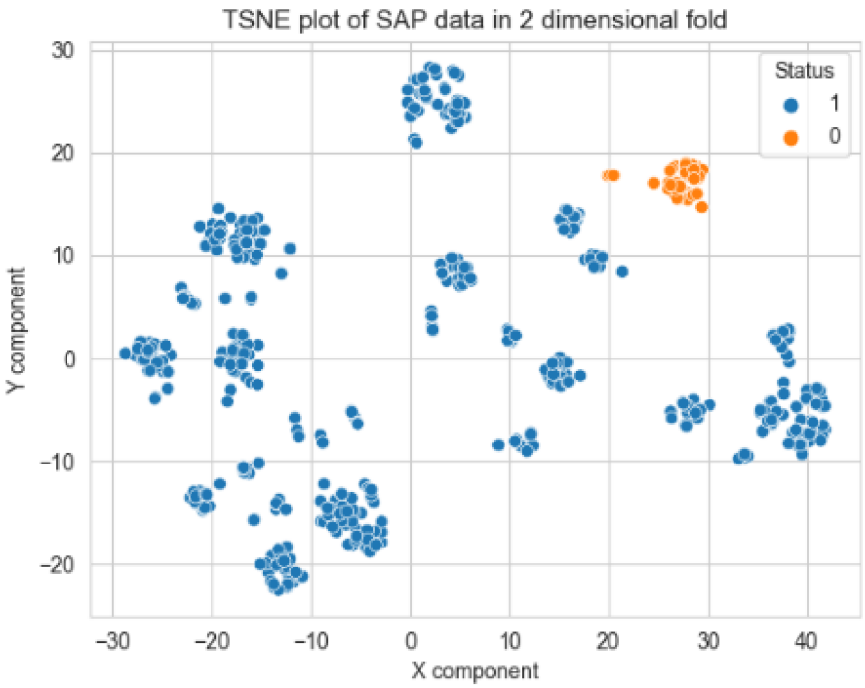



Figure 6.6 Autoencoder output at latent space

In a bid to further augment the model's explainability and enable more comprehensive visual verification, an additional step was taken. The output of the autoencoder model's decoder was compressed to two dimensions utilizing t-distributed Stochastic Neighbor Embedding (TSNE) technique as visualized in figure 6.7. This process was undertaken to simplify the high-dimensional data and present it in a more interpretable form. Subsequently, the two-dimensional output of the decoder was plotted, with specific added anomalous transactions marked as '0'. These marked transactions represent instances where the model identified a reconstruction error, suggesting that it considered these transactions as anomalies or potential frauds. This aligns with the model's intended functionality of detecting irregularities in the data.




6.7 TSNE plot for original data and detected (added) anomalies after decoder step.

The results of this process were in line with expectations and corroborated the findings presented in Table 6.2. This demonstrates the utility of the autoencoder model in accurately identifying and representing anomalies within the SAP data. Therefore, the combination of visual checks, latent space output examination, TSNE compression, and error marking, collectively contribute to a more thorough understanding and explainability of the autoencoder model.

