Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2023) 31: 163 – 179
© TÜBİTAK
doi:10.55730/1300-0632.3976

Variational autoencoder-based anomaly detection in time series data for inventory record inaccuracy

Halil ARĞUN, S. Emre ALPTEKİN*
Industrial Engineering, Galatasaray University, Turkey

Received: 12.06.2022 • **Accepted/Published Online:** 29.11.2022 • **Final Version:** 19.01.2023

Abstract: Retail companies monitor inventory stock levels regularly and manage them based on forecasted sales to sustain their market position. Inventory accuracy, defined as the difference between the warehouse stock records and the actual inventory, is critical for preventing stockouts and shortages. The root causes of inventory inaccuracy are the employee or customer theft, product damage or spoilage, and wrong shipments. In this paper, we aim at detecting inaccurate stocks of one of Turkey's largest supermarket chain using the variational autoencoder (VAE), which is an unsupervised learning method. Based on the findings, we showed that VAE is able to model the underlying probability distribution of data, regenerate the pattern from time series data, and detect anomalies. Hence, it reduces time and effort to manually label the inaccuracy in data. Since the distribution of inventory data depends on selected product/product categories, we had to use a parametric approach to handle potential differences. For individual products, we built univariate time series, whereas for product categories we built multivariate time series. The experimental results show that the proposed approaches can detect anomalies both in the low and high inventory quantities.

Key words: Inventory record inaccuracy, variational autoencoder, anomaly detection, time series data

1. Introduction

Inventory management is an essential aspect of operations management [1]. Inventory control issues can result in both understocking and overstocking of products. Late deliveries, lost sales, customer complaints, and production bottlenecks arise from understocking, whereas overstocking wastes space and money that could be better spent elsewhere. The overall purpose of inventory management is to deliver exceptional customer service, while keeping inventory costs within reasonable bounds [2]. In retail industry, correct operation of automated inventory management plays a vital role since they operate with low-single-digit net profit margins. Many retail companies implement enterprise resource planning systems (ERP) to accurately track inventory levels and reduce human-related errors [3]. In general, the company information system's inventory data are based on daily sales and shipment calculations, but mistakes are made in accepting goods or sending goods from one store to other stores and ERPs cannot always handle these errors. Inventory record inaccuracy (IRI), the name given to discrepancies in inventory levels in the literature, causes 1% sales and 3% gross profit loss in the retail industry [4]. These discrepancies is usually referred to as an anomaly. Anomaly detection (AD) aims to distinguish a significant dissociation of a point itself from its past value or predicted future value [5]. In the literature, there are various algorithms on different types of anomaly detection, from credit card fraud to medical diagnosis [6].

*Correspondence: ealptekin@gsu.edu.tr

Machine learning is one of the methods that can detect anomalies, hence the IRI. The output of AD algorithms can be of two types: anomaly score and binary label. The anomaly score provides the level of anomaly, where binary label indicates whether a point is an outlier. Although several algorithms directly return binary labels, outlier scores can also be transformed to binary labels. Anomaly score has more advantages compared to binary label, because we can rank scores and define a threshold to convert from anomaly score to binary label. Ordering is critical for the business units to analyze most suspicious events. AD methods can be classified as supervised and unsupervised depending on whether labels are used in the training phase. The supervised technique is not well-suited to real world applications because labeling data usually necessitates domain knowledge and a significant amount of time. Moreover, anomaly data are frequently unbalanced with varied types, and the properties of anomalous data may be unknown [7]. Many researchers have looked at standard unsupervised AD methods assuming that anomalies constitute a distinct minority class via the usage of distance-based, density-based, and angle-based methods [8–10]. However, as the number of features and the size of the data increase, calculating corresponding similarity for each point in the data becomes challenging. If dimensional reduction techniques are applied, the details in the data may be lost [11]. To this end, many deep learning algorithms have been developed [7]. In order to achieve good performance and state-of-the-art outcomes, large labeled data sets are required for training these algorithms. However, as previously said, large-scale labeled data sets are difficult to access by, and the annotation process necessitates domain expertise. Hence, researchers are working to develop unsupervised learning models for problems and tasks that have previously been disregarded. On that front, unsupervised learning in the works like [12] demonstrated remarkable results using techniques such as autoencoders (AE). Nevertheless, unsupervised learning is still a complex topic to tackle as it consistently underperforms supervised learning in various tasks.

In this work, methods based on deep learning are used to detect the IRI. Since the errors in the inventory levels are to some content hidden in multidimensional data, Variational Autoencoder (VAE), one of the deep algorithms, is applied to catch these errors. VAE is suitable for dealing with this task, as it proposes a way of describing the probability of observation in latent space. VAE first encodes the high dimensional data to latent variables and then decodes data to the original dimension. Instead of creating a single value encoder or static encoder to describe each latent attribute, VAE is designed to define a probability distribution for each latent attribute [13]. At the end, the original data and algorithms' output have differences that are known reconstruction errors, and anomalies in this context are defined as data points with higher reconstruction probabilities [14]. In this framework, the numerical inventory stocks are examined as daily time series. A time series is a series of data points collected in equal time intervals. A time series collects observations or data for a particular variable collected over a defined and ordered set of periods. This paper aims to detect the anomalies with the VAE method before they are reflected on the store by using the real inventory stock data of one of Turkey's largest supermarket chains. Inventory record inaccuracy is a big problem in the retail sector [15], and detecting the anomaly before it occurs and generating an alarm in case of a mistake can prevent this error before it happens. Thus, the contributions of this work can be summarized as follows: • Detecting anomalies in data whose characteristics change over time; people's consumption habits can change or the company's strategy can change. • Detecting anomalies in univariate time series; the company may want to observe just one product. • Detecting anomalies in multivariate time series; the company may want to analyze more than one product at the same time. • Reducing the number of false alarms in the anomaly detection system; companies have limited sources to investigate the alarm. • Establishing a dynamic anomaly detection structure; • Proposing a method that ease of adding a new feature, extracting the current features and dynamically defining the thresholds.

2. Related work

The relevant literature is reviewed in two parts: 1. The studies using inventory management to deal with IRI, and 2. The studies using unsupervised time series anomaly detection methods.

2.1. Inventory record inaccuracy

The information gathered from automated inventory management information systems is crucial for a successful business. The presence of IRI in companies is identified via experimental proofs in several papers, which define the difference between QR (recorded inventory quantity) and QP (psychical stock quantity). IRI research can be divided into five areas: the presence of IRI, the effects of IRI, the causes of IRI, the reduction of IRI, and the method of measuring IRI [4]. To decrease IRI, many organizations can use advanced information technologies like enterprise resource planning and radio frequency identification [16]. Several researchers suggest several managerial actions. Kök and Shang provide a simple recursion to estimate the system's cost and propose a heuristic approach to obtain efficient baselines [17].

Similar to our paper, there are studies in literature that consider inventory stock errors as anomalies in time series. Anomaly detection is assessed as supervised, semisupervised, or unsupervised [18]. We concentrate on unsupervised learning algorithms which are mainly categorized into classical anomaly detection and deep anomaly detection methods [18]. Classical methods use linear calculation, and some of them are distance-based, density-based, angle-based methods [8–10]. These methods are based on statistical and similarity calculation, because of calculations when data is increasing the training phase takes too much time. Another classical method is isolation forest [19], and other tree-based approaches are particularly appealing 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. Another popular classic method is one-class support vector machine (OC-SVM) that differentiates one class of observations from another by using hyperplanes in multidimensional space [20].

2.2. Deep anomaly detection methods

We can divide DL anomaly detection models into two groups: forecasting and using the reconstruction error of input values [18]. 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 [21], the authors use an RNN-based model to detect anomalies in multivariate time series data to prevent cyberattacks by minimizing mean squared error between actual and predictions. In [22], they use LSTM to detect anomalies in manufacturing processes by using advantages of long term effects of processes. AEs are frequently used for AD by learning to rebuild a given input. The model is only trained on normal data. When it cannot rebuild the input with the same quality as regular data reconstruction, the input sequence is classified as anomalous data [18]. LSTM autoencoder models can reveal anomalies by using long term effects [23]. VAEs are a special form of AE that models the relationship between two random variables, latent variable z and visible variable x. Typically, a prior for z is a multivariate unit Gaussian $\mathcal{N}(0; \mathrm{I})$. VAE learns the distribution of variables, and this feature provides a dynamic structure for different variables. The details of VAE is given in Section 3.4. In [14], authors define the reconstruction probability for anomaly detection as the average probability of the original data generated from the distribution. Data points with a high chance of reconstruction are classed as anomalies, and vice versa.

