Method Article

Sales Anomaly Detection Using Automatic Time Series Decomposition

ABSTRACT

Aims: This study proposes an automated time series decomposition (ATSD) technique for sales anomaly detection using weekly ice cream sale data from Google Trends with the caption "(Mobile AL-Pensacola (Ft. Walton Beach) FL.
Methodology: In this study, an empirical approach based on automatic time series decomposition (ATSD) was used to detect anomalies in sales data. A historical quantitative weekly ice cream sales data between periods of 9 April 2017 to 3 April 2022 consisting of 261 data points was used. The process for discovering the anomalous sales point includes importing ~~of~~ the necessary libraries, data visualization, construction of the ATSD model, creating the time series components from original data (seasonal, trend, residual), calculating the estimated from the original, and finally, the extraction of the anomalous sales.
Results: Anomalies were first discovered during the data visualization stage, which revealed some abnormal sales in mid-2019, late ~~-~~2019, and early 2020. However, after applying the ATSD, our model detected anomalies with the specific dates of these anomalies.
Conclusion: Following the application of the ATSD, our results indicate that the proposed anomaly detection approach can reliably detect anomalies with the dates of these anomalies. Because this technique worked well with our data, we assume it will work with any time series data.

Keywords: Anomaly Detection, Ice cream Sales, Automatic Time Series Decomposition

1. INTRODUCTION

In today's world, a massive quantity of data is created every day in many industries and saved in databases; as a result, there is an increasing demand for evaluating more efficient and effective software to use the information included in the data. Several methods for extracting information from data, including classification, exploratory data analysis, and clustering, are available using different software [1]. One technique to use data is to highlight abnormal behavior and identify outliers or anomalies in the data. In any organization, the sales department is critical to the company's success. The unique and critical duty of sales is to bridge the gap between the demands of potential customers and the products/services offered by the organization that can meet those needs [2].

Organizations are being pushed to deal with the surge in demand and users across their digital goods, including websites and mobile applications, as e-commerce sales continue to rise [3]. Companies that began employing anomaly detection software and technologies before the pandemic arrived were better equipped to comprehend their clients' abrupt and ever-changing demands [4]. They might better foresee the increase in e-commerce transactions and online buying, making it simpler to align goals and provide their consumers with an excellent online shopping experience [5]. Anomaly detection is critical in resilient distributed software systems because it improves communication about the system behavior, improves root cause investigation, and reduces risks to the software ecosystem. For instance, during the emergence of the COVID 19 pandemic, everyone went into a frenzy over purchasing face masks, alcohol-based hand sanitizers, and Clorox to maintain personal hygiene. The sudden rise in purchases of this nature could lead to a system jam, culminating in anomalous data. Anomaly detection technology also allows merchants to remain ahead of the curve and spot critical patterns before impacting their bottom line [6]. This paper aims to detect anomalous data based on the automatic time series decomposition. Also, since sales are attached to everyday human life, this work seeks to match the anomalous data points to their respective dates, which will apply to other time-series data.

Anomaly detection refers to any procedure that identifies outliers in a dataset, those things that do not belong there. These anomalies might indicate unexpected network activity, a faulty sensor, or simply highlight data that has to be cleaned before analysis. In today's distributed system environment, controlling and monitoring system performance is a vital responsibility. With hundreds of thousands of things to monitor, anomaly detection can assist point out where an error is occurring, improve root cause investigation, and obtain technical support fast. Anomaly detection aids in the monitoring of the source of chaos engineering by finding outliers and alerting the appropriate parties to take action. The anomalies are classified into three types: point, contextual, and collective. A point anomaly occurs when a single data point deviates from the general pattern of the data. A rapid and significant fall in the number of transactions relative to the typical range of purchases, for example, may indicate a point anomaly. A data point is considered a contextual anomaly if abnormal in a certain context [7]. A set of data points is referred to as a collective anomaly when they exhibit aberrant behavior as a group compared to the rest of the data [8]. The data points in the collective anomaly region may not be unusual individually, but their occurrence as a group may be. In the study carried out in [9], point anomalies are highlighted because rapid and fast changes in data are critical for e-commerce software platform providers to satisfy SLA (Service Level Agreements) with brands.