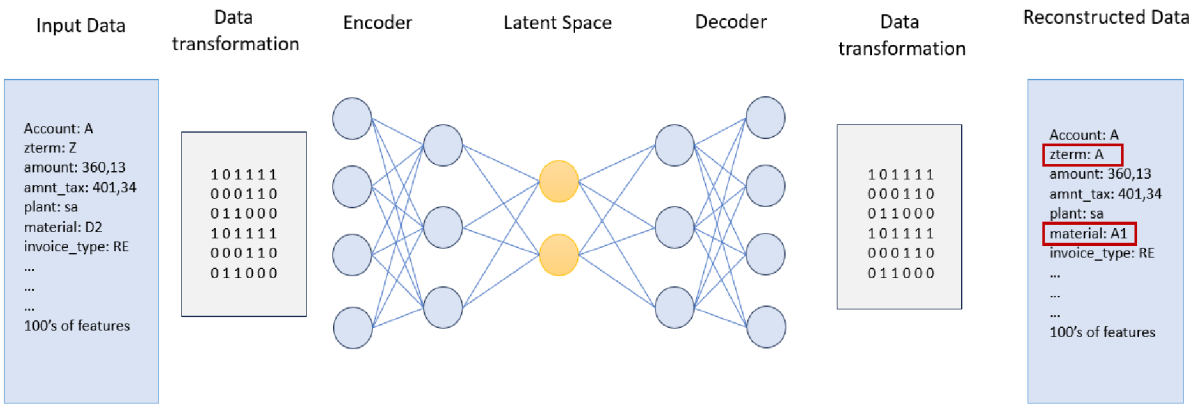



Figure 6.8: Complete end to end cycle of SAP data.

In Figure 6.8, a comprehensive explanation and visual representation of the entire data transformation process, from beginning to end, is provided. This depiction serves as a crucial guide to understanding the different stages of the research experiment, showcasing the journey taken by SAP data to align with the specific requirements of the use case and emerge as the most optimal solution for the given dataset needs. Through this visual representation, the aim is to display the complete process through which SAP data is transformed, step by step, to seamlessly integrate with the autoencoder's encoder. The diagram further delves into the emergence of the latent space, a critical component in this journey, and finally ends with the reconstruction process performed by the decoder. In essence, Figure 6.8 acts as a graphical narrative, encapsulating the essence of the research and portraying the comprehensive path that SAP data traverses to effectively fulfill the research objectives.

.

# 7. Discussion

The autoencoder model showed significant accuracy in detecting real-world anomalous data. However, there were a few instances where the model also reconstructed false positives, marking non-anomalous instances as anomalous. This issue was examined to understand the underlying reasons behind such occurrences. Upon further investigation, it was found that the model's false positives were attributed to instances with rare data distributions. In specific cases, certain instances were infrequently observed in the dataset, making them uncommon and challenging for the autoencoder to accurately reconstruct. Consequently, the model classified these instances as anomalies due to its reliance on learned embeddings that did not adequately capture the rarity of such data patterns. Our investigation revealed that these sensitive features and instances with rare data distributions were not typical in the general dataset. For example, instances involving the granting of additional discounts to buyers or allowing vendors to pay after an extended period of 120 days post-purchase were legitimate but rare practices. Identifying such patterns as anomalies was a positive outcome for the model since it successfully detected irregular and infrequent occurrences that might warrant closer scrutiny.

Comparing this research with existing studies in the field, autoencoders have been recommended in several prior works for anomaly detection in SAP transaction data. However, notable improvements were made in this approach by utilizing real-world data and pushing the model's limits to learn from a significantly larger and diverse dataset. Rather than selecting only a few fields and tables, the autoencoder model in this study covered the complete transaction cycle, enhancing its ability to detect anomalies in a more comprehensive and contextually relevant manner. The performance of the autoencoder model in detecting anomalies in this research is a promising development. However, it is proposed that future research in this direction should address certain limitations. One particular limitation encountered was the challenge of incorporating numeric data like prices and taxes in the reconstruction process. In this research case, the option was to bin these numeric features, which might have led to some loss of information and precision. Future research could explore more sophisticated techniques to effectively reconstruct numeric data in a way that retains its original granularity and accuracy.

# 8. Conclusion

The autoencoder model proved to be a powerful tool for detecting anomalies in real-world transactional data. Although it occasionally reconstructed false positives, the underlying reasons were identified, largely related to rare data distributions and sensitive features. The research showcased significant improvements over existing studies by utilizing real-world data and leveraging a more extensive and diverse dataset. While the autoencoder's performance was commendable, it is suggested that future research addresses the challenge of incorporating numeric data in the reconstruction process to further enhance the model's accuracy and utility in anomaly detection for transactional data. These advancements can contribute to more robust and effective anomaly detection solutions in SAP transactional environments and similar contexts.