3. Methodology

This section summarizes the anomaly detection framework proposed for a Turkish retail company (Figure 1). We first describe how to decide product categories and collect data from the big data platform. Next, we introduce variables and explain how to create time-related features. After explaining how to overcome the missing and extreme values, we introduce the theoretical background of VAE. Finally, we explain how to use the result of VAE and how to evaluate the generation of alarms.

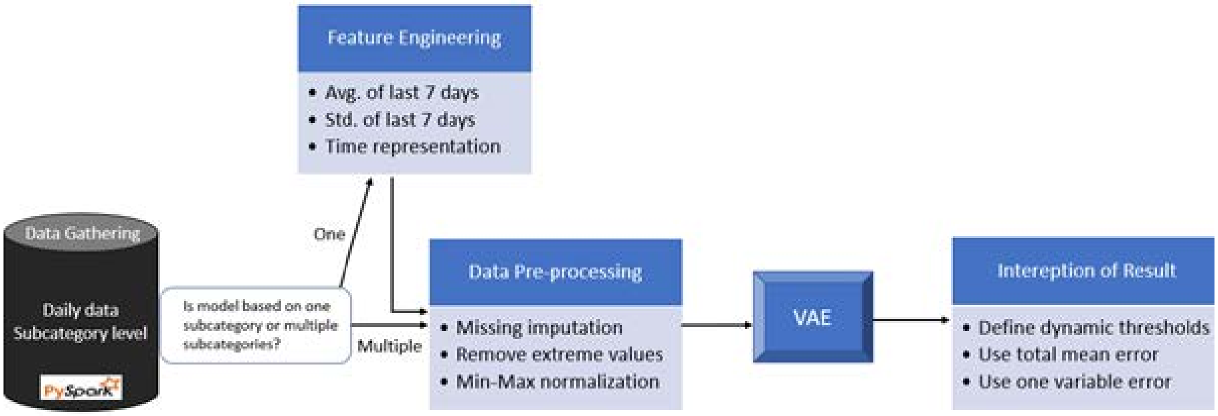


Figure 1. The overview of methodology.

3.1. 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. To understand business requirements, we had a meeting with the data quality department before starting to analyze data. In the meeting, as the items have changed over the years, they suggested to make subcategories that includes different brands of the same product. We aggregate inventory stock information based on subcategories daily, so our problem becomes to detect anomalies in time series data. To give an example for product hierarchy, 'brand A' and 'brand B' pasteurized milk are in the same pasteurized milk subcategory; organic milk, pasteurized milk, yogurt, and ayran are in the Milk & Yogurt category, while ice-cream, cheese and milk & yogurt are in the dairy main category. We pick a store and sum up the number of products under the same subcategory daily, so we have a data set including day and inventory stock quantity. The product hierarchy is shown in Figure 2.

If the firm wants to examine only one subcategory, it takes only that subcategory from the data and include it in the model. If the firm wants to consider more than one subcategory at the same time and analyze the interaction between them, they should integrate other categories in the model. The first case is handled as a univariate time series, while the other is treated as a multivariate time series. Examples for univariate and multivariate time series are shown in Figure 3.

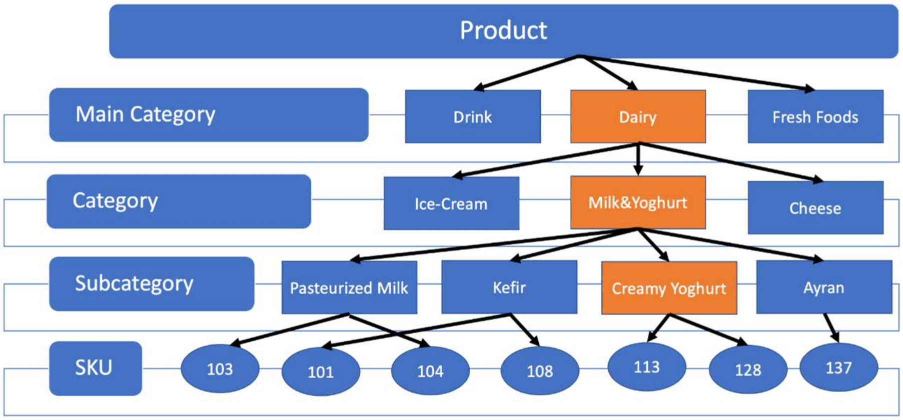
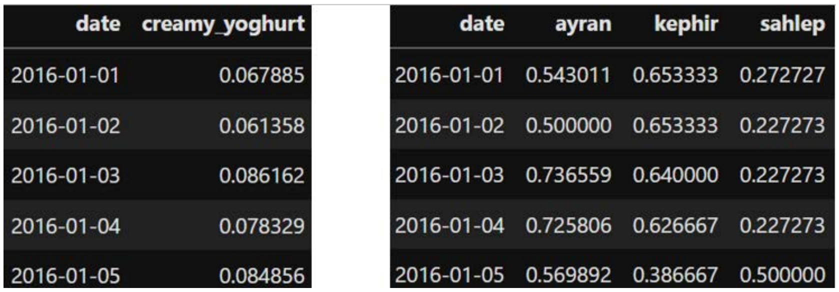


Figure 2. An example of product hierarchy.



date	creamy_yoghurt
2016-01-01	0.067885
2016-01-02	0.061358
2016-01-03	0.086162
2016-01-04	0.078329
2016-01-05	0.084856

date	ayran	kephir	sahlep
2016-01-01	0.543011	0.653333	0.272727
2016-01-02	0.500000	0.653333	0.227273
2016-01-03	0.736559	0.640000	0.227273
2016-01-04	0.725806	0.626667	0.227273
2016-01-05	0.569892	0.386667	0.500000

Figure 3. Univariate and multivariate time series.

3.2. Data explanations and feature engineering

After collecting the data, we need to decide which type of anomalies we want to detect. The first step is to create features based on current (shown as x_t) and past quantities. These features give information about history of data and changes in data. The list of variables with their calculation is presented in Table 1.

Table 1. Calculation of variables.

Variable	Explanation	Calculation
x_t	Current day stock quantity	No need to calculate
$x_{pctchange}$	Percent change from previous days	$\frac{(x_t - x_{t-1})}{x_{t-1}}$
$x_{avglast7}$	Average of last 7 days	$\frac{1}{7}\sum_{i=0}^{6} x_{t-i}$
$x_{stdlast7}$	Standard deviation of last 7 days	$\sqrt{\frac{1}{7-1}\sum_{i=0}^{6}(x_{t-i} - \overline{x})^2}$
$x_{CVlast7}$	Coefficient of variation	$\frac{x_{stdlast7}}{x_{avglast7}}$

The second step is to represent time that is critical to catch seasonality and pattern depending on the date. One of the most popular method is one-hot encoding, where each categorical value is transferred to a particular feature in the data set containing binary values, 1 or 0. A disadvantage of this method is that it increases the size of the data. For example, if we use one-hot encoding for the day of year variable, we have to add 365 columns to our data. In order to cope with it, based on [24], we defined time as (x,y) coordinates on a circle to represent cyclical features (Table 2) with only two variables instead of 365, resulting that the lowest value of every feature appears right next to the largest value. For example, the first day of the year (1st of January) should be close to the last year's last day (31st of December). At this point, the coordinates should be able to handle the closeness of time-related values with each other. However, if we stick to the ordinal values of a day in the year, we cannot achieve the desired result. Hence, we use the sine and cosine components of a day in the year as represented in Figure 4. The upper part of the graph presents sine of the year's day, whereas the lower part presents its cosine. When we look to the sine of day representation, we can notice that the values of the first day and the last day are close as desired. However, if only a single component is used to represent time (either sine or cosine), we observe that the sine value of 1st of June (midyear) is equal to the sine value of the 1st of January. To overcome this symmetry issue in the model, the cosine component for the cyclical feature should be used as well, as suggested in [24]. Using the sine and cosines features together, the proximity is represented correctly using locations on the circle (Figure 5), where each value for each variable is represented with a unique point.

