KADİR HAS UNIVERSITY
SCHOOL OF GRADUATE STUDIES
PROGRAM OF COMPUTER ENGINEERING

ANOMALY DETECTION VIA MACHINE LEARNING

GÖRKEM ERDEM

MASTER OF SCIENCE THESIS

İSTANBUL, ŞUBAT, 2023

ANOMALY DETECTION VIA MACHINE LEARNING

GÖRKEM ERDEM
ADVISOR: PROF. FEZA KERESTECIOĞLU
CO-ADVISOR: ASST. PROF. MESUT ÇEVIK

A thesis submitted to
the School of Graduate Studies of Kadir Has University
in partial fulfilment of the requirements for the degree of
Master of Science in
Computer Engineering

İstanbul, February, 2023

APPROVAL

This thesis titled ANOMALY DETECTION VIA MACHINE LEARNING submitted by GÖRKEM ERDEM, in partial fulfillment of the requirements for the degree of Master of Science in Computer Engineering is approved by

Prof. Feza Kerestecioğlu (Advisor) .....
Kadir Has University

Asst. Prof. Mesut Çevik (Co-Advisor) .....
Altınbaş University

Asst. Prof. Hasan Abdulkader .....
Altınbaş University

Asst. Prof. Baran Tander .....
Istanbul Aydın University

Assoc. Prof. Atilla Özmen .....
Kadir Has University

I confirm that the signatures above belong to the aforementioned faculty members.

.....
Prof. Dr. Mehmet Timur Aydemir
Director of the School of Graduate Studies
Date of Approval: 05.01.2023

DECLARATION ON RESEARCH ETHICS AND PUBLISHING METHODS

I, GÖRKEM ERDEM; hereby declare

- that this Master of Science Thesis that I have submitted is entirely my own work and I have cited and referenced all material and results that are not my own in accordance with the rules;
- that this Master of Science Thesis does not contain any material from any research submitted or accepted to obtain a degree or diploma at another educational institution;
- and that I commit and undertake to follow the "Kadir Has University Academic Codes and Conduct" prepared in accordance with the "Higher Education Council Codes of Conduct".

In addition, I acknowledge that any claim of irregularity that may arise in relation to this work will result in a disciplinary action in accordance with university legislation.

GÖRKEM ERDEM

.....

05.01.2023

To my dear family

ACKNOWLEDGEMENT

I would like to thank my advisors Asst. Prof. Mesut Çevik and Prof. Feza Kerestecioğlu, all my professors in the department, my beloved family, my managers Ahmet Gözmen, Birol Yüceoğlu and Ömer Zeybek, who helped me be successful in my academic and business life. I would also like to thank Migros Ticaret A.Ş. for allowing me to use company data.

ABSTRACT

Retail companies monitor inventory stock levels regularly and manage stock levels based on forecasted sales to sustain their market position. The accuracy of inventory stocks is critical for retail companies to create a correct strategy. Many retail companies try to detect and prevent inventory record inaccuracy caused by employee or customer theft, damage or spoilage and wrong shipments. This study is aimed to detect inaccurate stocks using machine learning methods. It uses the real inventory stock data of Migros Ticaret A.Ş. of Turkey's largest supermarket chains. A multiple of machine learning algorithms such as Isolation Forest (IF), Local Outlier Factor (LOF), One-Class Support Vector Machine (OCSVM) were used to detect abnormal stock values. On the other hand, generally, researchers use public data to develop methods, and it is challenging to apply machine learning algorithms to real-life data, especially in unsupervised learning. This thesis shows how to handle real-life data noises, missing values etc. The experimental findings show the performances of machine learning methods in detecting anomalies in low and high level inventory stock.

Keywords: Machine learning, anomaly detection, retail, inventory stock

ÖZET

Perakende şirketleri, envanter stok seviyelerini düzenli olarak izler ve pazar konumlarını korumak için satışlara dayalı tahminlere göre stok seviyelerini yönetir. Envanter stoklarının doğruluğu, perakende şirketlerinin doğru bir strateji oluşturması için kritik öneme sahiptir. Birçok perakende şirketi, çalışan veya müşteri hırsızlığı, hasar veya bozulma, yanlış sevkiyatlar nedeniyle envanter stoğundaki yanlışlıkları tespit etmeye ve önlemeye çalışmaktadır. Makine öğrenmesi yöntemlerini kullanarak hatalı stokları tespit etmeyi amaçlayan çalışmamızda Türkiye'nin en büyük süpermarket zincirlerinden Migros Ticaret A.Ş.'nin gerçek envanter stoğu verileri kullanılmıştır. Anormal stok değerlerinin tespiti için İzolasyon Ormanı, Yerel Aykırı Değer Faktörü, Tek-Sınıf Destek Vektör Makinesi gibi birden fazla makine öğrenimi algoritması uygulanmıştır. Öte yandan, genellikle araştırmacılar yöntem geliştirmek için halka açık verileri kullanır; fakat özellikle denetimsiz öğrenme alanındaki makine öğrenmesi algoritmalarını gerçek hayattaki verilere uygulamak zordur. Bu tezde gerçek hayattaki verilerdeki problemlerin, örneğin verideki eksik ve ekstrem değerlerin vb. nasıl ele alınacağını gösteriyoruz. Deneysel sonuçlar, düşük ve yüksek seviyedeki envanter stoğundaki anormalliklerin tespitinde makine öğrenmesi yöntemlerinin performanslarını göstermektedir.

Anahtar Sözcükler: Makine öğrenmesi, anormallik tespiti, perakende, envanter stoğu

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

A	A point
B	A point
c	The average path length of an isolation tree
d	Distance
d_{reach}	Reachability distance
$\mathscr{D}$	A dataset
E	The expected path length
h	The path length of a datapoint
k	The number of neighbors
l_k	Local reachability density
ln	Natural logarithm
L_k	Local Outlier Factor score
n	Number of dataset
N_k	The k-nearest neighbors of a point
$\mathcal{N}(.,.)$	Normal distribution
p	A split value
Q	The feature
s	An anomaly score
T	The node of an isolation tree
x	A datapoint
X	A dataset
α	Alpha
γ	Gamma
ξ	Xi
Σ	Total
∞	Infinity
$\forall$	For all
$\in$	Element
$\emptyset$	Empty set

LIST OF ACRONYMS AND ABBREVIATIONS

AD	Anomaly Detection
API	Application Programming Interface
AR	AutoRegressive
ARIMA	AutoRegressive Integrated Moving Average
ERP	Enterprise Resource Planning
IF	Isolation Forest
IRI	Inventory Record Inaccuracy
KNN	K-Nearest Neighbor
LOF	Local Outlier Factor
LSTM	Long Short-Term Memory
MCDC	Multi-Channel Distribution Center
ML	Machine Learning
OCSVM	One-Class Support Vector Machine
OHE	One-Hot Encoding
QP	Psychical Stock Quantity
QR	Recorded Inventory Quantity
RFID	Radio Frequency Identification
RNN	Recurrent Neural Network
SHAP	Shapley Additive Explanations
SKU	Stock-Keeping Units
SVM	Support Vector Machine

1. INTRODUCTION

Many retail companies keep their inventory levels under constant control and manage the inventory level according to the estimated future sales to support their market position. Many companies develop a dynamic structure that automatically predicts demand instead of static methods or buying a service or product that automatically predicts the demand. In the retail industry, automatic demand forecasting plays a vital role in inventory management for companies. The correct operation of automated inventory management is based on the company information system that provides accurate stock information [1]. Generally, the company information system's stock data are based on daily sales and shipments calculations. However, the company information system's stock values may differ from the actual stock in the store and warehouse. If the stock in the system is lower than the existing stock, it may lead to excessive products, resulting in extremely high inventory levels [2]. On the contrary, the order of products may be delayed, and the company cannot satisfy customer demand. In either case, the faulty stock automated system cannot operate to its full potential, resulting in a loss of profit. Inventory record inaccuracy (IRI), the name given to errors in stocks in the literature, can be caused by manual adjustments, theft, damage, and wrong shipments. IRI causes 1% sales and 3% gross profit loss [3] in the retail industry.

Considering that there are millions of stock-keeping units (SKU) in the retail industry, we can easily express that it is challenging to ensure stock accuracy and follow it. Many retail companies implement Enterprise Resource Planning (ERP) systems [4] to ensure consistency of inventory stock. To prevent wrong inventory stocks, the human factor decreases day by day, but mistakes can be made in accepting goods or sending goods from one store to other stores and ERP cannot handle these errors. These directly affect the inventory accuracy in the system. Machine learning is one

of the methods that can apply to detect these errors in the inventory stock.

1.1 Inventory Management

Inventory is defined as a stock or store of goods in [5]. Companies generally stock hundreds or even thousands of items in their inventory, from little products like pencils, glasses, string, and buttons to major items like machinery, cranes, construction equipment, and trucks. The majority of things in a company's inventory are, of course, tied to the sort of business it does. As a result, manufacturing companies provide raw materials, buy components, semi-finished products, and completed goods. Fresh and canned foods, packaged and frozen meals, home supplies, periodicals, baked goods, dairy, fruit, and other items are all available in the inventory of supermarkets. We can categorize inventory into six groups: raw materials, semi-finished products, finished goods, equipment and supplies, maintenance and repairs, and goods and services in transit [5].

Inventory management is an essential aspect of operations management [5]. Most organizations and supply chains rely on effective inventory management, influencing operations, marketing, and finances. On the other hand, poor inventory management stymies operations, lowers customer satisfaction, and raises operational expenses. Some businesses have outstanding inventory management, while others have adequate inventory management. Many, on the other hand, have poor inventory management. They have insufficient or excessive inventory, poor inventory tracking, or misplaced priorities. What is missing is a clear picture of what needs to be done and how it should be done.

Inventory control issues can result in both understocking and overstocking of products. Late deliveries, lost sales, customer complaints, and production bottlenecks arise from understocking; overstocking wastes space and money that could be better spent elsewhere. Although excessive overstocking may appear to be the lesser of two evils, the cost of excessive overstocking can be staggering when inventory holding

costs are high, and things can easily spiral out of control. The overall purpose of inventory management is to deliver exceptional customer service while keeping inventory costs within reasonable bounds. When it comes to inventory management, the two most essential considerations (decisions) are when to order and how much to order.

It is indicated in [5] that there are five requirements for effective inventory management:

- A way to keep track of what's in stock and what's on order.
- A dependable demand projection that includes a cautionary note regarding possible forecast errors.
- Information on delivery timeframes and variability in delivery time.
- Inventory holding, ordering, and shortfall costs are all plausible estimates.
- A classification system for items in stock.

The first requirement is crucial since it assures that the others exist. The accurate stock can aid in creating accurate forecasts at the right time. We focus on how to increase the accuracy of inventory. The overview of the inventory management process is shown in Figure 1.1.

Figure 1.1 Inventory management stages [6]

1.2 Anomaly Detection

Machine learning usage areas are increasing [7]. One of these usage areas is anomaly detection. In machine learning, the problems of detecting abnormal values in data are processed as anomaly detection (AD). AD means that a situation or formation is different from its natural flow. In other words, it is a significant dissociation of a point itself from its past value or predicted future value. These classes are usually highly unequally distributed, with the normal class dominating the anomaly one. However, recognizing anomalies can provide helpful information about the application inside. Some examples are as follows [8]:

- Intrusion detection systems: Computers gather a large amount of data (information) regarding system calls, logs, network traffic, and various other user actions. Because of malevolent activities or prohibited instances, data might sometimes indicate strange patterns. Intrusion detection is the process of detecting these actions.
- Credit-card fraud: Fraud detection is one of the most critical financial system cases. Credit card usage shows different patterns in many fraud cases, such as buying costly items from different locations. Anomaly detection techniques can detect such patterns.
- Interesting sensor events: Sensors are used in many real applications to monitor the environment for the detected incident.
- Medical diagnosis: Information about patients is gathered via a variety of tests and scans in numerous medical applications. Unusual patterns in data usually indicate diseases.
- Time-series monitoring: Variables change over time, but sometimes they increase or decrease suddenly. These events can provide important information.
- Law enforcement: Anomaly detection has a variety of uses in law enforcement, particularly when unexpected patterns may be identified over time with several actions by an asset. Trading operations frequently need to recognize unexpected patterns in data to detect fraud in financial transactions.

- Earth science: Many systems capture a large quantity of data regarding weather patterns and climate changes. Those data reveal a unique pattern of human activity trends and environmental changes.