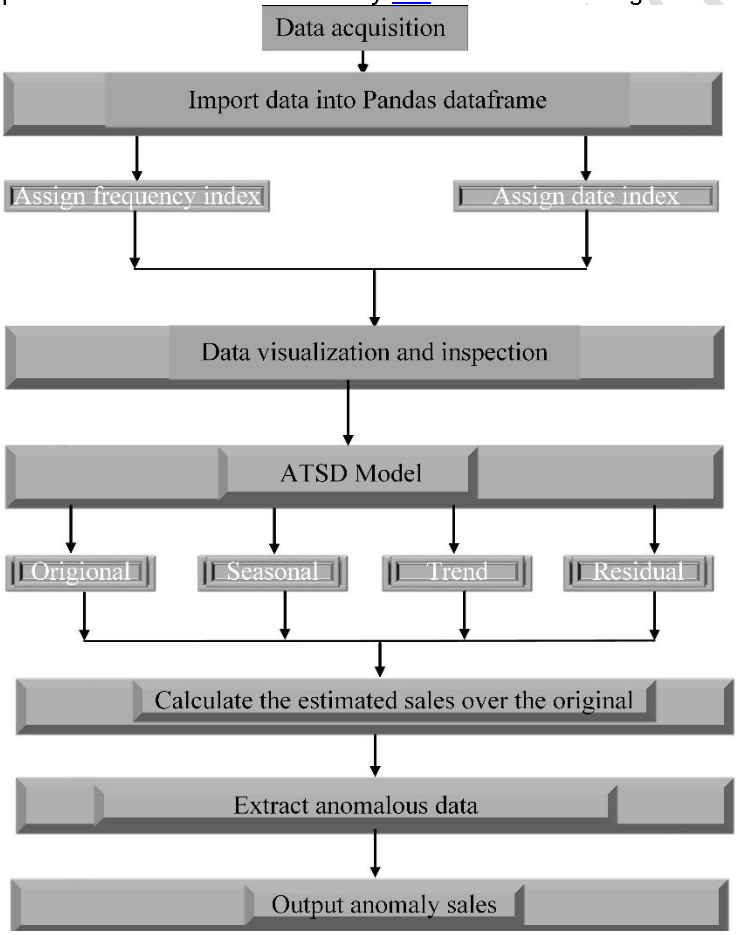
Since sales data are mostly in time series, various techniques have been deployed to detect anomalies. The Long-Short Term Memory (LSTM) network for anomaly prediction in [9] achieves the greatest F1 Scores and recall values on the test sets of both KPIs, demonstrating that LSTM suits the dynamics of e-commerce KPIs better than other time-series-~~–~~based prediction approaches. Then, in addition to the univariate analysis of the methods, it feeds the campaign information into the LSTM network because campaigns significantly affect the values of KPIs in the e-commerce domain. This information can be useful in preventing false positives that may occur during campaign periods. [10] presented K-means for hard, crisp clustering and (FCM) Fuzzy C-means for soft clustering in a large point-of-sale database for outlier detection. According to the experimental data, the K-means algorithm surpasses the (FCM) Fuzzy C-means method in terms of outlier identification efficiency, and it is an effective outlier detection solution. [11]'s thesis examined the performance of several clustering algorithms, including k-Means, DBSCAN, and OPTICS, as an anomaly detector. The data is created in R using the MixSim program. The techniques were evaluated on various cluster overlap and dimension scenarios. Evaluation criteria such as recall, precision, and F1 Score were utilized to assess the performance of clustering algorithms. The results reveal that DBSCAN outperformed other algorithms when given low-dimensional data with varied cluster overlap settings but not when given high-dimensional data with different cluster overlap settings. It concluded that, when dealing with high-dimensional data, k-means outperformed DBSCAN and OPTICS with varying cluster overlaps. A comparative review on intelligent financial fraud detection [12] focuses on computational intelligence-based techniques. In [1] and [13], anomaly detection and revenue loss estimation in accounting data methods to market anomaly detection was proposed, which detected 45 percent of anomalies.

All of the contributions to the research described above are fascinating. However, they do not give a method for matching anomalous spots to precise dates of anomaly occurrence, which is the purpose of our study.

The remaining of this paper is structured as follows: Section 2 outlines the methodology used in this study, including data acquisition and description and Automatic Time Series Decomposition; Section 3 consists of experiments results and discussion, and the conclusion on our findings is outlined in section 4.

2. METHODOLOGY

This section introduces the data and the proposed Automatic Time Series Decomposition technique. The processes for the sales anomaly are~~is~~ illustrated in Fig. 1.



~~Fig.~~Figure 1. Sales anomaly detection process

2.1 Data Description

The available Google trend (Mobile AL-Pensacola (Ft. Walton Beach) FL ice cream data was downloaded from [14]. The data consist of two columns: Colum 1 is for dates in weeks in which sales were captured; Column 2 captures the Icecream sale values. The period of the data starts from 9th April 2017 to 3rd April 2022. The length of the entire data is 261.

2.2 Automatic Time Series Decomposition

Time series decomposition is the process of dividing time series data into its essential components. These components are a possible trend (overall rise or decline in the mean), seasonality (a recurring cycle), and the remaining random residual [8],[15]. Almost all time series encountered are naturally unstable, meaning the mean, variance, or covariance will be time-dependent. Data from time series have a natural temporal ordering. This distinguishes time series analysis from other frequent data analysis tasks where there is no natural ordering of the observations (for example, explaining people's wealth relative to their education level, when the individuals' data might be input in any order). Time series analysis differs from spatial data analysis in that the observations are often related to physical places (e.g., house prices) [16]. A time series model will generally represent that data near together in time are more tightly connected than observations further away.