Table 2. Representation of time as a sin-cos function.

Variable	Explanation	Calculation	Variable	Explanation	Calculation
daysin	Sine of the year's day	$\sin(\frac{2*\pi*currentday}{365})$	*weekcos*	Cosine of the week of the year	$\cos(\frac{2*\pi*currentweek}{52})$
daycos	Cosine of the year's day	$\cos(\frac{2*\pi*currentday}{365})$	*monthsin*	Sine of the month of the year	$\sin(\frac{2*\pi*currentmonth}{12})$
weeksin	Sine of the week of the year	$\sin(\frac{2*\pi*currentweek}{52})$	*monthcos*	Cosine of the month of the year	$\cos(\frac{2*\pi*currentmonth}{12})$

On the other hand, we do not create base features, such as average and time representation features, for multivariate time series, because the company wants to catch the anomaly that caused interaction between the subcategories. For example, let us consider the scenario of what happens if a store may order traditional milk instead of pasteurized milk. The traditional milk subcategory inventory stock will increase, and new products do not replace the pasteurized milk subcategory stock. The store has too many stocks in the traditional milk subcategory, which causes overstocking, but the store will stock out because of less pasteurized milk. This mistake directly affects the cost and customer satisfaction.

3.3. Data preprocessing

As each data point has unique properties, it is a challenge to standardize data. Besides, data may be collected from several sources, which causes an issue with the system's flow. The prepossessing removes inconsistency or duplications in data that might impair a model's accuracy. It also guarantees that no values are wrong or missing due to human mistakes or defects. Hence, this step involves transforming or encoding data for easily being interpreted by an algorithm [25]. We preprocess data in two steps: The first step is data cleaning

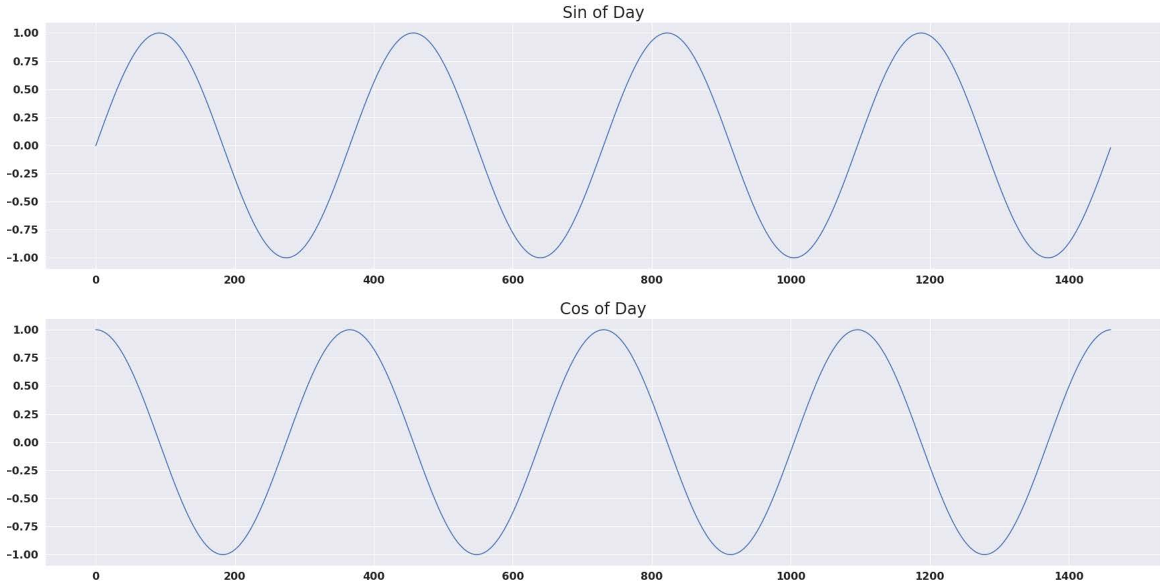


Figure 4. Representation of time as a function of sin and cos.

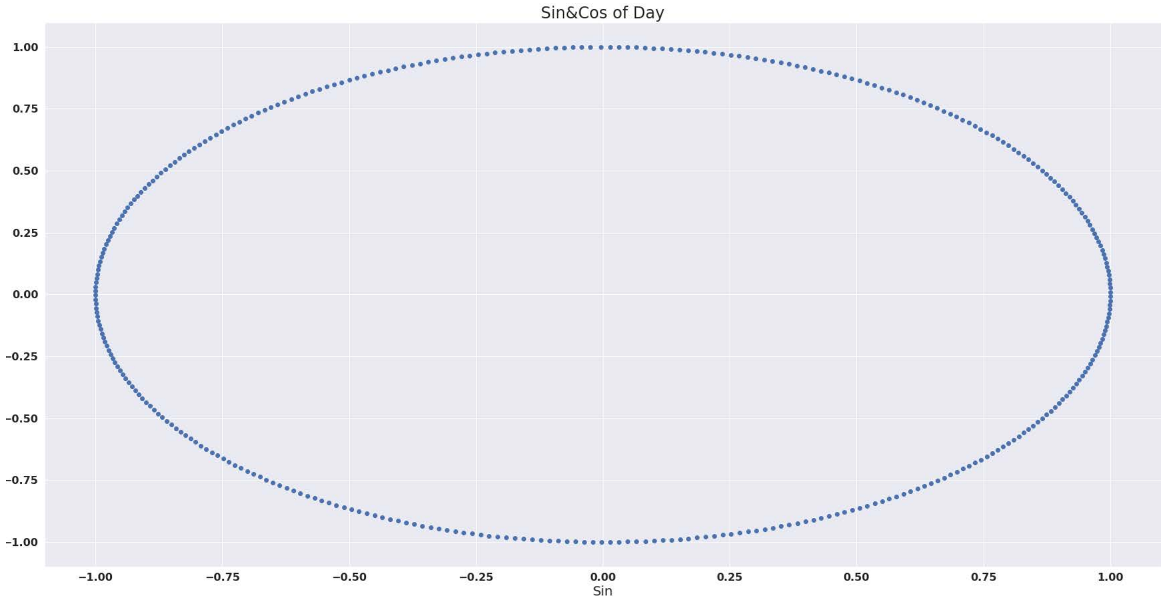


Figure 5. The day of year as a function of sin and cos.

that refers to handling missing or extreme values in the dataset. The missing values due to system errors are replaced with 0. Moreover, occasionally inventory stock is observed to be less than zero because, on some days, the number of products returned by customers may be higher than the product sold in subcategories. We also replace the negative stocks values with 0. On the other hand, a dataset may contain extreme values that are outside the acceptable range. Identifying and even deleting them enhances ML modeling. For this purpose, we use the interquartile range (IQR) method. IQR is defined as the difference between the 75^{th} and 25^{th} percentiles, $(Q_3 - Q_1)$ of data [28]. Generally, extreme values are defined as $1.5 * IQR$ below Q_1 or $1.5 * IQR$

above Q_3 (Tukey's method). Based on our discussion with the business unit, they indicated that they want to avoid losing potential anomalies in data; therefore, we used $3 * IQR$ instead of $1.5 * IQR$ to highlight extreme values. The business unit analyzed these extreme values separately and confirmed the appropriateness of the range. Accordingly, we replace the extreme values with null values, and then the null values are filled with imputation with linear interpolation [29]. The second step is data transformation, that is converting data from one format to another useful one. We use feature-wise min-max normalization transformation (Equation 1) because variables measured at different scales do not contribute equally to the model fitting and learned function and they may result in bias.

$$x_{scaled} = \frac{x - min(x)}{max(x) - min(x)} \tag{1}$$

3.4. Variational autoencoders (VAE)

Compared to AE, VAE provides more options to adjust, allowing us to have more flexibility in modeling our latent distribution. VAE learns the data distribution, which is crucial when the data changes over time. Besides, VAE can be combined with other NN architectures, such as LSTM [26] and CNNs [27]. In our context, the customer behaviors change over time, and accordingly the companies have to change their strategies. Companies need to manage a huge amount of data about their customers and their habits to increase satisfaction and decrease their costs. This is the main reason of choosing VAE, which is a dynamic method.