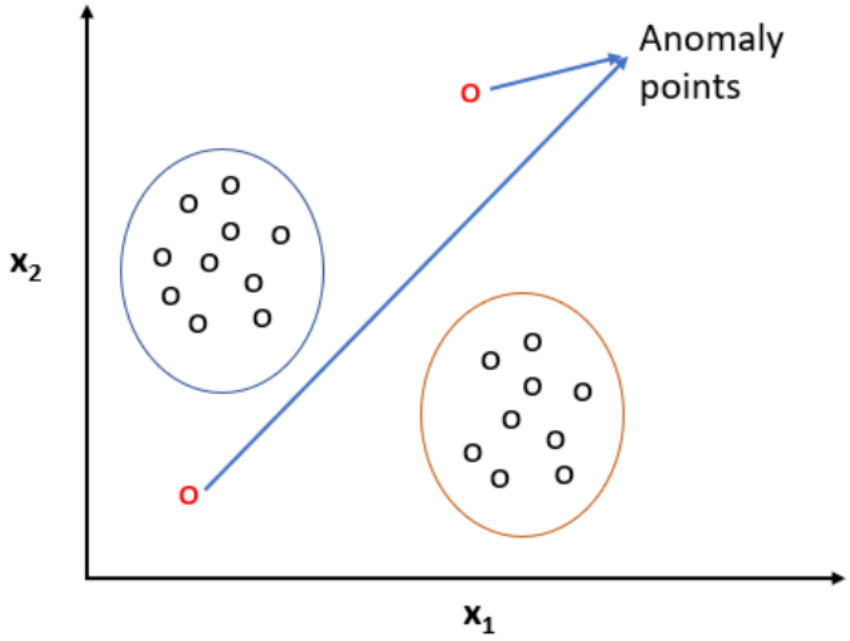

Figure 1.2 An example of anomaly points

In these applications, the data is located in a typical state. It is referred to as the normal data model, and for unusual circumstances, there is abnormal data. Normal data is also known as inlier, and abnormal data is called as outlier. An example of anomaly points are shown in Figure 1.2. There are many algorithms for detecting anomalies in the data [9]. The output of these algorithms can be of two types:

- Anomaly score: Most anomaly detection algorithms output a score for each data point. This score provides the level of the anomaly. The most suspicious points can be shown by ranking the data points. Also, the analyst can decide on a threshold to flag anomaly points.
- Binary label: Some algorithms give binary labels that indicate whether or not a data point is an outlier. Although some algorithms return binary labels directly, outlier scores can also be transformed to binary labels.

When comparing anomaly score and binary label, we can easily say that anomaly

score has more advantages than binary because we can order score and define a threshold to convert from anomaly score to binary label. Especially ordering is critical for the business unit to analyze most suspicious events.

Anomaly detection methods have benefited from machine learning in recent years, especially from deep learning. In this context, studies on anomaly detection are generally carried out by applying supervised learning algorithms on labelled data. However, obtaining labelled data in many applications is very time-consuming and costly. In addition, each different situation for labelling may require a separate area of expertise. Therefore, supervised learning algorithms are limited by the availability of labelled data.

1.3 Why Is Unsupervised Anomaly Detection Used?

Labelled data are scarce, researchers are working to develop unsupervised learning models to apply them to problems and tasks that have previously been disregarded. However, unsupervised learning is still a complex topic to tackle, as it consistently under-performs supervised learning in various tasks.

Supervised learning allows you to produce an output with results from previous experiences. But unsupervised learning can reveal unknown patterns in the data, allowing you to obtain a unique result that has not been experienced before. In other words, while output variables are given along with the input in supervised learning, in unsupervised learning it is desired to produce an output that can solve the problem by only giving the input data. In this way, supervised machine learning helps solve a variety of real-world computational problems.

When we look at people's diseases, it is easy to deal with diseases that have a cure, such as flu and malaria, and to compare them with each other. However, SarS-CoV-2 (CoViD-19) is an anomaly that shows features other than other diseases. Had this anomaly been detected at an early stage, its spread could have been contained and would not have led to a pandemic. A supervised learning procedure would fail

to detect it as an anomaly, as it is a new anomaly that has not been seen before. Because the supervised learning model learns patterns only from the labeled data in the current dataset. But an unsupervised learning algorithm could detect this virus as an anomaly, as it would not match data from pre-existing diseases.

Finally, it is worth noting that the human reaction to unanticipated world discoveries, precisely the human form of realizing anomaly detection, is mainly unsupervised. Here is a simple example inspired by another unsupervised learning example given in [10]. In the early days of life, a newborn is typically perplexed by the environment he senses. On the other hand, a newborn can create his sense of normalcy, that is, learn and become aware of it, after experiencing the new world for a while and gathering some observations of the surroundings without any monitoring. He is frequently disturbed when he sees an unfamiliar face since he does not recognize it.

The idea underlying this simple example underpins the principles of unsupervised anomaly detection, which will be the primary emphasis of this thesis. Because the dataset that inspired this thesis is fully unlabeled, this thesis aims to improve unsupervised anomaly detection.

1.4 Motivation & Purposes

In the previous sections, we focused on the importance of inventory management. Accurate inventory stocks are essential for the customer experience and the company expenses. In this framework, when we show the inventory stocks numerically, we can analyze the trend. We can say that the errors in the inventory are an anomaly. To find these errors, we can use anomaly detection methods via machine learning.

This study aims to detect the anomalies before they are reflected on the store by using the real inventory stock data of one of Turkey's largest supermarket chains with outlier detection methods based on machine learning. Inventory record inaccuracy is a big problem in the retail sector [1], and detecting the anomaly before it occurs and generating an alarm in case of a mistake can prevent this error before it happens.

The points we aim to contribute to the literature in this study are as follows:

- Detect anomalies in data whose characteristics change over time; because people's consumption habits can change or the company's strategy can change.
- Reduce the number of false alarms in the anomaly detection system; companies have limited sources to investigate the alarm.
- Establishing a dynamic anomaly detection structure:
  - Adding a new feature should be easy.
  - Extracting the current features should be easy.
  - Defining the thresholds should be dynamic.

1.5 Thesis Outline

The rest of this thesis is organized as follows:

- Section 2 presents the background of inventory record inaccuracy (IRI) and some unsupervised anomaly detection methods.
- Section 3 provides a theoretical background for the unsupervised anomaly detection methods.
- Section 4 explain our methodology in seven parts; they are data gathering, data preprocessing, feature engineering, feature selection, application of models, SHAP analysis and interpretation of results.
- Section 5 describes our experimental results of comparing the models.
- Section 6 explains the findings in the thesis and makes recommendations for further research.

2. LITERATURE REVIEW

In this section, the relevant literature has been reviewed in two stages. Firstly, the studies carried out to ensure stock accuracy based on inventory management are examined. Then, unsupervised anomaly detection methods and the studies they used are explained.

2.1 Inventory Record Inaccuracy

Inventory management processes are among the most significant key factors for the inventory carrying companies aiming minimum operational cost while providing high-quality service with maximum product availability. Therefore, the information gathered from automated inventory management information systems is crucial for a successful business. However, while the correct data supports the right decisions, incorrect ones can lead to revenue losses due to out-of-stock cases arising from Inventory Record Inaccuracy (IRI) [11]. The presence, effects, causes, reduction and measurement methods of IRI consists of the five main classifications regarding studies of IRI.

The presence of IRI in companies, regardless of having enterprise resource planning (ERP) systems or not, is identified via experimental proofs from several papers which define the difference between QR (recorded inventory quantity) and QP (psychical stock quantity) [3]. For instance, [11] disprove the popular opinion that retailers are good at knowing the number of products they actually have in their stores while working with distinguished sponsor companies of Auto-ID Center at MIT. One of these global retail companies' stores is better at perfect inventory accuracy rather than other stores; there is a maximum 80% match between QR and QP. In contrast, two-third of inventory records were inaccurate. It is also demonstrated in [12] that

similar results with the study on systematic variation in IRI. By observing 37 stores belonging to one retailer, it is discovered that only 35% of the 370,000 inventory records are matched with the quantity at the store and do not show IRI problems. In contrast, 65% of leading retail chain records are inaccurate.

Customer service level can also be seriously affected by IRI, and even at a low level [13]. It is also found in [13] that IRI can risk supply chain stability due to their modelling, covering a numerical simulation that supposes varied mistakes. It is also refered in [14] that damaging effects of IRI on operating performance by observing the daily variation of IRI that is indicated by daily collected data from discrete-event simulation experiments to a multi-spectral retailer. It is revealed that IRI declines service levels, so implementations based on ignorance of daily IRI variation need to be revised to multi-day counting for assessing daily IRI and identifying its reason. It is argued in [15] that MCDC (multi-channel distribution center) is highly sensitive about IRI. The inventory system's stability and convenience can be negatively affected by IRI if it is at substantial levels and has a changing level of consistency. Product availability can suffer from a low level of IRI, which can lead to a reputational loss for retailers.

Some of the causes of IRI are described in [16] as backroom and shelf shrinkage as well as part-time labour with a lack of feedback loop and unsuccessful in reducing IRI. Theft, damage and poor quality are also considered as factors besides transactions [12]. Based on the dependency of transactions, IRI can have permanent or temporary reasons.

The frequency of audit, the quantity of annual selling, cost of the product, volume of monetary, variety of products, mode of distribution, physical distribution model, the density of inventory are also casual agents of IRI [12]. It is stated in [17] that increased product variety and inventory levels result in high deformity rates. Employee related issues like turnover, training, workload, the effort also consists of drivers.

In order to reduce IRI, studies provide different management options. The replenishment policy development is highlighted in [18] as the design of better cycle count. It is refered in [19] that RFID (Radio Frequency Identification) technology as an opportunity to monitor continuously via RFID readers.

The Bayesian Inventory Record, which is introduced in [12] and used to estimate the probability distribution of the inventory in the presence of stochastic IRI triggers. The distribution is utilized to create a strict replenishment rule in an operational scenario. One of the essential aspects of the computations is replenishment, which captures the fluxes of the goods in the same manner that NRI does in the proposed IRI measure.

2.2 Unsupervised Anomaly Detection Methods

Anomaly detection is assessed as supervised, semi-supervised, or unsupervised, supported by whether the labels are used within the training process [20]. We concentrate on unsupervised learning algorithms, which are mainly grouped into classical anomaly detection and deep anomaly detection methods. The overview of anomaly detection methods can be found in Figure 2.1.

2.2.1 Classical anomaly detection methods

Classical methods use linear calculation, and some of them are distance-based, density-based, angle-based methods. These methods are based on statistical and similarity calculation; therefore, the training phase takes too much time when data is increasing.

Density-based methods focus on data points in the specific region of space. Density-based methods are significantly related to clustering, and distance-based methods use the distances specific region, and we can say these regions are similar to clusters. Density-based methods detect locality-sensitive outliers.

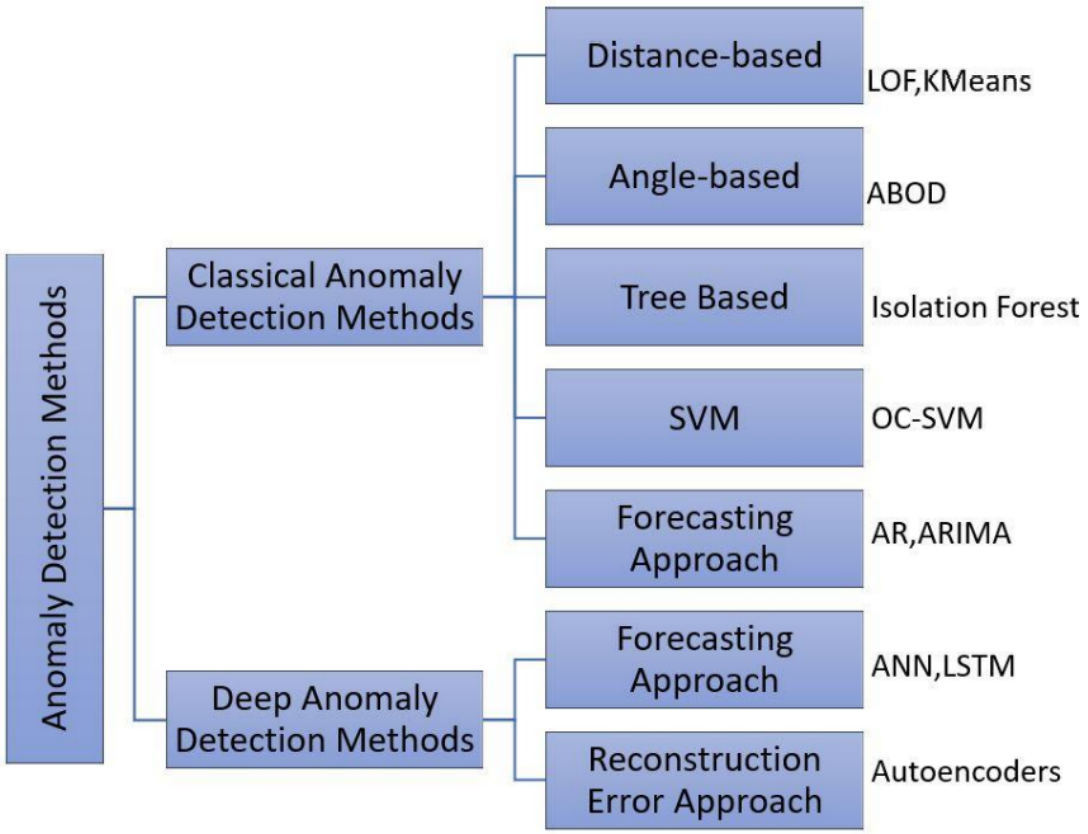

Figure 2.1 The overview of anomaly detection methods

The one popular density-based method is the Local Outlier Factor (LOF) [21]. LOF can measure data points' patency to adjust variations of different local densities. It considers that samples whose density is significantly lower than their neighbours are an outlier. LOF calculates the score for each data point by computing the average densities of the neighbours to the density of the point itself. The LOF methods to discover data outliers, data inaccuracies and traffic irregularities in real-world scenarios such as accidents, traffic jams, and low volume are used in [21].