# References

[1] Ki, Y. and Yoon, J.W. (2018) PD-FDS: Purchase Density Based Online Credit Card Fraud Detection System, Proceedings of the KDD 2017: Workshop on Anomaly Detection in Finance. Available at: https://proceedings.mlr.press/v71/ki18a.html (accessed April 16, 2023)

[2] Bakumenko, A. and Elragal, A. (2022) Detecting anomalies in financial data using machine learning algorithms, MDPI. Available at: https://www.mdpi.com/2079-8954/10/5/130 (accessed April 16, 2023)

[3] Hamelers, L.H. (2021) Detecting and explaining potential financial fraud cases in invoice data with Machine Learning. Available at: http://essay.utwente.nl/85533/ (accessed April 22, 2023)

[4] Tritscher, J. et al. (2022) Open ERP system data for Occupational Fraud Detection, arXiv.org. Available at: https://arxiv.org/abs/2206.04460 (accessed May 11, 2023)

[5] Baader, G. and Krcmar, H. (2018) Reducing false positives in fraud detection: Combining the Red Flag Approach with process mining, International Journal of Accounting Information Systems. Available at: https://www.sciencedirect.com/science/article/pii/S146708951630077X (accessed May 23, 2023)

[6] Schultz, M. and Tropmann-Frick, M. (2020) Autoencoder neural networks versus external auditors: Detecting unusual Journal Entries in Financial Statement Audits. In Proceedings of the 53rd Hawaii International Conference on System Sciences. Available at: https://www.semanticscholar.org/paper/Autoencoder-Neural-Networks-versus-External-Unusual-Schultz-Tropmann-Frick/093de7204b44b9ebc5c1efc2ded698165bd8478e (accessed April 12, 2023)

[7] Schreyer, M. et al. (2018) Detection of anomalies in large scale accounting data using Deep Autoencoder Networks, arXiv.org. Available at: https://arxiv.org/abs/1709.05254 (accessed June 02, 2023)

[8] Mario Zupan ,Svjetlana Letinic ,Verica Budimir (2018) Journal Entries with Deep Learning Model , International Journal of Advance Computational Engineering and Networking (IJACEN) , pp. 55-58, Volume-6, Issue-10. Available at: http://iraj.doionline.org/dx/IJACEN-IRAJ-DOIONLINE-13840 (accessed May 21, 2023)

[9] Oliverio, W.F.M. et al. (2019) A hybrid model for fraud detection on purchase orders, SpringerLink. In: Intelligent Data Engineering and Automated Learning – IDEAL 2019. Cham, Switzerland. Available at: https://link.springer.com/chapter/10.1007/978-3-030-33607-3_13 (accessed April 24, 2023)

[10] Kishore Singh & Peter Best, (2016) Interactive visual analysis of anomalous accounts payable transactions in SAP enterprise systems, In: Managerial Auditing Journal. Available at: https://ideas.repec.org/a/eme/majpps/maj-10-2014-1117.html (accessed May 04, 2023)

[11] ACFE (2022) "Occupational Fraud 2022: A Report to the nations". Available at: https://legacy.acfe.com/report-to-the-nations/2022/ (accessed April 27, 2023)

[12] ACFE (2020) 2020 Global Occupational Fraud Study. Report To the Nations. Available at: https://legacy.acfe.com/report-to-the-nations/2020/ (accessed April 07, 2023)

[13] Nonnenmacher, J. and Jorge, M.G. (2021) Unsupervised anomaly detection for internal auditing: Literature review and research agenda. Available at: http://rabida.uhu.es/dspace/bitstream/handle/10272/19256/Unsupervised.pdf (accessed May 07, 2023)