VAE, a popular generative model, uses a network frame similar to AE. It has two parts: an encoder and a decoder [30]. The encoder's goal is to build a compressed feature set, named a latent space representation based on the input, whereas the decoder's goal is to reproduce the input data depending on the latent variables [31]. AE's training goal is to get the reconstructed term $\tilde{x}$ as close as possible to the original x, requiring AE to learn the latent attributes of the original data. The latent variable z is required in VAE to be distributed according to a prior distribution $p_\theta(\mathbf{z})$ in Equation 2, which is often multivariate unit Gaussian $\mathcal{N}(0; \mathrm{I})$. When mapping from input data x to latent variable z, $p_\theta(\mathbf{z} \mid \mathbf{x})$ is frequently intractable since $p_\theta(\mathbf{x})$ is also intractable.

$$p_\theta(\mathbf{z} \mid \mathbf{x}) = \frac{p_\theta(\mathbf{x} \mid \mathbf{z})}{p_\theta(\mathbf{x})} \tag{2}$$

As a result, by obtaining an approximation posterior $q_\phi(\mathbf{z} \mid \mathbf{x})$ in Equation 3, variational inference techniques are applied to achieve the tractability.

$$\mathbf{z} \sim q_\phi(\mathbf{z} \mid \mathbf{x}) = \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{z}}, \boldsymbol{\sigma}_{\boldsymbol{z}}{}^2 I) \tag{3}$$

The encoder calculates the mean $\boldsymbol{\mu}_{\boldsymbol{z}}$ and standard deviation $\boldsymbol{\sigma}_{\boldsymbol{z}}$ of the approximation posterior $q_\phi(\mathbf{z} \mid \mathbf{x})$. The evidence lower bound (ELBO) can be calculated using an inference model $q_\phi(\mathbf{z} \mid \mathbf{x})$ as follows (Equation 4):

$$\begin{aligned}\log p_\theta(\mathbf{x}) &= \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log p_\theta(\mathbf{x})\right] = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log \frac{p_\theta(\mathbf{x} \mid \mathbf{z}) p_\theta(\mathbf{z})}{p_\theta(\mathbf{z} \mid \mathbf{x})}\right] = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log \frac{p_\theta(\mathbf{x} \mid \mathbf{z}) p_\theta(\mathbf{z})}{p_\theta(\mathbf{z} \mid \mathbf{x})} \frac{q_\phi(\mathbf{z} \mid \mathbf{x})}{q_\phi(\mathbf{z} \mid \mathbf{x})}\right] \\ &= \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log p_\theta(\mathbf{x} \mid \mathbf{z}) + \log p_\theta(\mathbf{z}) - \log q_\phi(\mathbf{z} \mid \mathbf{x})\right] + D_{KL}\left(q_\phi(\mathbf{z} \mid \mathbf{x}) \| p_\theta(\mathbf{z} \mid \mathbf{x})\right)\end{aligned} \tag{4}$$

ELBO is the first term in Equation 4, and the Kullback-Leibler (KL) divergence of the estimated $q_\phi(\mathbf{z} \mid \mathbf{x})$ from the exact posterior $p_\theta(\mathbf{z} \mid \mathbf{x})$ is the second term. The KL divergence between them must be decreased to guarantee that $q_\phi(\mathbf{z} \mid \mathbf{x})$ approaches $p_\theta(\mathbf{z} \mid \mathbf{x})$. The objective of decreasing KL divergence can be achieved by increasing ELBO; hence, the VAE loss function can be written as Equation 5 [30]:

$$\mathcal{L}_{VAE}(\theta, \emptyset; \mathbf{x}) = -\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log p_\theta(\mathbf{x} \mid \mathbf{z}) + \log p_\theta(\mathbf{z}) - \log q_\phi(\mathbf{z} \mid \mathbf{x})\right] \tag{5}$$

4. Numerical results

The numerical application of our proposed framework is executed on real life data, taken from the biggest Turkish retail company includes real inventory stock of 18 different subcategories under the milk & yogurt category. It consists of the data between January 2016 and December 2019, with 2020 and 2021 omitted due to the pandemic. The company first wanted to see the result for 'normal' years. Our sample size is 1461 with 18 columns, and the raw data includes daily inventory stock information, hence 1461 different days. The inventory records inaccuracy is defined as an anomaly, and in this research we aim to present a generic framework for unsupervised anomaly detection that can be used in all-time series data. Firstly, we propose univariate time series anomaly detection for the creamy yogurt subcategory. Next, we develop a model including 18 different subcategories.

4.1. Univariate time series anomaly detection

This type of AD allows the company to focus on only one subcategory, especially, products with short expiration dates are very critical to catch. Therefore, we selected the creamy yogurt subcategory including 1461 samples and one column (inventory stock quantity). We created related features, such as percent change from the previous day, average and standard deviation of the last 7 days, coefficient of variation, time representation with sine & cosine for the day of the year, week of year, and month of the year. At the end of this feature engineering step, we obtained a total of 11 variables. The missing and negative values are replaced by 0, and daily inventory stock data is shown in Figure 6.

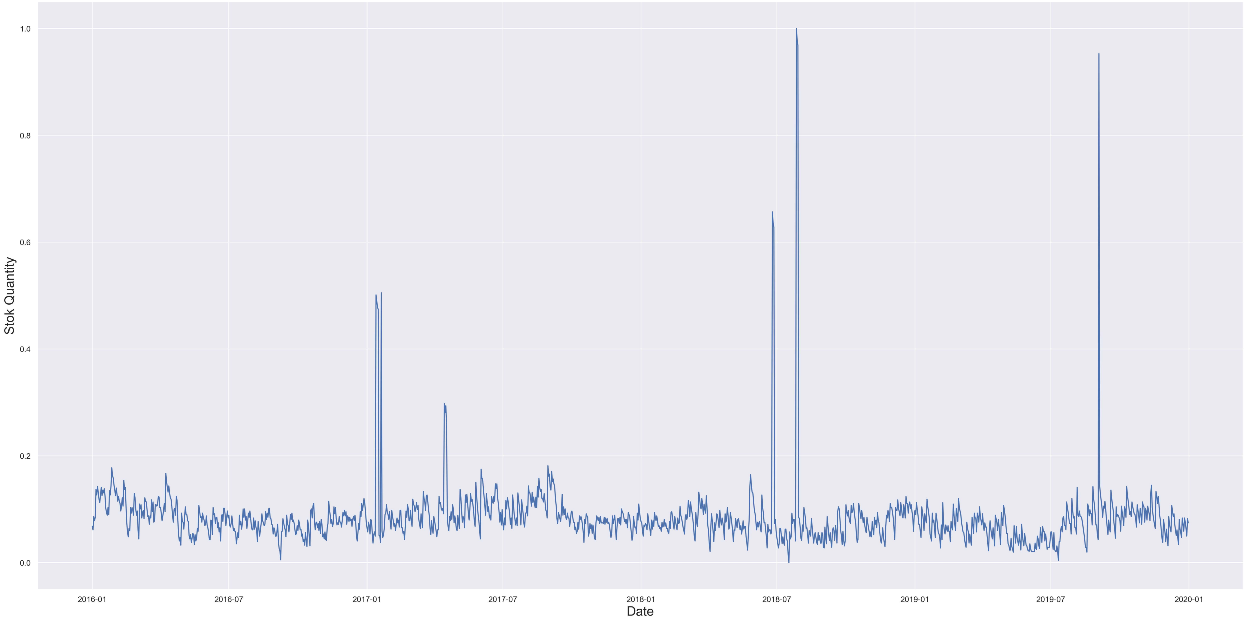


Figure 6. Daily stock of creamy yogurt.