An example of LOF outlier scores is shown Figure 2.2. In Figure 2.2, there are two clusters in the upper right and lower left, and the points that diverge from these clusters are determined as outliers. LOF outlier scores are circled.

Another density-based method is clustering. There is a very strong relationship between clustering and anomaly detection. There may be some outlier values as well as points that are similar to each other. Clustering aims to divide a large

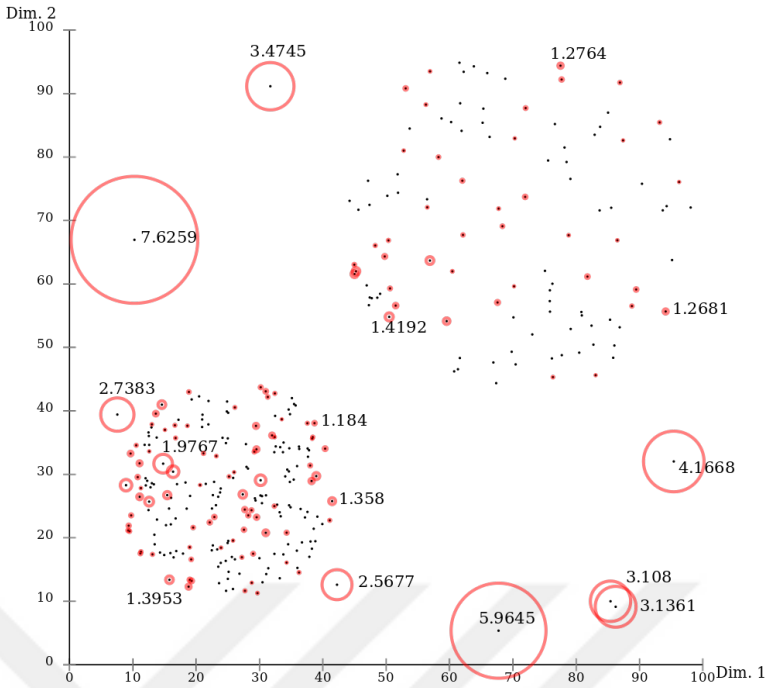

Figure 2.2 LOF outlier scores example [22]

data collection or sample into subgroups based on point similarity and evaluate them individually. Also, anomaly detection aims to find not similar points with other points. Generally, many algorithms defined outlier values as a side-product in their outputs. However, it is essential to understand why these points are outlier and how they are similar to their clusters. To find the reason for the anomaly, we have to find outlier scores that give the points different from other points [23]. The simple definition of clustering form is to find the distance between points and cluster centroids. Generally, there is more than one cluster, and we have to find the distance between points and their closest cluster. To understand cluster-based techniques, we can focus on the basic clustering algorithm that is K-means. K-means separate the data to a fixed number (k) of clusters. Each cluster is a group of data points grouped by their similarity and have a centroid, and all points in the clusters are close to their cluster centroids. Each point is assigned to the closest cluster centroid, and the process aims to keep the centroids small. K-means are used in [24] to detect intrusions, and simulations show that this approach effectively detects unknown intrusions in real-world network connections.

Another popular method is Isolation Forest (IF) is similar to decision tree algo-

rithms. The algorithm aims to isolate the anomaly values by randomly selecting an attribute from the dataset and then randomly splitting the value between the minimum and maximum values. The randomly partitioning of the attribute proves a shorter path for the anomaly points that will be separated from normal data. Since they scale well with huge datasets and offer rapid prediction speeds. Furthermore, they operate well with various features, such as discrete and continuous, so the features do not need to be normalized. The disadvantage of tree-based methods is that they are susceptible to variables in data with a small number of samples. For example, they can mark all special days as anomalies. There are many applications of the Isolation Forest, such as network traffic anomaly [25], credit card fraud detection [26] and monitoring machines [27].

Another popular classical method is a one-class support vector machines (OCSVM) that differentiate one class of observations from another by using hyper-planes in multidimensional space [28]. One-class support machines are linear and logistic regression variants with margins used to avoid overfitting, like regularization in regression models. In addition, the sum of squared errors can be employed as a loss function in SVMs. An SVM model attempts to segregate data by as large a separation as feasible while penalizing incorrect predictions on the wrong side of the gap. The SVM model then forecasts by dividing points to one side of the gap.

Another classical approach uses time series forecasting methods such as AutoRegressive (AR) and AutoRegressive Integrated Moving Average (ARIMA) to predict and calculate the difference between actual and predicted values. If the difference is high, the point has the potential for anomalies [29].

2.2.2 Deep anomaly detection methods

We can divide deep learning anomaly detection models into two groups: forecasting and using the reconstruction error of input values [20]. In forecasting models, the aim is to train the model with past values and make predictions for the future

using this model. The difference between the estimates and the actual values gives us information about the degree of abnormality of the points. Recurrent neural network (RNN) and long short-term memory (LSTM) are very popular for sequence prediction. In [30], authors use RNN based model to detect anomalies in multivariate time series data to prevent cyber-attacks by minimizing the mean squared error between actual and predictions. In [31], they use LSTM to detect anomalies in manufacturing processes by using advantages of long-term effects of processes. Also, [32] combines LSTM and OCSVM to increase the learning performance of LSTM because the training of deep learning forecasting models takes time and memory. The reconstruction model focuses on several approaches for lower reconstruction error. For example, autoencoders are frequently used for anomaly detection by learning to rebuild a given input. The model is only trained on normal data. When it cannot rebuild the input with the same correctness as ordinary data reconstruction, the input sequence is labelled anomalous data [20].

LSTM autoencoder models can reveal anomalies using long-term effects [33]. Variational autoencoders are autoencoders representing the relationship between two random variables, latent variable z and visible variable x. A prior for z is typically a multivariate unit $\mathcal{N}(0; 1)$. VAE learns the distribution of variables, and this feature provides a dynamic structure for different variables. In [34], the authors define the reconstruction probability for anomaly detection as the average probability of the original data provided by the distribution. Anomalies are data points with more significant reconstruction possibilities and vice versa.

We prepared this study to reduce the Inventory Recording Inaccuracy (IRI) highlighted in [11] in Migros Ticaret A.Ş. inventory. In this thesis, we identified the differences between amount of psychic stock and amount of recorded inventory on retail company inventory records, similar to the studies in [11] and [12]. The historical stock averages of the SKUs were used to detect inconsistencies in the inventory stock. In our study, we embraced the idea in [13] that IRI could compromise supply chain stability. In addition to the current stock amount, we also examined the stock

movement amounts and included them in the model. [14] examined diurnal variation of IRI. In the data set in this study, we did not include sequential daily stock data. We created a model that compares the current stock data of a day with its historical average and detects anomalies. By scheduling this model, we have set it up to run again every day. As noted in [15], IRI has a negative impact on distribution centers. While confirming the model outputs with the store managers, we found that the mistakes made in the distribution centers seriously affect the store stocks. It is demonstrated in [17] that increased product diversity affects IRI. In our study, we found that stock quantities were wrong in SKUs with similar names. It is stated in [19] that continuous monitoring is possible with an RFID reader. However, in the errors we encountered in the process of preparing this study, we also encountered IRI originating from the RFID reader. In [20], it was stated that anomaly detection studies were classified according to whether or not labels were included in the training. In this study, we applied the Unsupervised Learning method because we used unlabeled data. Unsupervised anomaly detection methods are also classified within themselves. In this thesis, we used distance-based LOF, tree-based IF, and OCSVM that is a type of SVM, which are in the classical anomaly detection methods group.

In [21], the LOF method is used in real-world scenarios. Similarly, we used the LOF method on real-life data in our study. In [24], an intrusion detection study was conducted with real-life scenarios and K-means is used as anomaly detection method. We did not use the K-means method in our study. Because, just like K-means, distance-based LOF gave successful results in our data. In [25], [26] and [27], IF, another method we used in our study, was used. However, the aim of our study is different from those in those studies. Network intrusions were used in [25], [26] a fraud study on credit cards, and monitoring machines in [27]. The data in our study belong to the retail sector and we aimed to detect stock anomalies in the inventory.

In addition, we examined deep anomaly detection methods, but we did not use them in our study. In the future, a stock estimation mechanism can be created

using autoencoders to improve this work. With this thesis, we included OCSVM, another classical anomaly detection method apart from IF and LOF, into our model in order to solve the current business problem.

3. THEORETICAL BACKGROUND

3.1 Isolation Forest (IF)

Isolation Forest (IF) is an unsupervised machine learning algorithm that is used to detect outliers or anomalies in a dataset. IF differs from other algorithms in that it tries to isolate abnormal data points from the beginning. Anomalies are points that deviate from other points, and also there are fewer outliers than inliers, therefore they are more susceptible to isolation.

A tree structure called Isolation Tree or iTree is used to isolate outliers. Anomalies are isolated closer to the roots of the tree. Because they are sensitive to isolation. Normal points other than anomalies are isolated at the deep ends of the tree. This characteristic is the basis of IF's anomaly detection [35]. An example of iForest with multiple iTrees is shown in Figure 3.1.

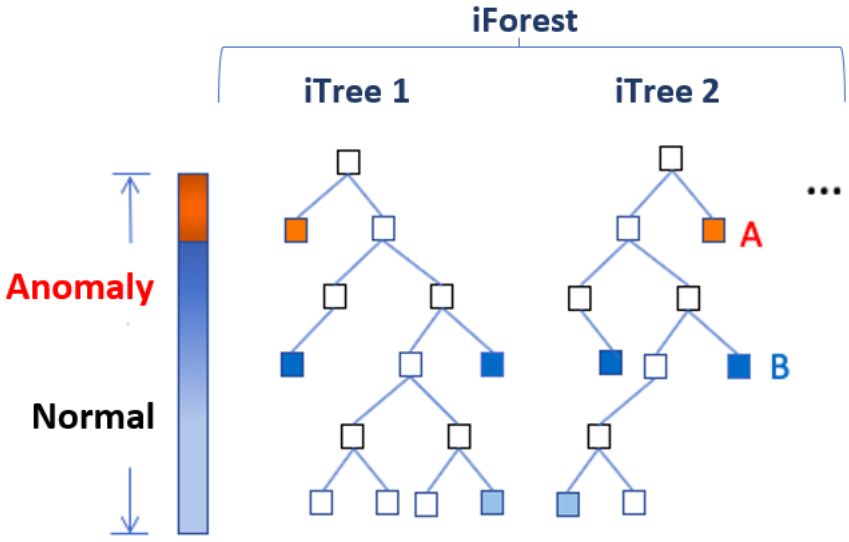

Figure 3.1 Isolation Forest (iForest) structure

In a proper binary tree (iTree), each node has exactly zero or two additional nodes. T is assumed to be the node of an isolation tree. Internal nodes have exactly two

daughter nodes with one test; T_L and T_R. External nodes do not have additional nodes.

Suppose there is a dataset X of n instances from a d-variate distribution and IF is used on it, then X is recursively divided by randomly selecting the feature $Q_i \in Q_1$, ..., Q_d with equal probability from the set of features and a split value p which gives an isolation tree. After the feature Q_i is chosen, the value belonging to this specific feature $X(Q_i)$ is compared with the split value p for every datapoint. If $X(Q_i) < p$, the datapoint will go to T_L and otherwise it will go to T_R. This is done until the tree reaches a height limit[1] or there is only one datapoint from the set X left or all data in X have the same values. This is the first stage of the model and it is called the training stage. In this stage, the isolation trees are constructed from a sub-sample of the data. In Figure 3.2, an example of an isolation tree created from a small dataset can be seen.