[14] IFAC. International Standards on Auditing 240, The Auditor's Responsibilities Relating to Fraud in an Audit of Financial Statements. (2009). https://www.ifac.org/_flysystem/azure-private/publications/files/A012%202013%20IAASB%20Handbook%20ISA%20240.pdf (accessed May 02, 2023)

[15] S. Becirovic, E. Zunic and D. Donko, (2020) A Case Study of Cluster-based and Histogram-based Multivariate Anomaly Detection Approach in General Ledgers, 19th International Symposium INFOTEH-JAHORINA (INFOTEH), East Sarajevo, Bosnia and Herzegovina, 2020, pp. 1-6, Available at: https://ieeexplore.ieee.org/document/9066333 (accessed June 17, 2023)

[16] KDD '20 (2020) Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining Pages 3035–3043. Available at: https://doi.org/10.1145/3394486.3403354 (accessed April 29, 2023)

[17] Parikshit Ram, Alexander G. Gray (2018). Fraud Detection with Density Estimation Trees. Proceedings of the KDD 2017: Workshop on Anomaly Detection in Finance: 85-94. Available from https://proceedings.mlr.press/v71/ram18a.html (accessed May 11, 2023)

[18] Perera, Pramuditha, and Vishal M. Patel. (2019) Learning Deep Features for One-Class Classification. Available at: https://arxiv.org/abs/1801.05365 (accessed June 07, 2023)

[19] Seliya, N., Abdollah Zadeh, A. & Khoshgoftaar, T.M. (2021) A literature review on one-class classification and its potential applications in big data. J Big Data 8, 122. Available at: https://doi.org/10.1186/s40537-021-00514-x (accessed June 15, 2023)

[20] Bing Xu, Naiyan Wang (2015) Empirical Evaluation of Rectified Activations in Convolution Network. Available at: https://arxiv.org/pdf/1505.00853.pdf (accessed June 24, 2023)

[21] Ilya Loshchilov & Frank Hutter (2019) Decoupled Weight Decay Regularization, Published as a conference paper at ICLR 2019. Available at: https://arxiv.org/abs/1711.05101 (accessed July 03, 2023)

[22] Chandola, V., Banerjee, A., and Kumar, V. (2009). Anomaly detection: A survey. ACM computing surveys (CSUR). Available at: https://dl.acm.org/doi/10.1145/1541880.1541882 (accessed May 05, 2023)

[23] Chen S, Guo W. (2023) Auto-Encoders in Deep Learning—A Review with New Perspectives. Mathematics.11(8):1777. Available at: https://doi.org/10.3390/math11081777 (accessed July 12, 2023)

[24] Tang, F. and Ishwaran, H. (2017) Random Forest Missing Data Algorithms, arXiv.org. Available at: https://arxiv.org/abs/1701.05305 (accessed April 26, 2023)

[25] Bank, D., Koenigstein, N. and Giryes, R. (2021) Autoencoders, arXiv.org. Available at: https://arxiv.org/abs/2003.05991 (accessed June 23, 2023)

[26] Yang, K., Kpotufe, S. and Feamster, N. (2021) An Efficient One-Class SVM for Anomaly Detection in the Internet of Things, arXiv.org. Available at: https://arxiv.org/abs/2104.11146 (accessed May 18, 2023)

[27] Xu, H. et al. (2023) Deep Isolation Forest for Anomaly Detection, arXiv.org. Available at: https://arxiv.org/abs/2206.06602 (accessed May 18, 2023)

[28] R. Xu and D. Wunsch. (2009) Clustering. Wiley-IEEE Press. ISBN 9780470276808. Available at: https://onlinelibrary.wiley.com/doi/book/10.1002/9780470382776 (accessed May 22, 2023)

[29] Bank, E. (2019). Fifth report on card fraud. European Central Bank: Frankfurt am Main, Germany. Available at: https://www.ecb.europa.eu/pub/cardfraud/html/ecb.cardfraudreport201809.en.html (accessed May 13, 2023)