From Figure 6, we can easily see that some values are considerably high compared to others, so we applied the IQR method as explained in Section 3.3. The processed graph of actual inventory stock is shown in Figure 7.

As the data transformation, we applied min-max normalization, so all variables had values between 0 and 1. Next, we created a VAE model using the Keras library. The architecture of the model has three separate layers for encoder and decoder, each with 11, 20, and 3 neurons (Figure 8). Eleven neurons represent the variable number in the input layer. The hidden layer includes 20 neurons, and the latent variable consists of three neurons, where each z represents different latent variable and each of them have a different average μ and standard deviation σ. The decoder is symmetric to the encoder (3-20-11). We use the batch size of 32, and the epoch number of 250.

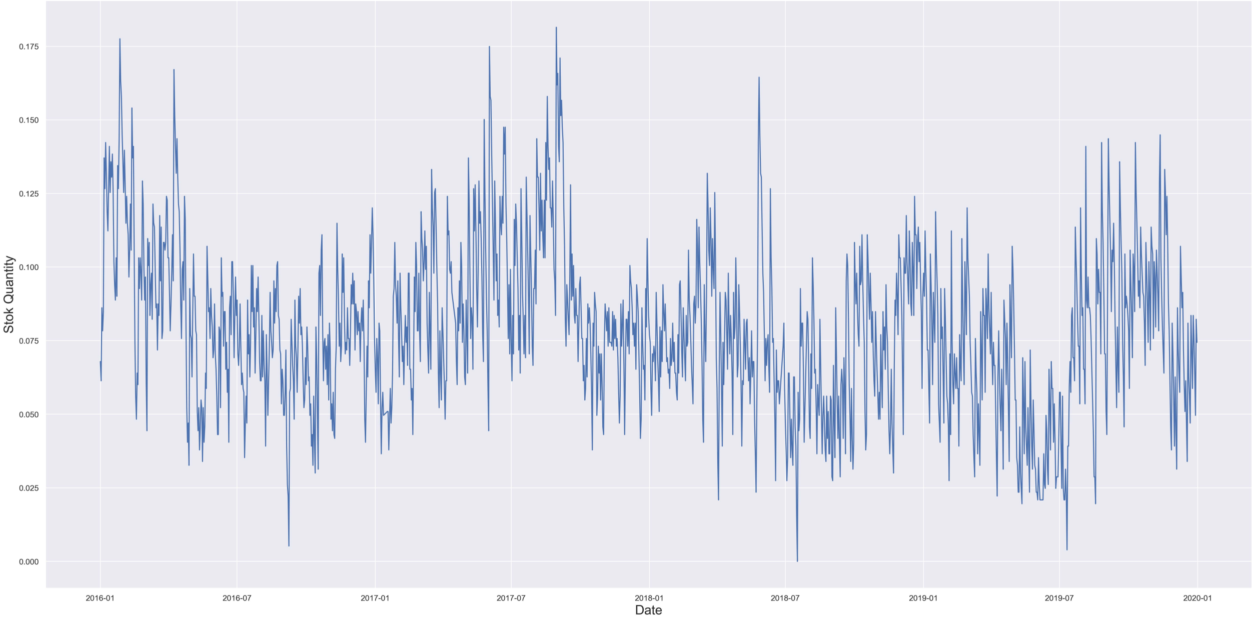


Figure 7. Daily stock of creamy yogurt after removing extreme values.

We used mean squared error as the loss function. After 250 epochs, the final loss is 0.3113. If we increase the epoch number or create a more complex structure, there is a risk that the model overfits the data. The absolute error for each feature and the total anomaly score are calculated. We define a day as an anomaly if the total anomaly score is greater than Q_3 + 1.5*IQR or the error of actual inventory stock is higher than Q_3 + 1.5*IQR. Hence, we identified 27 anomalies, which corresponds only to 3.2% of all records. As few alarms are produced, the business unit can easily review the anomalies. Besides, the model is able to detect high values as well as low values as anomalies. Moreover, the proposed univariate model can be modified to flag data as anomaly by defining additional features, such as percent change from the previous day. Using different features, the business unit potentially analyzes and makes inferences about customer behaviors. Various anomaly flags are presented in Table 3. The thresholds are different for each variable and each one of them may be analyzed separately.

In Figure 9, anomaly points in creamy yogurt that is based on the total anomaly score and actual stock are shown. Generally, high-level inventory stocks are marked, but we can also see several low ones. Another critical variable is the percent change of inventory stock. Figure 10 shows most of the big jumps of the model.

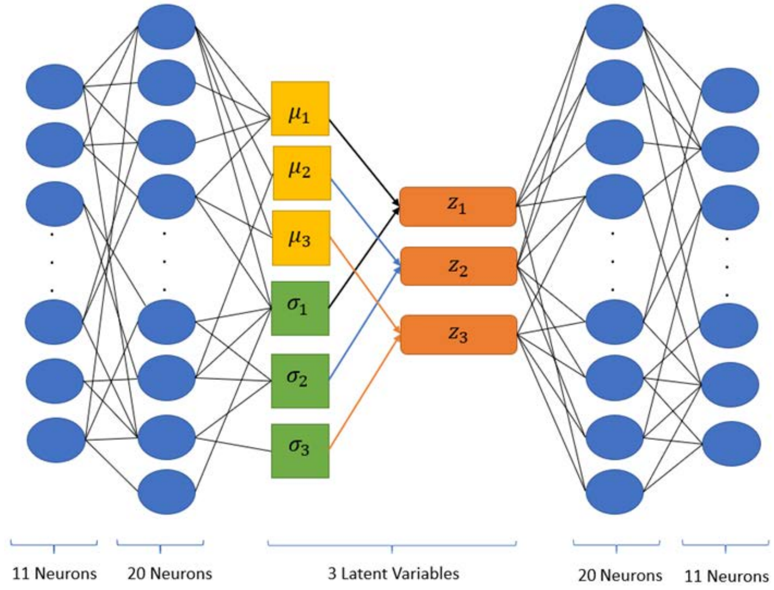


Figure 8. The architecture of VAE.

Table 3. Thresholds and number of anomalies.

Variable	Flag threshold	Number of anomalies	Variable	Flag threshold	Number of anomalies
x_t	0.396	27	$x_{stdlast7}$	0.259	47
$x_{pctchange}$	0.034	59	$x_{CVlast7}$	0.303	53
$x_{avglast7}$	0.366	42	$x_{total_anomaly_score}$	0.181	28

For comparison purposes, we tested the same data with isolation forest [19], which is a popular tree-based AD method that detects anomalies using isolation rather than modeling the normal points. The method with the standard parameters defined by sklearn library gave 146 anomaly points (Figure 11). When examining the anomaly points, we noticed that the model marked almost all high stock values as anomalies, which was not useful for the business unit. Furthermore, isolation forest calculates a single anomaly score for all the features, whereas VAE enables to analyze each feature separately. Hence, we believe that the model helps to pinpoint more effective features to identify anomalies and could be adapted to different stores of the retailer with different demand and supply behavior.