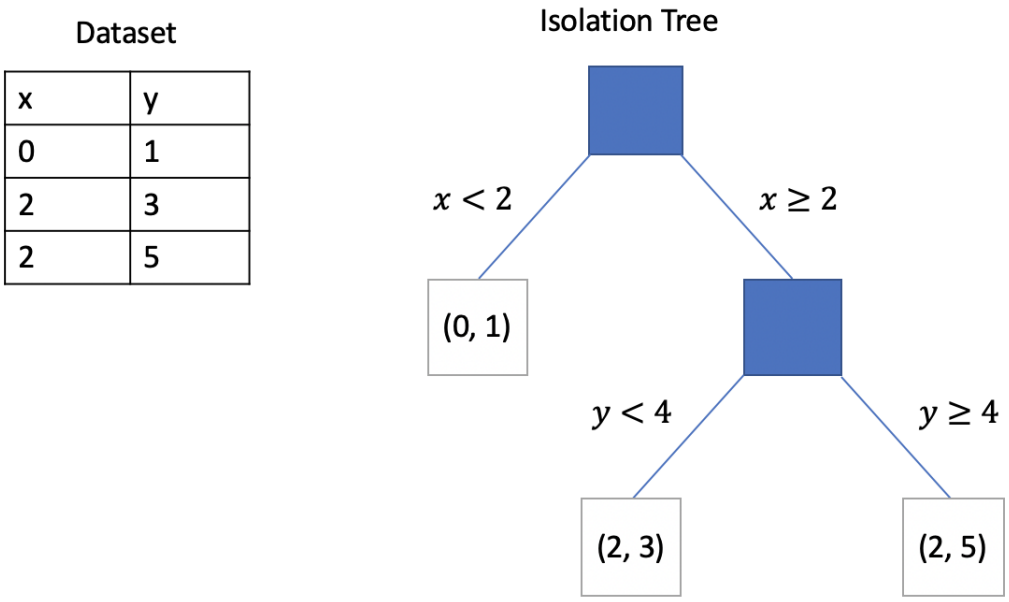

Figure 3.2 An example of an isolation tree [35].

In isolation trees, instances are partitioned recursively until all of them are isolated. Anomalies are isolated earlier in the trees than the normal instances, because of their

[1]The trees are cut off at the pre-set height limit to reduce the computation time of the IF algorithm.

distinguishable attribute-values. To illustrate that outliers are more susceptible to isolation than inliers, an example is given in Figure 3.3.

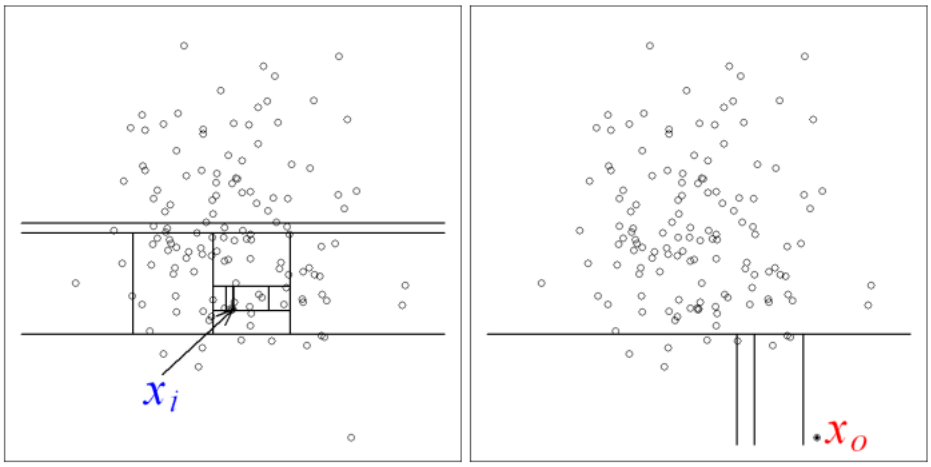

Figure 3.3 In a 135-points Gaussian distribution, 12 random segments are required to be isolated from a normal point x_i, while 4 segments are required to isolate from an anomaly x_o [35].

The second stage is the evaluation stage, where an anomaly score s is derived from the expected path length $E(h(x))$ for each instance. The path length $h(x)$ of a datapoint x is represented by the number of edges from the root node to a terminating node as this point x passes through the isolation tree. The expected path length is derived by passing all the datapoints through each isolation tree in the isolation forest. It is then the average value of $h(x)$ from all the isolation trees that were built. However, the trees have a height limit, and thus, it can happen that the tree is not fully grown. These are early terminated nodes, which means that these nodes will contain more than one datapoint. If this is the case, an extra constant $c(n)$ is added to the path length of the instance in the early terminated node. This $c(n)$ is the average path length of an isolation tree that is built with n datapoints. Finally, the anomaly score of a datapoint x for a dataset of size n is given by

$$s(x, n) = 2^{\frac{-E(h(x))}{c(n)}}, \tag{3.1}$$

with $E(h(x))$ being the average value of $h(x)$ from all the isolation trees that were built. We see the following:

$$As\ E(h(x)) \to c(n) : s(x, n) = 2^{-\frac{c(n)}{c(n)}} = 2^{-1} = 0.5, \tag{3.2}$$

$$As\ E(h(x)) \to 0 : s(x,n) = 2^{-\frac{0}{c(n)}} = 2^0 = 1, \tag{3.3}$$

$$As\ E(h(x)) \to n-1 : s(x,n) = 2^{-\frac{n-1}{c(n)}} = 2^{-\frac{1}{2} \cdot \frac{n-1}{\ln n - 1 + 0.5772 - 1}} \to 0\ as\ n \to \infty\,. \tag{3.4}$$

This score s can be used to identify outliers. If $s(x)$ is close to 1, then x is an outlier. If $s(x)$ is much smaller than 0.5, then x is normal observation. If $s(x)$ is around 0.5 for all instances, then the set of instances does not contain clear outliers.

It is clear that anomalies will be isolated closer to the root of the tree than inliers. IF uses multiple isolation trees for a given dataset to isolate the anomalies. The input of this method only consists of two variables, namely, the number of trees to build and the sub-sampling size. The sub-sampling size controls the training size of the algorithm. A sample of the overall data is taken randomly and used to construct an isolation tree. Two major advantages of this method are that the detection performance converges quickly with a very small number of trees and it only requires a small sub-sampling size to achieve high detection performance with high efficiency. Better isolation trees are built from small sample sizes, because the swamping and masking effects are reduced. Swamping happens when the method wrongly labels normal instances as outliers. Masking is the case if there are too many outliers. Another characteristic from IF is that it does not need additional measures to detect the outliers. This reduces the computational cost. It can also easily handle large data sizes. Moreover, it can be explained why a datapoint is an outlier in a certain dataset by looking at the specific paths in the isolation trees and the different features used at every split.

3.2 Local Outlier Factor (LOF)

This approach is suggested in [21] to find outliers in a dataset. Local Outlier Factor (LOF) is an unsupervised machine learning algorithm that tells us the degree of abnormality of a point (observation). LOF is one of the best known algorithms and is widely used in anomaly detection.

The LOF algorithm is defined in [21] by using density-based methods. LOF is

actually similar to the K-Nearest Neighbor (KNN) classification algorithm. Because it carries the idea of nearest neighbors when determining the score of outliers or anomalies. The common points of the two algorithms are that the distances to the neighbors are taken as a basis when making judgments. Also, there is a big difference between the two algorithms. KNN tries to find out who the point of interest looks like. On the other hand, it is tried to find which point is not similar to its neighbors in LOF. The k value here indicates how many neighbors will be judged. So it is a hyper parameter. If we set the k value too small, the algorithm becomes susceptible to noise. For large k values, it may miss local anomalies. To understand LOF, we have to learn a few concepts sequentially:

- k-distance
- Reachability distance (RD)
- Local reachability density (LRD)
- Local Outlier Factor (LOF)

k-Distance

The distance of a point A to its neighbors is calculated according to the selected k number, and these distances are ordered from smallest to largest. The distance in the kth order gives us the k-distance. So it is the kth nearest neighbor. Let the set of k nearest neighbors of A be denoted by $\mathcal{N}_k(A) = \{B \in \mathscr{D} \text{ - } \{A\} : d(A, B) \leq d_k(A)\}$. If there are n points (observations), since the distances of all points will be calculated, a total of $n(n-1)/2$ distances will be calculated. For example, for a data set of 150 observations, 11,175 distance calculations are required. Figure 3.4 shows k-distance (k=3) for point A.

Reachability Distance (RD)

k-distance is used when calculating the reachability distance. This k-distance specifies the distance between two points. Normally, there is already a distance (like Euclidean) between two points. There is also a k-distance between the two points. Reachability distance is the farther of these two distances between two points.

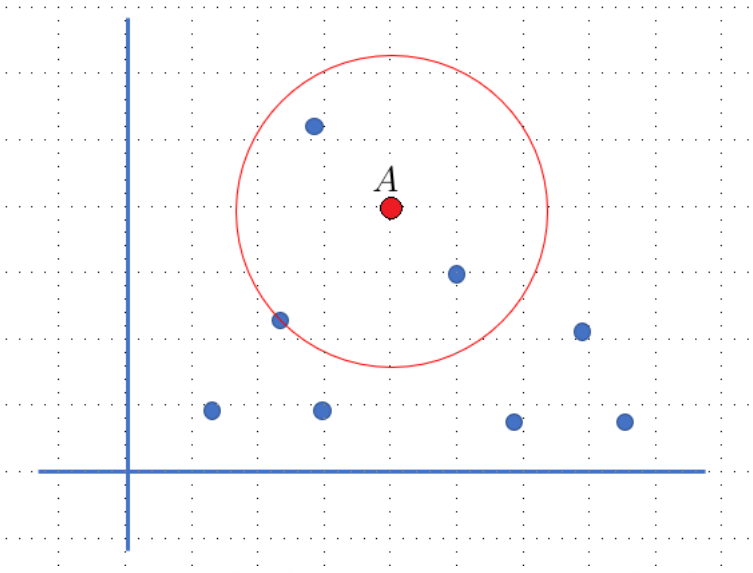

Figure 3.4 k-distance $\rightarrow d_k(A)$

RD is formulated as follows:

$$d_{reach}(A, B) = \max\{d_k(B), d(A, B)\}. \tag{3.5}$$

The average reachability distance of A is

$$\overline{d}_{reach}(A) = \frac{\sum_{B \in \mathcal{N}(\mathcal{A})} d_{reach}(A, B)}{\mid N_k(A) \mid}. \tag{3.6}$$

Local Reachability Density (LRD)

LRD of a point is reciprocal of reachability distance, i.e.,

$$l_k(A) = [\overline{d}_{reach}(A)]^{-1}. \tag{3.7}$$

LRD for point A is shown in Figure 3.5.

Local Outlier Factor (LOF)

The resulting LRD is compared with LRDs of all points in $N_k(A)$, and the ratio is defined as LOF:

$$L_k(A) = [\frac{\sum_{o \in N_k(A)} \frac{l_k(o)}{l_k(A)}}{\mid N_k(A) \mid}]. \tag{3.8}$$

LOF generates a score for each point. If points have large LOF values, they are determined as outliers.

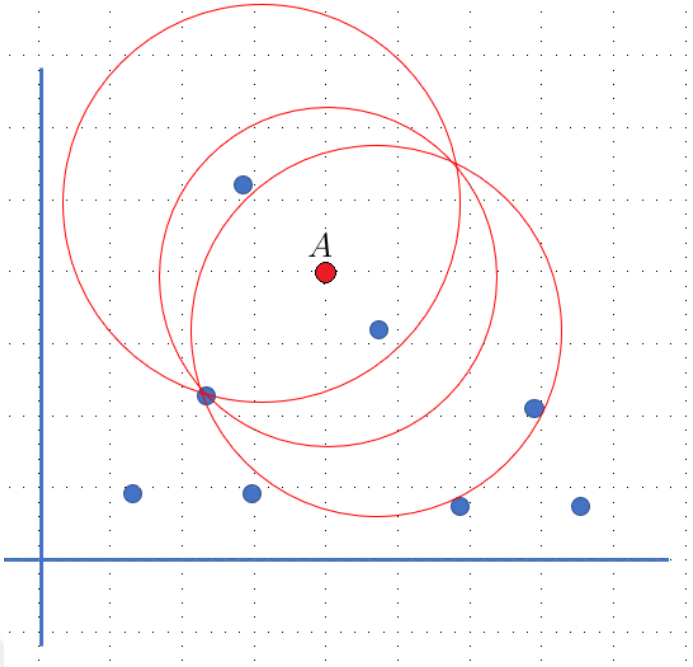

Figure 3.5 Local Reachability Density (LRD)

To account for k, the result is determined as follows: $L_k(A)$ is calculated for the selected k values in the pre-specified range and max $L_k(A)$ is kept. A point with a large LOF value is determined as anomaly [36]. A detailed description of LOF Computation is shown in Figure 3.6.

3.3 One-Class Support Vector Machine (OCSVM)

The One-Class Support Vector Machine (OCSVM) is a specially designed variant of SVM that is an unsupervised machine learning model used for outlier or anomaly detection. OCSVM is different from the generaly used the regular supervised SVM algorithm. In the OCSVM, there are no target values (labels) for model's training. However, by learning the limits of the inliers (normal values), it determines the values outside this limit as outliers [37]. An example of OCSVM outlier detection is shown in Figure 3.7.

The SVM is proposed in [37] that maps the input vector in the high-dimensional feature space. Then, the decision boundary or separation hyperplane determined by the support vectors is obtained. The negative values indicate outliers, and positive

Require: $k, \mathcal{D}$.
Ensure: L_k - LOF score for each object in $\mathcal{D}$

$L_k = \emptyset$.
for $A \in \mathcal{D}$ **do**
  $N_k(A) = NULL$
  for $B \in \mathcal{D}$ **do**
    if $|N_k(A)| < k$ **then**
      Add B in $N_k(A)$
    else
      Let $s* \in N_k(A)$ be such that $dist(A, s*) \geq dist(A, s)$ for all $s \in N_k(A)$;
      if $dist(A, s*) > dist(A,B)$ **then**
        Replace $s* \in N_k(A)$ by B
      end if
    end if
  end for
  $d_k(A)$=max$\{dist(A, s)|s \in N_k(A)\}$
end for
for $A \in \mathcal{D}$ **do**
  for $B \in \mathcal{D}$ **do**
    $d_{reach}(A,B) = \max\{d_k(A), d(A,B)\}$
  end for
end for
for $A \in \mathcal{D}$ **do**
  $l_k(A) = \frac{|N_k(A)|}{\sum_{B \in N_k(A)} d_{reach}(A,B)}$
end for
for $A \in \mathcal{D}$ **do**
  $L_k(A) = [\frac{\sum_{o \in N_k(A)} \frac{l_k(o)}{l_k(A)}}{|N_k(A)|}]$
end for
return L_k

Figure 3.6 Algorithm of LOF Computation [36]

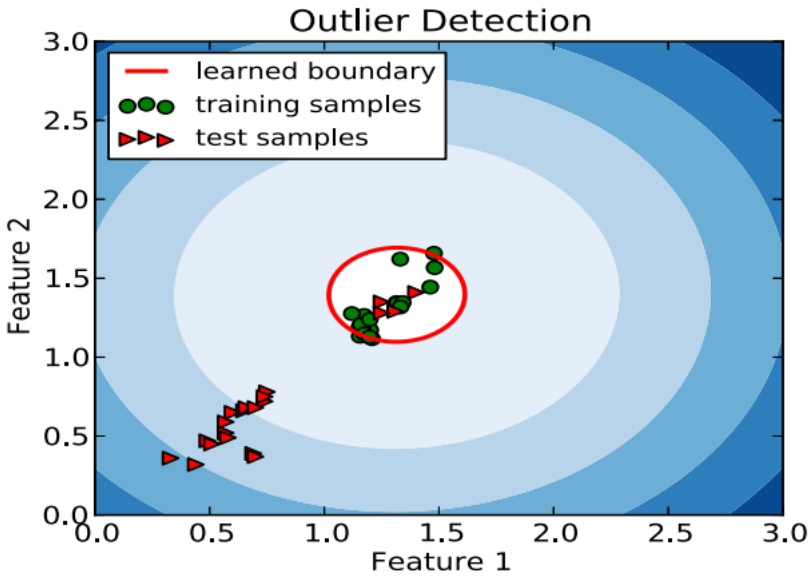

Figure 3.7 OCSVM boundary and outlier detection [38]

values indicate normal data. OCSVM learns a hyperplane in a breeding kernel Hilbert space to separate data points from the origin. It maximizes the distance from the origin to the hyperplane. Data points far from the origin are positively labeled. The origin is labeled as negative [39]. The data are separated into normal and abnormal parts by the hyperplane. Those that deviate abnormally from normal data pattern are determined as abnormal data [37].

OCSVMs have two formulations. The first formulation uses an n-dimensional plane or hyperplane for the decision boundary. The second one uses hypersphere instead of hyperplane for decision boundary. When we examine the second formulation we see that, the hypersphere has a center named $\mathbf{a}$ and its radius is $R > 0$. Let's consider n as the number of data points given by x_i ; $i = 1, 2, ..., n$.

Euclidean distance from hypersphere center is expressed as $\mid x_i - \mathbf{a} \mid$. Also, the Euclidean distance to a given data point is the same. It is desirable to minimize the R^2 cost function. Each point locates on or within the hypersphere. The constraint is $\mid x_i - \mathbf{a} \mid^2 \leq R^2 \; \forall i$.

In this way, anomalies greatly affect tuning. Let's change the cost function to $R^2 + C \sum_i \xi_i$. Then, let's change the constraint to $\mid x_i - \mathbf{a} \mid^2 \leq R^2 + \xi_i \; \forall i$.

Here, ξ_i represent positive weights and are associated with each data point. The higher its value, the less it will affect the setting of the data point R. C acts as a trade-off between classification and volume errors.

If we combine this with the Lagrange Multipliers method, one obtains the cost function as

$$\mathcal{L}(R, \mathbf{a}, \alpha_i, \gamma_i, \xi_i) = R^2 + C \sum_i \xi_i - \sum_i \alpha_i (R^2 + \xi_i - (|\mathbf{x}_i|^2 - 2\mathbf{a} \cdot \mathbf{x}_i + |\mathbf{a}|^2)) - \sum_i \gamma_i \xi_i,$$

where, $\gamma_i > 0$ and $\alpha_i > 0$ are Lagrange multipliers. $\mathcal{L}$ should be maximized with respect to γ_i and α_i. But, R must also be minimized with respect to $\mathbf{a}$ and ξ_i [40].

4. METHODOLOGY

This section introduces the proposed anomaly detection framework for a retail company. We first describe how to decide stores and products, and collect data from big data platform. Secondly, we introduce variables and explain transforming the missing values. Then, we present how to apply machine learning (ML) methods. Finally, we describe how to use the results of ML methods and how to evaluate the generation of alarms. The workflow of the methodology is shown Figure 4.1.

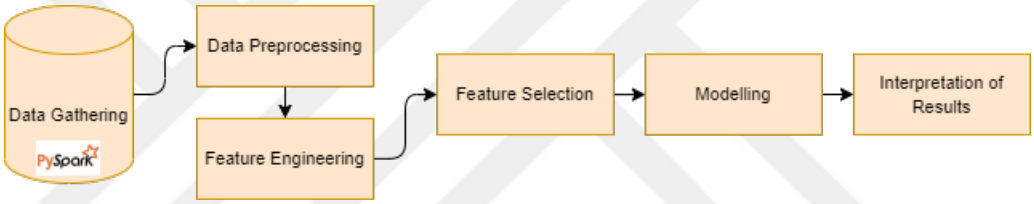

Figure 4.1 The overview of the methodology

4.1 Data Gathering

We collected data from the big data platform by using PySpark. PySpark has been released to support the collaboration of Apache Spark and Python, and it is a Python API for Spark.

In the meeting held with the business unit, it was discussed which stores and which products had most of the problems. Based on the needs, the data in the necessary tables were matched and combined.

The business unit suggested that we group them separately on store-SKU basis, as the quantity of stock varies for each store and each product. To give an example for product hierarchy, X brand chicken wing(201) and Y brand chicken wing(202) are in

the same Wing Subcategory; drumstick, wing and breast are in the Piece of Chicken Category; whole chicken, chicken offal and piece of chicken are in the Chicken Main Category. The product hierarchy is shown in Figure 4.2.

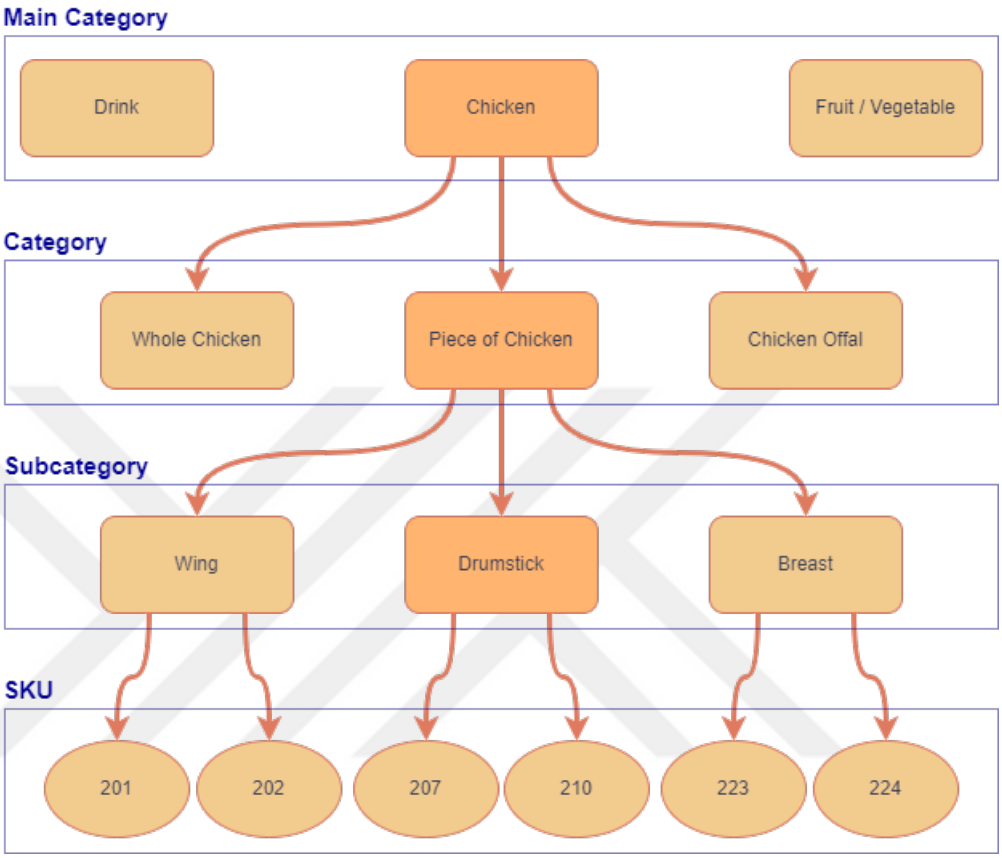

Figure 4.2 An example of product hierarchy

In this study, the products and stores with the most errors in the quantity of stock recommended by the business unit were examined. However, products in other stores and product categories can also be included in the model when they are examined. The masked version of the data used in the model is shown in Figure 4.3.

Store_Number	Store_Name	Product_Number	Product_Name	Stock	Transaction_Name	Transaction_Quantity
57	Store X	207	J Chicken Drumstick	0.496434	Warehouse to Store Upload	0.316543
50	Store Y	224	B Chicken Breast	0.629896	Warehouse to Store Upload	0.374138
62	Store Z	215	G Chicken Tenderloin	0.580467	Editing Stock Quantity	0.063137
52	Store Q	215	G Chicken Tenderloin	0.557360	Warehouse to Store Upload	0.157051
58	Store W	201	J Chicken Wing	0.825500	Warehouse to Store Upload	0.648524

Figure 4.3 Masked version of a certain part of the data

4.2 Data Preprocessing

Data preprocessing refers to the steps involved in transforming or encoding data to be easily interpreted by an algorithm [41]. Each data point may have unique properties, making it challenging to standardize the data. Furthermore, several issues may occur while gathering data. Data may be collected from several sources, there may be an issue with the system's flow, or the data going to the source may be incorrect.

For instance, suppose you collect data from numerous sources for students at school. Similarly, it is doubtful that all data with hundreds of records will be correct. For example, the age information or the student's surname may be missing in some records.

We preprocess the data to make it easier to understand and use. This procedure removes inconsistency or duplications in data that might impair a model's accuracy. Data preparation also guarantees that no values are wrong or missing due to human mistakes or defects. In short, applying data preparation techniques improves the completeness and exactness of the data.

4.2.1 Data imputation

The substitution of approximated values for missing or unreliable data elements is known as data imputation [42]. The replacement values are meant to provide a data record that is not prone to errors while editing. In our problem, some values of inventory stock can be missing because of system errors, or if there is no stock. We impute missing values with 0. Also, occasionally inventory stock is less than zero because, on some days, the number of products returned by customers may be higher than the product sold in subcategories; we replace negative stocks with 0.

4.3 Feature Engineering

Feature engineering increases the predictive power of machine learning algorithms and leads to better results by creating new features from raw data that helps streamline the machine learning process. We do feature engineering in two parts; Categorical Encoding and Data Normalization.

4.3.1 Categorical encoding

The most popular method is one-hot encoding (OHE), where each categorical value is transferred to a particular feature in the dataset containing binary 1 or 0. This operation requires mapping categorical values to integer values first. Then, each integer value is represented as a binary vector with all zero values except the integer index marked with 1.

OHE makes the representation of categorical data more impressive and easy. Many machine learning algorithms cannot work directly with categorical data. Therefore, categories must be converted to numbers. This operation is required for input and output variables that are categorical. The transformation of values using OHE is shown in Table 4.1.

Table 4.1 Transformation of transaction names using OHE

Original Value (Transaction Name)	**Transformed Value**
Warehouse to Store Upload	0
Editing Stock Quantity	1

4.3.2 Data normalization

The second step is data transformation, converting data from one format to another useful format. We use featurewise min-max normalization transformation because variables measured at different scales do not contribute equally to the model fitting

and learned function and may result in bias. We can formulate min-max transformation as:

$$x_{new} = \frac{x_i - \min(x)}{\max(x) - \min(x)} \tag{4.1}$$

When we look at Table 4.2, the original values are between 8 and 12. When we apply min-max normalization, the range becomes 0-1. The maximum value is 12, and it becomes 1 in the new representation. We apply this method to all variables in the dataset.