4.2. Multivariate time series anomaly detection

This type of AD allows the company to focus on more than one time series. The business unit can create a single model to detect anomalies in different subcategories. In this model, we used real inventory stock of 18 different subcategories under the milk & yogurt category, instead of focusing only creamy yogurt category. Hence, we have 1461 samples and 18 columns representing various subcategories' inventory levels. We did not include additional features as we did in univariate model as that would increase the total number of features to 60 (4 features for a variable plus time representation). Similar to the previous univariate model, we performed imputation of negative, missing, and extreme values and filled the missing values using IRQ method and subsequently

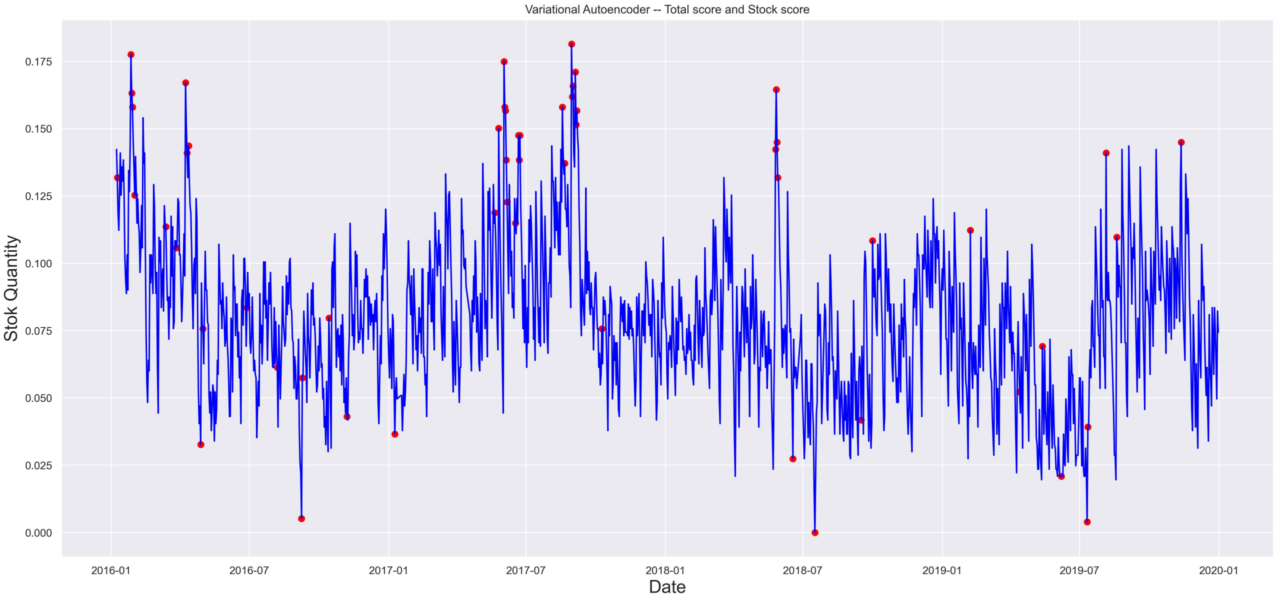


Figure 9. Anomaly points in creamy yogurt.

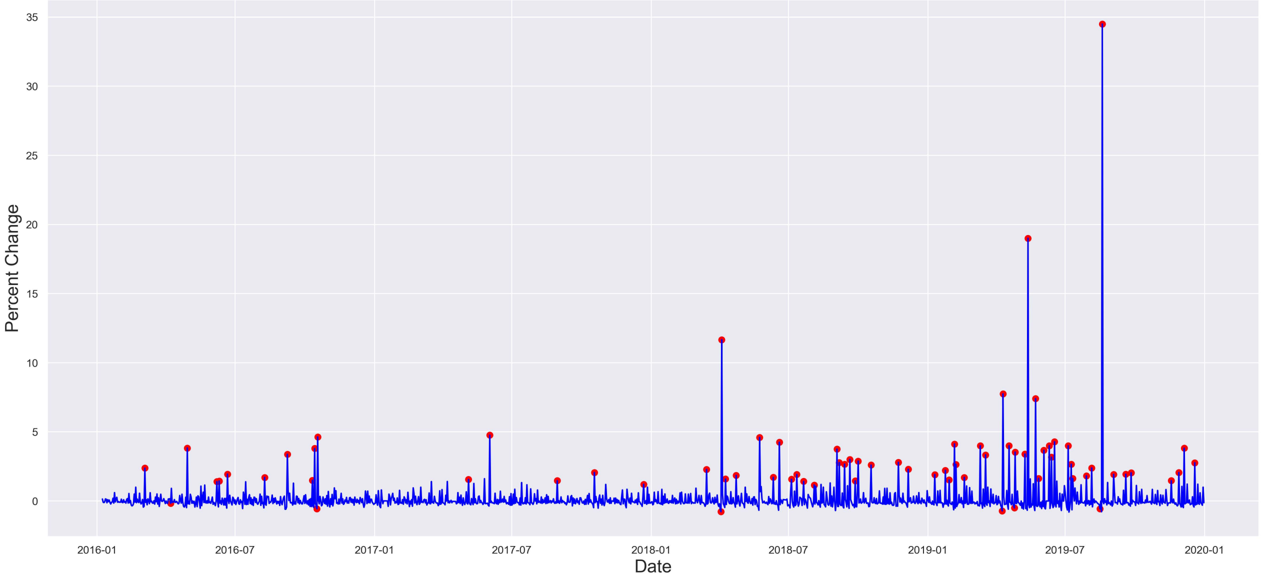


Figure 10. Anomaly points based on percent change variable.

normalized data. The representation of two of the 18 subcategories daily inventory stock data is shown in Figure 12.

Similar to the univariate model, we created a VAE model using the Keras library. The architecture of the model has three separate layers for encoder and decoder, each with 18, 25, and 5 neurons (Figure 13). Twenty neurons represent the variable number in the input layer. The hidden layer includes 25 neurons, and the latent variable consists of 5 neurons. The decoder is symmetric to the encoder (5-25-18). We used the batch size

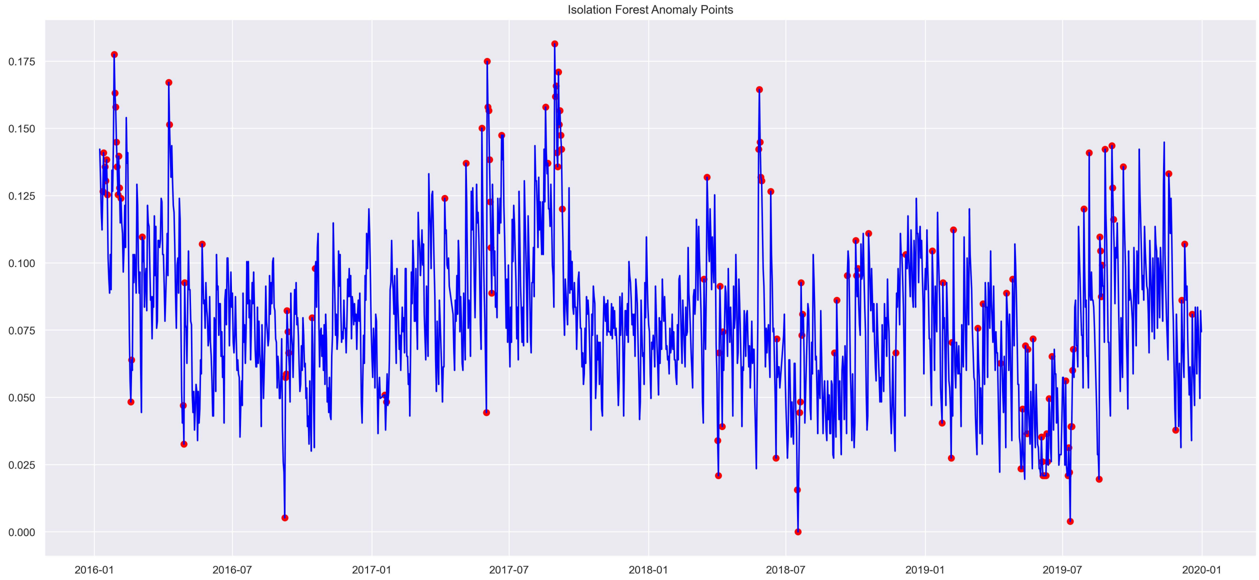


Figure 11. Anomaly points based on the isolation forest method.