Table 4.2 An example of Min-Max Normalization

Original Value	**Calculation**	**Normalized Value**
10	$\frac{(10-8)}{(12-8)}$	0.5
8	$\frac{(8-8)}{(12-4)}$	0
12	$\frac{(12-8)}{(12-8)}$	1
9	$\frac{(9-8)}{(12-8)}$	0.25

4.4 Feature Selection

The main criterion for success is to find the right features and set up the model using it cleaned. Using too many variables can reduce the performance of the model. Feature Selection is also required to get a model that is easier to understand and works faster. Additionally, it reduces the risk of overfitting in the model.

It is more difficult to implement feature selection in unsupervised learning than in supervised learning. Label information is expensive to obtain which requires both time and efforts. Unsupervised methods seek alternative criteria to define feature relevance. In this study, we used more than one method while applying feature selection and we also benefited from our business knowledge.

We used the Variance Threshold, Mean Absolute Difference and Dispersion Ratio methods. The features with high correlation were determined with the Pearson

Correlation method. Then, we created a correlation matrix using Python's data visualization library Seaborn. The correlation matrix we created is shown in Figure 4.4.

The correlation matrix in Figure 4.4 shows the relationship between all the features and each other and values are positioned between -1 and 1. Values close to -1 indicate negative correlation and values close to 1 indicate positive correlation. The values of two positively correlated variables increase or decrease together. As the value of one of the two negatively correlated variables increases, the value of the other decreases. If the value is close to 0, it shows that there is no relationship between these two variables.

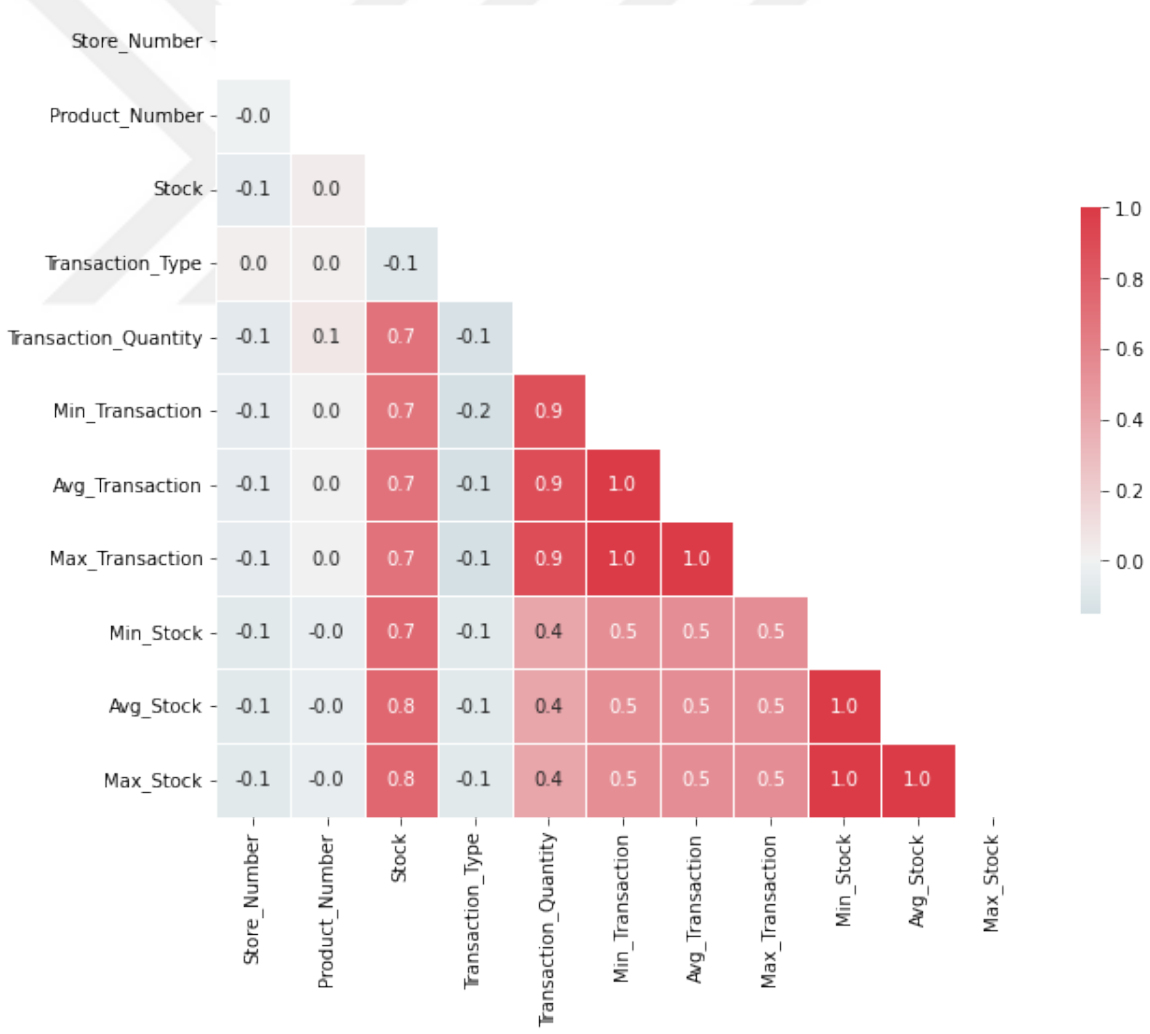

Figure 4.4 Correlation matrix

We detected the features with a correlation value close to 0 with the Stock feature via the correlation matrix. In order for our model to produce more accurate outputs, we removed Store_Number, Product_Number and Transaction_Type features from the data set and did not include them in the model.

4.5 Modelling

We used three machine learning algorithms within the scope of the research, namely, Isolation Forest, Local Outlier Factor and One-Class Support Vector Machine methods. The theoretical background of the ML methods used in modeling is mentioned in the Chapter 3. Since the splits of the decision tree are chosen at random, Isolation Forest is faster to train. In general, SVM are slow to train, especially with respect to the training set size. Since LOF works locally, it has the ability to catch the points missed by Isolation Forest and OCSVM. Since LOF works locally, it can catch different outliers than IF and OCSVM. Details on the use of the methods are mentioned in Chapter 5.

4.6 SHAP Analysis

SHAP(SHapley Additive ExPlanations) Analysis provides us results by calculating Shapley values from game theory, using the logic in Shapley game theory [43]. Shapley values are a method of showing the relative impact of each feature (or variable) we measure on the final output of the machine learning model, by comparing the relative impact of the inputs to the mean. Machine learning models are difficult to explain with traditional methods. SHAP is an alternative method that makes models more explainable. The SHAP method allows observing the effect of each feature on model success. It can offer the opportunity to create the same or higher model success with the most effective features detected [44].

The effect of the features that affect the prediction result while making a prediction is shown in Figure 4.5. In the SHAP Analysis example in Figure 4.5, the input

variables Age, Sex, BP, and BMI were used. The effects of these variables on the output of the model are shown as positive (red) and negative (blue) on the SHAP waterfall plot.

Figure 4.5 Example model estimation explainability with SHAP [45]

4.7 Interpretation of Results

When samples formerly identified by the business unit were tested in our model, all three algorithms detected anomalies correctly. Since there is no comparable target value in the current data used in our study, we worked in coordination with the business unit. One by one, e-mails were sent to the stores for the stores and products identified as outliers in the modeling results. The success performance of the algorithms has emerged with the answers from the store managers.

5. EXPERIMENTS

This thesis is aimed to find anomalies in the inventory stock by using real-life data. We define inventory records inaccuracy as an anomaly; when the proposed system detects the inventory errors, the inventory management will be more effective, and customer satisfaction will increase. On the other hand, generally, researchers are forced to use artificial data in their studies because of data privacy. Therefore, it is extremely important to use real-life data and demonstrate how to handle problems. However, another goal of this thesis is to provide a generic framework for unsupervised anomaly detection that can be used in all inventory stock data.

We collect the main dataset for a store type ve a product subcategory type by using PySpark from the big data platform. Our data includes Migros Ticaret A.Ş.'s the real inventory stock of 590 different products under the Piece of Chicken Category belonging to 331 different stores. Our sample size is 3344 with 14 columns. Our data set consists of the inventory stock quantities and the stock transactions affecting those quantities. In addition, the average of that week's 7-day inventory stock and the average of that week's 7-day inventory transactions are contained. Also, there are based on these averages, a lower limit and upper limit variable(column) for the inventory stock, and a lower limit and upper limit variable(column) for stock transactions in data. It is planned to run the model on a daily basis, to evaluate the results and correct inventory stock within that day. A snapshot of the dataset is shown in Figure 5.1.

Stock	Transaction_Type	Transaction_Amount	Min_Transaction	Avg_Transaction	Max_Transaction	Min_Stock	Avg_Stock	Max_Stock
1.000000	0.0	1.000000	0.868421	0.872417	0.872507	0.368421	0.375552	0.375629
0.980201	0.0	0.770784	1.000000	1.000000	1.000000	0.657895	0.671944	0.672048
0.922572	0.0	0.497521	0.552632	0.555010	0.555016	0.842105	0.853621	0.853647
0.903131	0.0	0.688960	0.500000	0.505424	0.505372	0.710526	0.720478	0.720581
0.888538	0.0	0.356183	0.526316	0.513946	0.513896	0.868421	0.856735	0.856865

Figure 5.1 A snapshot of the dataset

The business unit's request was that we review products with the most incorrect inventory stocks. According to business unit's recommendation, we analyzed the products affiliated with the Piece of Chicken Category. These inventory stock errors were found to occur almost daily. Errors are usually caused by human error. When the product arrives at the store, mistakes are made, such as the staff typing the weight value of the product incorrectly or the people in the warehouse sending more products than the store wants. When the store managers were contacted for the products detected as anomalies, we learned the reasons for the errors.

In the data transformation step, we apply min-max normalization to our data; so, all variables are between 0 and 1. The data consisting of 14 columns and 3344 rows was reduced to 11 columns after feature selection processes.

After making our data to be used in modeling, we used the Isolation Forest method adding the scikit-learn library in Python. While establishing the Isolation Forest model, we set up a structure with 100 trees. We determined the contamination value of the model as 0.005. Because when we increase this value, more outliers will appear. But it will be difficult to confirm all of them. We visualized the detected anomalies in the scatter plot.

We plotted the two of them, one after the other, to see the anomaly points clearer. Figure 5.2 shows the intersections of outliers in the Stock and Transaction Quantity features.

Then, we used the Local Outlier Factor method adding the scikit-learn library in Python. While establishing the LOF model, we set up a structure with 20 neighbors. We determined the contamination value of the model as 0.005. Because when we increase this value, more outliers will appear. But it will be difficult to confirm all of outliers. We visualized the detected outliers in Figure 5.3.