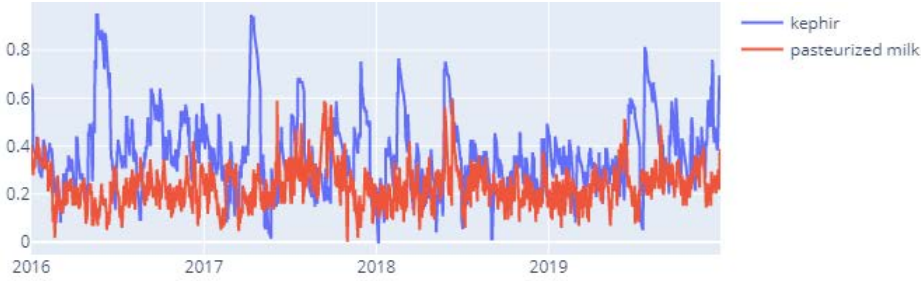


Figure 12. Daily stocks after removing extreme values.

of 32, and the epoch number of 500. Mean squared error is chosen as the loss function and after 500 epochs, the final loss is found as 0.3325. We increased the epoch number and create a more complex structure than the previous model in order to deal with higher number of features. We did not add extra layers nor increase training time to prevent overfitting. After fitting the model, we calculated the absolute error for each feature and the total anomaly score. The loss range of the multivariate model is found lower than the univariate model, and it enables us to analyze loss for each subcategory separately like Figure 14. We define a day as an anomaly, if the subcategory reconstruction error is greater than Q_3 + 1.5*IQR. Accordingly, we had different anomaly numbers for each subcategory. In this model type, the business unit can focus on one subcategory or more than one category at the same time. The number of anomaly flags is given in Table 4. The number of anomalies of the creamy yogurt subcategory is 33, which is greater than the one in the first model (it was 27). It means that the different subcategories interact with each other. For comparison purposes, we did not apply the isolation forest method to multivariate time series for each univariate series because this method gives only one total anomaly score.

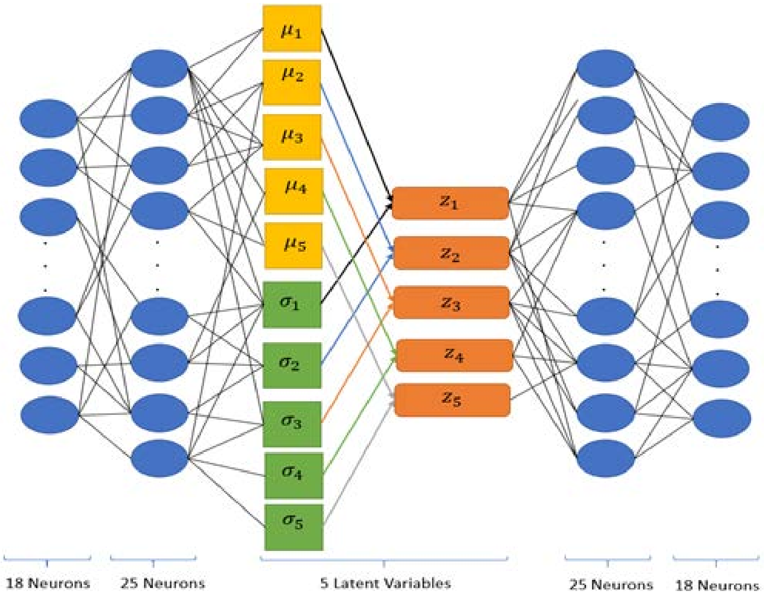


Figure 13. The architecture of VAE for multivariate time series.

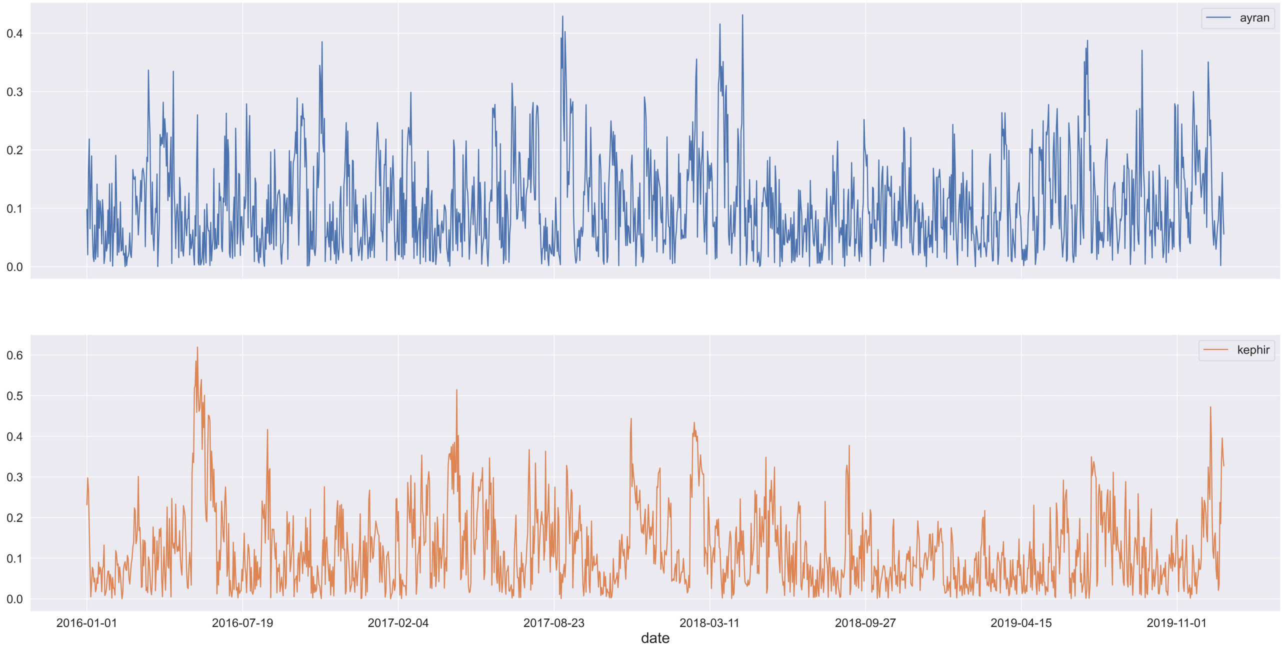


Figure 14. Loss for different subcategories.

5. Discussion and conclusion

The inventory record inaccuracy can cause major problems in the retail industry, such as stockouts and revenue loss due to poor stock replenishment. In this research, where we were able to work on real-life data, we detect the errors in the inventory and define them as anomalies. Doing so, we developed an unsupervised approach for anomaly identification. The key benefit of the proposed model is that it is applicable to different types

Table 4. The thresholds and number of anomalies for subcategories.

Subcategory	Flag threshold	Number of anomalies	Subcategory	Flag threshold	Number of anomalies
A	0.311	29	K	0.304	40
B	0.299	49	L	0.311	28
C	0.293	23	M	0.288	23
D	0.283	24	N	0.208	82
E	0.251	33	Creamy yogurt	0.289	33
F	0.296	40	O	0.296	24
G	0.225	59	P	0.292	30
H	0.287	38	R	0.295	33
J	0.284	36	S	0.323	24

of time series data. The lack of labels are the biggest challenge for all supervised models, as they introduce limitations for evaluating previous cases. Furthermore, in unsupervised AD, defining anomaly in formal terms composes another big challenge. In large-scale AD 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 can be checked and confirmed by the company or service provider.

This work proposes a generic, unsupervised, and scalable framework for AD in two types of time series data (univariate and multivariate). When making the right choice, business people need to know all the advantages and disadvantages of both models. Both frameworks are created as generic, so that it is applicable to different companies, with different subcategories. Univariate time series AD aims to find anomaly points in only one subcategory. All separately created features are related to that specific subcategory. Hence, business side has to build individual models for each subcategory. Multivariate time series AD could be beneficial, if all subcategories want to be covered in a single model. The main disadvantage of the second model is that, as all subcategories are examined in the same model, the interactions among them could create some anomalies. While unsupervised learning is significantly more challenging than supervised learning, we showed on the real data that the proposed system has a promising capacity to learn meaningful data representations and subsequently detect anomalous occurrences. Business unit people examined the obtained anomaly points. Generally, they confirmed many high-value points as anomalies, but they gave feedback that lower-valued ones needed further investigation.

Applying machine learning algorithms to real life data is not straightforward, especially in unsupervised learning. The paper 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 it 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 AD model. Another contribution is dynamic threshold definition, as it is not easy to analyze all features in the data set.