We plotted the two of them, one after the other, to see the anomaly points clearer. This chart shows the intersections of outliers in the Stock and Transaction Quantity

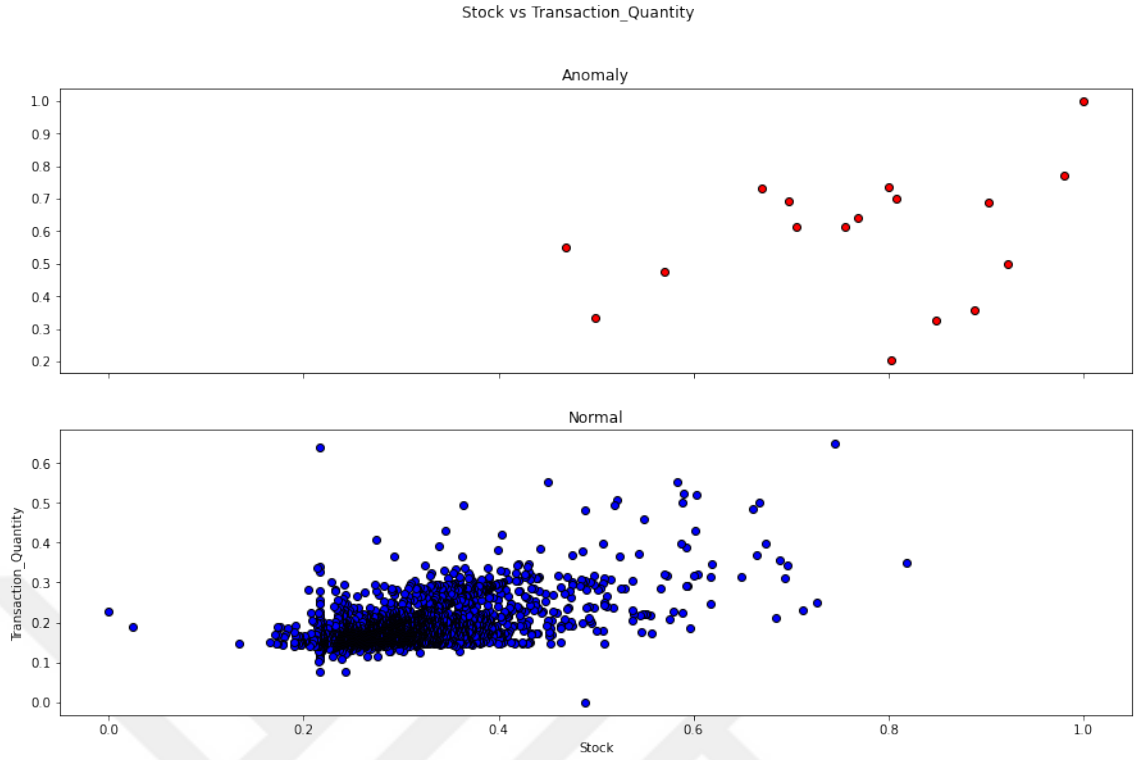

Figure 5.2 Stock vs. Transaction Quantity (Isolation Forest)

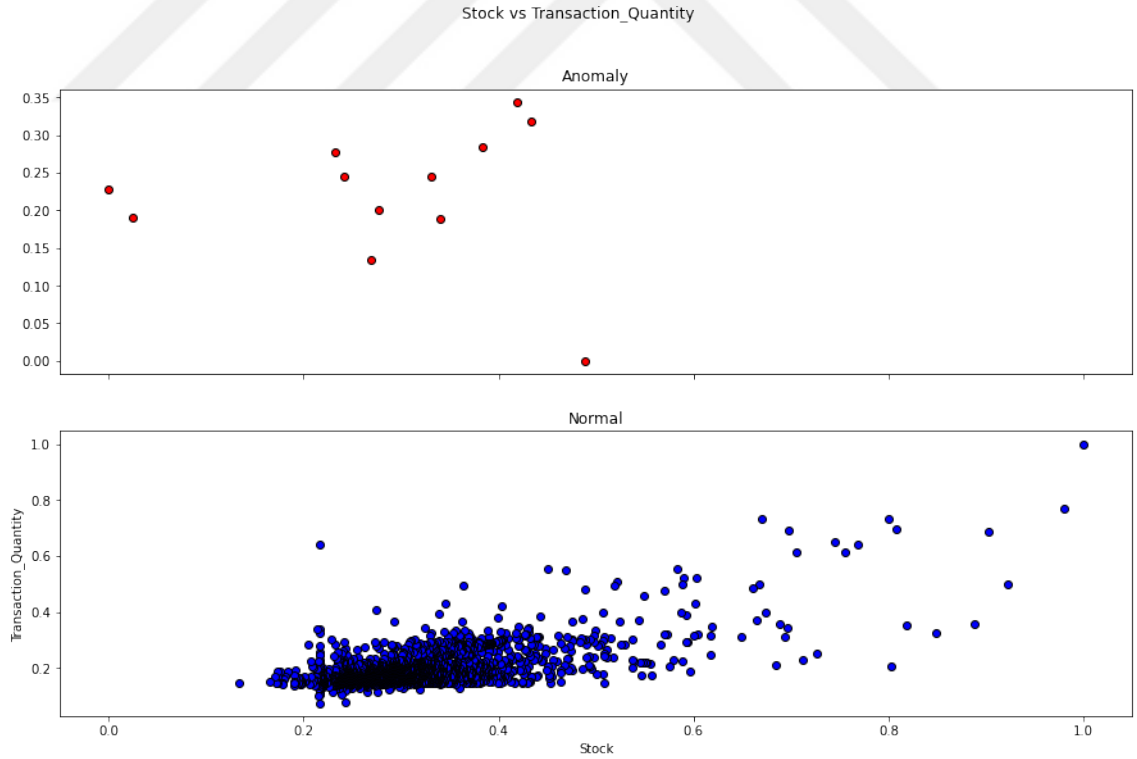

Figure 5.3 Stock vs. Transaction Quantity (LOF)

features. In this scatter plot, unlike the previous graph (the IF model's scatter plot), the intersection points with high Transaction Quantity values and low Stock values were determined as outliers.

Lastly, we used the One-Class Support Vector Machine method adding the scikit-learn library in Python. We determined the contamination value of the model as 0.005. Because when we increase this value, more outliers will appear. But it will be difficult to confirm all of anomalies. We visualized the detected anomalies in Figure 5.4.

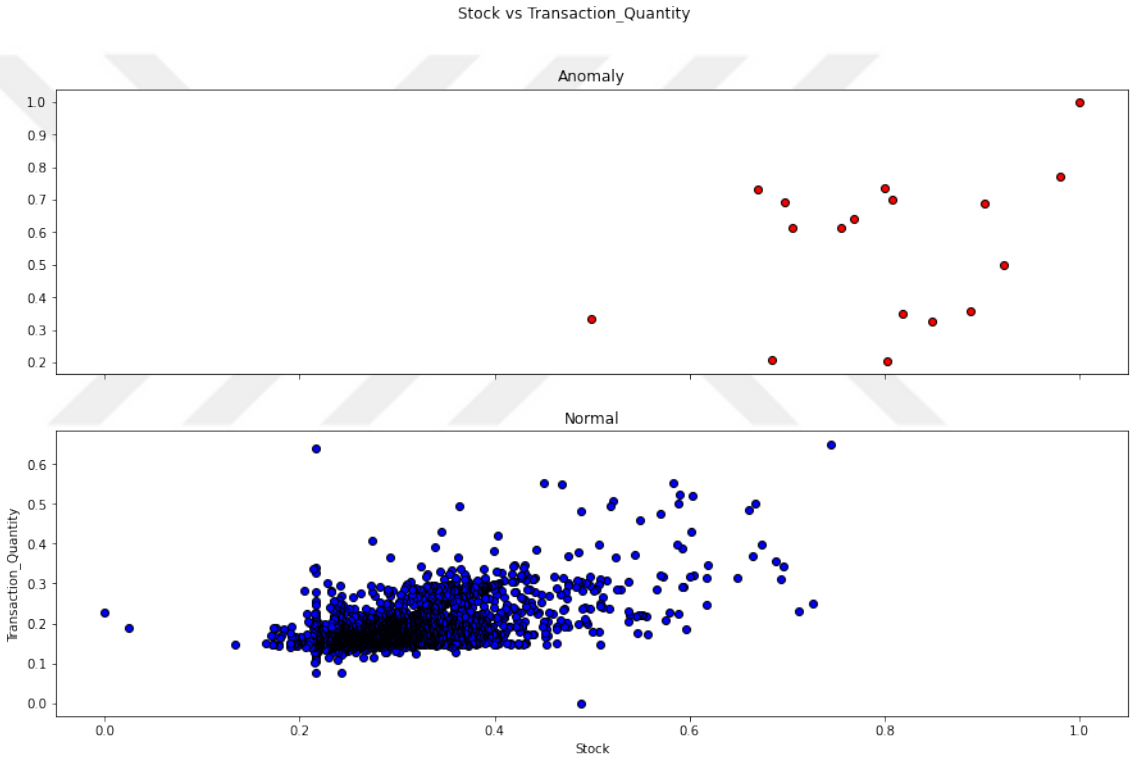

Figure 5.4 Stock vs. Transaction Quantity (OCSVM)

We plotted the two of them, one after the other, to see the anomaly points clearer. This chart shows the intersections of outliers in the Stock and Transaction Quantity features. The OCSVM's graph is different from the LOF's graph. But it is similar to the IF's graph.

After running the models and visualizing the anomalies found, we analyzed the features affecting the models with SHAP. We examined which features had a greater

effect on the models.

In Figure 5.5, the feature importance graph created by SHAP analysis is given. The high values (red parts) represent anomalies while the lower values (blue parts) are normal values. The order of importance of the features is listed from top to bottom in SHAP graph. The feature that affects the model the most was determined as Avg_Stock.

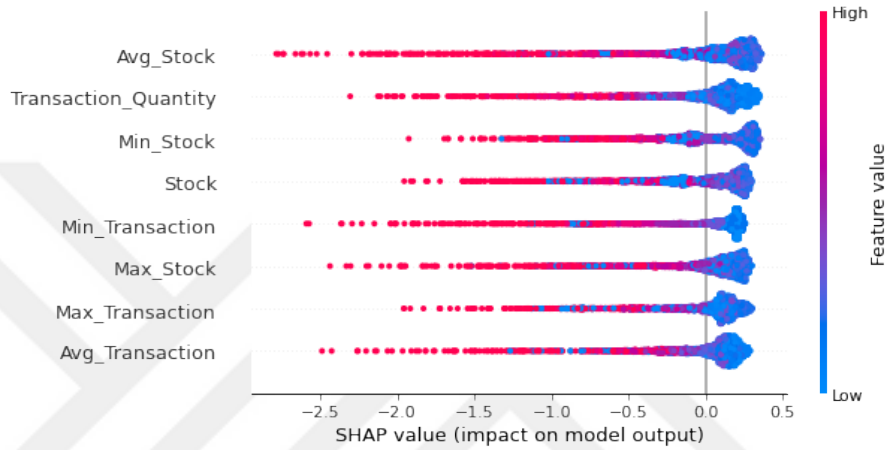

Figure 5.5 SHAP feature importance graph

We added the anomaly detection results of each algorithm to the dataframe as columns with the names of the algorithms. The results are shown in Figure 5.6 in descending order of the Stock value.

	Stock	Transaction_Quantity	isofr_anomaly	lof_anomaly	svm_anomaly
0	1.000000	1.000000	1	0	1
1	0.980201	0.770784	1	0	1
2	0.922572	0.497521	1	0	1
3	0.903131	0.688960	1	0	1
4	0.888538	0.356183	1	0	1
...	...	...	...	...	...
3339	0.170527	0.169273	0	0	0
3340	0.164782	0.150443	0	0	0
3341	0.133954	0.146487	0	0	0
3342	0.024749	0.189675	0	1	0
3343	0.000000	0.227286	0	1	0

Figure 5.6 Anomaly detection results by models

We counted the abnormal and normal values according to the model results and showed them in Table 5.1.

Table 5.1 Numbers of outliers and normal values in the results

Algorithms	**Outliers**	**Normal Values**
Isolation Forest	17	3,327
Local Outlier Factor	12	3,332
One-Class Support Vector Machine	15	3,329

Then, we moved on to the most important stage of our work. We confirmed the stock amounts by asking the store managers about the results determined by the algorithms in our model.

A comparison of model results with actual values is shown in Table 5.2. According to the final results, the most successful algorithm is LOF. In the sample data we mentioned in this study, LOF detected 12 outliers and 11 of them were indeed anomaly confirmed by the business unit. Although IF and OCSVM algorithms detected some real abnormal values, they did not provide as much benefit as LOF. In the sample data, IF detected 17 outliers. Only 6 of them are truly abnormal values. OCSVM's performance is also similar to IF. It detected 15 outliers in the sample data, 4 of which are true anomaly. According to the results of our study, the product with the highest number of errors in stock quantities is the 202 SKU product.

Table 5.2 Comparison of model results with real values

Month	SKU	Quantity	Algorithm	Output	The Real
Nov	226	64,614 kg	LOF	1	Anomaly
Sep	222	877 kg	LOF	1	Anomaly
Jun	211	625 kg	LOF	1	Anomaly
Oct	202	623 kg	LOF	1	Anomaly
Jun	236	560 kg	IF & OCSVM	1	Anomaly
Oct	235	130 kg	OCSVM	1	Normal
Jul	202	123 kg	IF & OCSVM	1	Anomaly
Oct	221	99 kg	OCSVM	1	Normal
Jun	202	90 kg	LOF	1	Anomaly
Oct	236	86 kg	IF	1	Normal
Jun	235	81 kg	LOF	1	Anomaly
Jul	209	58 kg	IF	1	Anomaly
Aug	235	55 kg	LOF	0	Normal
Jun	210	46 kg	IF	1	Anomaly
Nov	227	44 kg	LOF	1	Anomaly
Jul	202	41 kg	LOF	1	Anomaly
Jul	202	40 kg	IF	1	Anomaly
Jun	208	38 kg	IF & OCSVM	1	Anomaly
Jun	210	31 kg	OCSVM	1	Anomaly
Aug	203	22 kg	IF	0	Normal
Jun	216	15 kg	LOF	1	Anomaly
Jun	217	13 kg	LOF	1	Anomaly
Jun	203	10 kg	OCSVM	0	Normal
Nov	223	8 kg	IF & OCSVM	0	Normal
Jul	202	5 kg	LOF	0	Normal

6. CONCLUSION AND FUTURE WORKS

The discrepancy between the quantity recorded in a company's inventory management system and the number actually physically available is known as the inventory record inaccuracy (IRI). IRI can cause major problems in the retail industry, such as stockouts and revenue loss due to poor stock replenishment.

This study detects the errors in the inventory and defines these as an anomaly. It has a unique positioning as our dataset is taken from real-life while previous studies heavily relied on artificial datasets. Anomaly detection is a prevalent challenge for large industries such as retail industry. This research aims to contribute to the development of the unsupervised approach for anomaly identification. The key benefit of the method is its applicability to different products data. The lack of labels creates the biggest challenge for this study, as they introduce limitations for evaluating the previous cases. Indeed, in the absence of tags and ground truth, evaluating metrics and criteria for unsupervised anomaly detection algorithms remains a difficult practical challenge despite few recent studies on the subject.

Furthermore, in unsupervised anomaly detection, the idea of normality, which is typically intuitive for specialists, proves the presence of challenges to define anomaly in formal terms. However, from the point of anomaly detection, describing what is normal appears to be more rational than determining what is abnormal. In large-scale anomaly detection applications, the definition of what is abnormal is frequently conditioned by the company's ability to respond to these abnormalities. An AD algorithm under such constraints aims to select the most abnormal observations that are able to checked and confirmed by the company or service provider.

This thesis proposes a generic, unsupervised, and scalable framework for anomaly detection in Migros Ticaret A.Ş.'s data, which is retail inventory stock data in this

study. The suggested approach meets all of the objectives established in the early stages of this research and can detect aberrant behaviour in data from very various domains, contributing to several probable application areas of AD such as finance and health.

One of the primary contributions of this thesis is based on the approach's unsupervised character. While unsupervised learning is significantly more challenging than supervised learning, the proposed system has a promising capacity to learn meaningful data representations and subsequently detect anomalous occurrences.

On the other hand, applying machine learning algorithms to real-life data is not straightforward, especially in unsupervised learning. The study provides approaches on how to overcome the problems (e.g., noises, missing values) deriving from the structure of the data in real life. From a retail perspective, we can apply our method to different industry components such as product categories, stores, and warehouses. Implementation of our approach to the inventory management system could help industry players prevent errors before they occur. The business units could assess and validate the model results, which could further feedback and improve the unsupervised anomaly detection model. However, the business unit examined our anomaly points. For the most part, they confirmed many high-value points as anomalies but they give feedback that lower-valued ones needed further investigation.

We ran this model in the company on a daily basis and reported our findings monthly. Due to the redundancy of data and work/time constraints, it is not possible to confirm all outputs. The LOF approache has been shown to function well on the real dataset, allowing for a quick way to locate data that do not adhere to usual behaviour. We can list the algorithms as LOF $>$ IF $>$ OCSVM in terms of the benefits they provide to the company. The financial benefit provided to the company as a result of the 6 months running of this work is 2,492,129 TL.

Although our approach has shed some light on the subject with promising suggestions, further research could expand our knowledge about unsupervised anomaly

detection and overcome the study's shortcomings. Following recommendations could benefit the literature and provide opportunities for real-life applications:

- The scope of the current study is limited to one category, the replication of the study in different store type and products could provide additional insights.
- The attempts to improve the flexibility and applicability of the model may get us closer to the desired state for unsupervised anomaly detection.
- In order to automate the model, the operation of the model can be scheduled and the results can be sent to the stores automatically.
- A warning mechanism can be created for the screens used by the store personnel in order to eliminate the errors at the source.

BIBLIOGRAPHY

[1] N. Chehrazi, "Impacts of inventory record inaccuracy on retailers' internal operations," *SSRN Electronic Journal*, 2020.

[2] H. H.-C. Chuang, R. Oliva, and S. Liu, "On-shelf availability, retail performance, and external audits: A field experiment," *Production and Operations Management*, vol. 25, no. 5, pp. 935–951, 2015.

[3] A. Shabani, G. Maroti, S. de Leeuw, and W. Dullaert, "Inventory record inaccuracy and store-level performance," *International Journal of Production Economics*, vol. 235, p. 108111, 2021.

[4] E. Natsvlishvili, E. Lomtadze, and N. Khatiashvili, "ERP system implementation challenges in Georgian medium-sized enterprises, retail sector," *EPRINTS*, 01-Jan-1970. [Online]. Available: http://eprints.iliauni.edu.ge/9843/. [Accessed: 14-Feb-2023].

[5] W. J. Stevenson, *Operations management*, 14th ed. McGraw-Hill, 2021.

[6] D. Epstein, "Various types of inventory management systems for your business," *Financesonline.com*, 29-Dec-2022. [Online]. Available: https://financesonline.com/various-types-of-inventory-management-systems-for-your-business/. [Accessed: 14-Feb-2023].

[7] I. H. Sarker, "Machine learning: Algorithms, real-world applications and Research Directions," *SN Computer Science*, vol. 2, no. 3, 2021.

[8] C. C. Aggarwal, *Outlier analysis*. New York, NY: Springer New York, 2013.

[9] R. Chalapathy and S. Chawla, "Deep Learning for Anomaly Detection: A Survey," 2019.

[10] Y. LeCun, "Obstacle to Progress in Deep Learning & AI," *NYU Tandon School of Engineering - Polytechnic Institute*, 2018. [Online]. Available: https://engineering.nyu.edu/news/revolution-will-not-be-supervised-promises-facebooks-yann-lecun-kickoff-ai-seminar. [Accessed: 14-Feb-2023].

[11] Y. Kang and S. B. Gershwin, "Information inaccuracy in inventory systems: Stock loss and Stockout," *IIE Transactions*, vol. 37, no. 9, pp. 843–859, 2005.

[12] N. DeHoratius and A. Raman, "Inventory record inaccuracy: An empirical analysis," *Management Science*, vol. 54, no. 4, pp. 627–641, 2008.

[13] S. Cannella, J. M. Framinan, M. Bruccoleri, A. P. Barbosa-Póvoa, and S. Relvas, "The effect of inventory record inaccuracy in Information Exchange Supply Chains," *European Journal of Operational Research*, vol. 243, no. 1, pp. 120–129, 2015.

[14] T. J. Kull, M. Barratt, A. C. Sodero, and E. Rabinovich, "Investigating the effects of daily inventory record inaccuracy in Multichannel Retailing," *Journal of Business Logistics*, vol. 34, no. 3, pp. 189–208, 2013.

[15] M. Barratt, T. J. Kull, and A. C. Sodero, "Inventory record inaccuracy dynamics and the role of employees within multi-channel Distribution Center Inventory Systems," *Journal of Operations Management*, vol. 63, no. 1, pp. 6–24, 2018.

[16] H. H.-C. Chuang and R. Oliva, "Inventory record inaccuracy: Causes and labor effects," *Journal of Operations Management*, vol. 39-40, no. 1, pp. 63–78, 2015.

[17] Z. Ton and A. Raman, "The effect of product variety and inventory levels on retail store sales: A longitudinal study," *Production and Operations Management*, vol. 19, no. 5, pp. 546–560, 2010.

[18] A. G. Kök and K. H. Shang, "Inspection and replenishment policies for systems with inventory record inaccuracy," *Manufacturing & Service Operations Management*, vol. 9, no. 2, pp. 185–205, 2007.

[19] A. G. Kök and K. H. Shang, "Evaluation of cycle-count policies for supply chains with inventory inaccuracy and implications on RFID Investments," *European Journal of Operational Research*, vol. 237, no. 1, pp. 91–105, 2014.

[20] Y. Zhang, Y. Chen, J. Wang, and Z. Pan, "Unsupervised deep anomaly detection for multi-sensor time-series signals," *IEEE Transactions on Knowledge and Data Engineering*, pp. 1–1, 2021.

[21] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "LOF," *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, 2000.

[22] Chire, "Local outlier factor," *Wikipedia*, 19-Nov-2022. [Online]. Available: https://en.wikipedia.org/wiki/Local_outlier_factor#/media/File:LOF.svg. [Accessed: 14-Feb-2023].

[23] C. C. Aggarwal and C. K. Reddy, *Data clustering: Algorithms and applications*. Boca Raton, FL: Chapman and Hall/CRC, 2014.

[24] M. Jianliang, S. Haikun, and B. Ling, "The application on intrusion detection

based on K-means cluster algorithm," *2009 International Forum on Information Technology and Applications*, 2009.

[25] X. Tao, Y. Peng, F. Zhao, P. Zhao, and Y. Wang, "A parallel algorithm for network traffic anomaly detection based on Isolation Forest," *International Journal of Distributed Sensor Networks*, vol. 14, no. 11, p. 155014771881447, 2018.

[26] S. Ounacer, H. Ait El Bour, Y. Oubrahim, M. Y. Ghoumari, and M. Azzouazi, "Using isolation forest in anomaly detection: The case of credit card transactions," *Periodicals of Engineering and Natural Sciences (PEN)*, vol. 6, no. 2, p. 394, 2018.

[27] C. Li, L. Guo, H. Gao, and Y. Li, "Similarity-measured isolation forest: Anomaly detection method for machine monitoring data," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–12, 2021.

[28] R. Zhang, S. Zhang, S. Muthuraman, and J. Jiang, "One class support vector machine for anomaly detection in the communication network performance data," *Proceedings of the 5th conference on Applied electromagnetics, wireless and optical communications*, pp. 31-37, 2007

[29] E. H. Pena, M. V. de Assis, and M. L. Proenca, "Anomaly detection using forecasting methods Arima and HWDS," *2013 32nd International Conference of the Chilean Computer Science Society (SCCC)*, 2013.

[30] P. Filonov, F. Kitashov, and A. Lavrentyev, "RNN-based early cyber-attack detection for the Tennessee Eastman Process," *arXiv.org*, 07-Sep-2017. [Online]. Available: https://arxiv.org/abs/1709.02232v1. [Accessed: 14-Feb-2023].

[31] B. Lindemann, N. Jazdi, and M. Weyrich, "Anomaly detection and prediction in discrete manufacturing based on cooperative LSTM Networks," *2020 IEEE 16th International Conference on Automation Science and Engineering (CASE)*, 2020.

[32] T. Ergen and S. S. Kozat, "Unsupervised anomaly detection with LSTM neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 8, pp. 3127–3141, 2020.

[33] P. Malhotra, A. Ramakrishnan, G. Anand, L. Vig, P. Agarwal, and G. Shroff, "LSTM-based encoder-decoder for multi-sensor anomaly detection," *arXiv.org*, 2016

[34] J. An, and S. Cho, "Variational autoencoder based anomaly detection using reconstruction probability," *Special lecture on IE*, 2(1), pp. 1-18, 2015

[35] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation Forest," *2008 Eighth IEEE International Conference on Data Mining*, 2008.

[36] K. G. Mehrotra, C. K. Mohan, and H. M. Huang, "Anomaly detection principles and algorithms," *Terrorism, Security, and Computation*, 2017.

[37] S. Omar, A. Ngadi, and H. H. Jebur, "Machine learning techniques for anomaly detection: An overview," *International Journal of Computer Applications*, vol. 79, no. 2, pp. 33–41, 2013.

[38] H. Dogan, D. Forte, and M. M. Tehranipoor, "Aging analysis for recycled FPGA detection," *2014 IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT)*, 2014.

[39] R. Chalapathy, A. K. Menon, and S. Chawla, "Anomaly detection using one-class neural networks," *arXiv.org*, 2018

[40] Baeldung, "What is one class SVM and how does it work?," *Baeldung on Computer Science*, 04-Nov-2022. [Online]. Available: https://www.baeldung.com/cs/one-class-svm. [Accessed: 14-Feb-2023].

[41] S. García, J. Luengo, and F. Herrera, "Data preprocessing in Data Mining," *Intelligent Systems Reference Library*, 2015.

[42] B. Efron, "Missing data, imputation, and the bootstrap," *Journal of the American Statistical Association*, vol. 89, no. 426, pp. 463-475, 1994.

[43] S. M. Lundberg, and S. I. Lee, "A unified approach to interpreting model predictions," *Advances in neural information processing systems*, 30, 2017.

[44] G. A. Tataroglu, G. Ozbulak, and K. K. Eren, "Determination of the genetic variant reliability using Shap Approach," *2020 28th Signal Processing and Communications Applications Conference (SIU)*, 2020.

[45] "SHAP," *Welcome to the SHAP documentation - SHAP latest documentation.* [Online]. Available: https://shap.readthedocs.io/en/latest/index.html. [Accessed: 14-Feb-2023].

CURRICULUM VITAE

Personal Information

Name Surname : Görkem Erdem

Education

Undergraduate Education : Beykent University (2017)
Graduate Education : Kadir Has University (2023)
Foreign Language Skills : English

Work Experience

Name of Employer and Dates of Employment: Migros Ticaret A.Ş. 2020 May – Present
Migros Ticaret A.Ş. via Innotech Teknoloji Dan. 2019 February – 2020 April
Crede Danışmanlık 2018 June – 2019 January
Vodafone (Internship) 2016 July – 2016 September
SOFT İş Çözümleri A.Ş. (Internship) 2012 August – 2012 August
Turkcell Superonline (Internship) 2010 September – 2011 June