Although our approach has shed some light on the subject with promising suggestions, further research could expand our knowledge about unsupervised AD and overcome the study's shortcomings. Following recommendations could be beneficial and provide opportunities for real-life applications: • In the multivariate approach, clustering the subcategories may improve the overall model accuracy as clusters would have similar characteristics. • The scope of the current work is limited to one store of a retail company. The replication of the study in different geographical regions could provide additional insights. • The model's different variants (e.g., models with fewer layers, latent variables, batch size) can be generated at varying complexity levels. • Stock keeping unit-based prediction models could be designed, and the differences between prediction and real values can be examined to generate insights. • The development of a comprehensive model to cover all different products and their abnormalities is improbable. However, the attempts to improve the flexibility and applicability of the model may get us closer to the desired state for unsupervised anomaly detection.

References

[1] Stevenson WJ. Operations Management 14th Edition. McGraw-Hill, 2021.

[2] Chuang HHC, Oliva R, Liu S. On-shelf availability, retail performance, and external audits: a field experiment. Production and Operations Management 2016; 25: 935–951. https://doi.org/10.1111/poms.12519

[3] Natsvlishvili E, Lomtadze E, Khatiashvili N. ERP system implementation challenges in Georgian Medium-sized enterprises, retail sector. PhD, Ilia State University, USA, 2020.

[4] Shabani A, Maroti G, de Leeuw S, Dullaert W. Inventory record inaccuracy and store-level performance. International Journal of Production Economics 2021; 235: 1-16. https://doi.org/10.1016/j.ijpe.2021.108111

[5] Sarker IH. Machine learning: Algorithms, Real-World Applications and Research Directions. SN Computer Science 2021; 2 (160): 1–21. https://doi.org/10.1007/s42979-021-00592-x

[6] Aggarwal CC. An Introduction to Outlier Analysis. In: Outlier Analysis. New, York, NY: Springer New York, pp. 1-40, 2013. doi:10.1007/978-1-4614-6396-2

[7] Pang G, Shen C, Cao L, Hengel AVD. 2021. Deep Learning for Anomaly Detection: A Review. ACM Computing Surveys 2022; 54 (2): 1-38. https://doi.org/10.1145/3439950

[8] Tayeh T, Aburakhia S, Myers R, Shami A. Distance-Based Anomaly Detection for Industrial Surfaces Using Triplet Networks. In: 11th IEEE Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), Vancouver, VC, Canada; 2020, pp. 0372-0377. doi: 10.1109/IEMCON51383.2020.9284921

[9] Lou C, Zhao H. Local Density-Based Anomaly Detection in Hyperspectral Image. Journal of Applied Remote Sensing. 2015; 9 (1): 095070. https://doi.org/10.1117/1.JRS.9.095070

[10] Kriegel HP, Schubert M, Zimek A. Angle-based Outlier Detection in High-Dimensional Data. In: Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD'08); New York, NY, USA; 2008, pp. 444–452. https://doi.org/10.1145/1401890.1401946

[11] Zhang CK, Li SZ, Zhang H, Chen Y. VELC: A New Variational AutoEncoder Based Model for Time Series Anomaly Detection. arXiv:1907.01702; 2020. https://doi.org/10.48550/arXiv.1907.01702

[12] Hinton, GE, Salakhutdinov RR. Reducing the dimensionality of data with neural networks. Science 2006; 313 (5786): 504-507. https://doi.org/10.1126/science.1127647

[13] Kingma DP, Welling M. Auto-Encoding Variational Bayes. 2013. https://doi.org/10.48550/arXiv.1312.6114

[14] An J, Cho S. Variational Autoencoder Based Anomaly Detection Using Reconstruction Probability. Special Lecture on IE 2015; 2 (1): 1–18.

[15] Chehrazi N. Impacts of Inventory Record Inaccuracy on Retailers' Internal Operations. Available at SSRN; 2020. https://dx.doi.org/10.2139/ssrn.3637031

[16] Zadeh AH, Sharda R, Kasiri N. Inventory record inaccuracy due to theft in production-inventory systems. The International Journal of Advanced Manufacturing 2016; 83: 623–631. https://doi.org/10.1007/s00170-015-7433-3

[17] Kok AG, Shang KH. Evaluation of cycle-count policies for supply chains with inventory inaccuracy and implications on RFID investments. European Journal of Operational Research 2014; 237(1): 91–105. https://doi.org/10.1016/j.ejor.2014.01.052

[18] Zhang Y, Chen Y, Wang J, Pan Z. Unsupervised deep anomaly detection for multi-sensor time-series signals. IEEE Transactions on Knowledge and Data Engineering, 2021; 1-14. https://doi.org/10.1109/TKDE.2021.3102110

[19] Tokovarov M, Karczmarek P. A probabilistic generalization of isolation forest. Information Sciences 2022; 584: 433-449. https://doi.org/10.1016/j.ins.2021.10.075

[20] Zhang R, Zhang S, Muthuraman S, Jiang J. One class support vector machine for anomaly detection in the communication network performance data. In: Proceedings of the 5th Conference on Applied Electromagnetics, Wireless and Optical Communications; Tenerife Canary Islands, Spain; 2007. pp. 31–37. https://doi.org/10.5555/1503549.1503556

[21] Filonov P, Kitashov F, Lavrentyev A. RNN-based Early Cyber-Attack Detection for the Tennessee Eastman Process. arXiv preprint arXiv:1709.02232; 2017. https://doi.org/10.48550/arXiv.1709.02232

[22] Lindemann B, Jazdi N, Weyrich M. Anomaly Detection and Prediction in Discrete Manufacturing Based on Cooperative LSTM Networks. In: IEEE 16th International Conference on Automation Science and Engineering (CASE); Online Zoom Meeting; 2020. pp. 1003–1010. https://doi.org/10.1109/CASE48305.2020.9216855

[23] Malhotra P, Ramakrishnan A, Anand G, Vig L, Agarwal P et al. LSTM-based Encoder-Decoder for Multi-Sensor Anomaly Detection. arXiv preprint arXiv:1607.00148; 2016. https://doi.org/10.48550/arXiv.1607.00148

[24] Chakraborty D, Elzarka H. Advanced Machine Learning Techniques for Building Performance Simulation: A Comparative Analysis. Journal of Building Performance Simulation 2018; 12 (2): 193-207. https://doi.org/10.1080/19401493.2018.1498538

[25] Garcia S, Luengo J, Herrera F. Data Preparation Basic Models. In: Data Preprocessing in Data Mining. Switzerland: Springer Cham, 2015. pp. 39-57. https://doi.org/10.1007/978-3-319-10247-4

[26] Nguyen H, Tran KP, Thomassey S, Hamad M. Forecasting and Anomaly Detection Approaches using LSTM and LSTM Autoencoder Techniques with the Applications in Supply Chain Management. International Journal of Information Management 2021; 51: 1-13. https://doi.org/10.1016/j.ijinfomgt.2020.102282

[27] Metlapalli AC, Muthusamy T, Battula BP. Classification of Social Media Text Spam Using VAE-CNN and LSTM Model. Ingénierie des Systèmes d'Information 2020; 25 (6): 747-753. https://doi.org/10.18280/isi.250605

[28] Kokoska S, Zwillinger D. CRC Standard Probability and Statistics Tables and Formulae: CRC Press, 2000.

[29] Efron B. Missing Data, Imputation, and the Bootstrap. Journal of the American Statistical Association 1994; 89 (426): 463-475. https://doi.org/10.2307/2290846

[30] Chen, Tingting and Liu, Xueping and Xia, Bizhong and Wang, Wei and Lai, Yongzhi IEEE Access 2020; 47072–47081

[31] Akash D, Kanishk G, Deepak Kumar S. Chapter 1 - An introduction to deep learning applications in biometric recognition. In: Hybrid Computational Intelligence for Pattern Analysis, Trends in Deep Learning Methodologies. Academic Press, 2021. pp. 1-36. https://doi.org/10.1016/B978-0-12-822226-3.00001-5