Furthermore, time series models frequently use time's intrinsic one-way ordering, such that values for a particular period are depicted as originating in some way from past values rather than future values. A time series is defined formally as a set of random variables indexed in time $X_1,............,X_T$. An observed time series is indicated by $X_1,............,X_T$, where the sub-index identifies the time to which the observation X_T belongs. The first observed value X_1 may be the realization of the random variable X_1, which can alternatively be expressed as $X\,(t=1,w)$, where w indicates the sample space event. Likewise, X_2 is a realization of X_2, and so on. Different probability distributions can be used to characterize the T-dimensional vector of random variables. The probability space for socioeconomic time series is continuous, while the time measurements are discrete. When observations are made daily, weekly, or monthly, the frequency of measurements is considered high, while it is said to be low when the observations are made quarterly or yearly [17]. The challenge with any time series data is determining which predictors are most useful for the time series label. All time-series data may be divided into three categories: seasonality (i.e., a recurring cyclical pattern), trend (i.e., a growing mean), and residual (random noise). Trends and seasonality are not always apparent in time-dependent data. The residual is what remains after trends and seasonality are eliminated. Time series models presume that the data is stationary and that only the residual component meets the criterion for stationarity [18]. Similar to Dukumentov's seasonal-trend regression approach stated in [19], the ATSD of multiple seasonal decompositions of a time series is presented below:

$$y_t = T_t + S_t + R_t \tag{1}$$

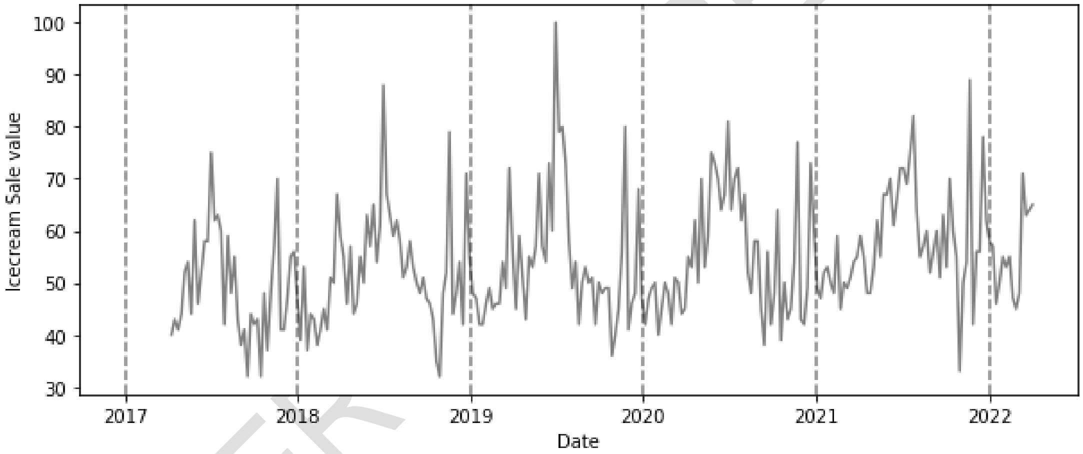
where the observed components in the series are denoted by y_t. Seasonal, trend~~,~~ and residual components are denoted by T_t , S_t, and R_t, respectively. The trend is the time series' growing and falling value. The season is the time series' recurring short-term cycle. The residual is the time series' random non-systematic fluctuation. If the latent components are dependent on one another, the correlation is described in a multiplicative model as:

$$y_t = T_t \times S_t \times R_t \tag{2}$$

3. EXPERIMENTAL RESULTS AND DISCUSSION

2.2 Experiment Results

The stats model is a Python module that provides classes and functions for estimating many different statistical models and conducting statistical tests and statistical data exploration [20]. It uses seasonality, trend, and decomposition using Loess (STL) to deconstruct a time series into three components: trend, seasonal, and residual. STL uses LOESS (locally estimated scatterplot smoothing) to compute smooth estimates for the three components. Before using the feature engineering approach, the input variables are filtered and normalized. We import several libraries such as~~:~~ Pandas, a two-dimensional, size-mutable, possibly heterogeneous tabular data. The data format also includes labeled axes (rows and columns) [21]. Arithmetic operations are aligned on both row and column labels; Matplotlib is a Python package that allows users to create static, animated, and interactive visualizations. Matplotlib makes simple things simple and difficult things possible. Creating plots of publishing quality, for example. Also, it gives the flexibility for creating dynamic figures with zoom, pan, and update, customizing the visual style and layout and exporting to a variety of file types [22]; The DateTime, a module that includes classes for modifying dates and timings [23]. While date and time arithmetic is provided, the implementation's primary focus is on efficient attribute extraction for output formatting and manipulation, including the seasonal-decompose function, which conducts automated time series decomposition (ATSD) from the python Jupyter notebook environment.

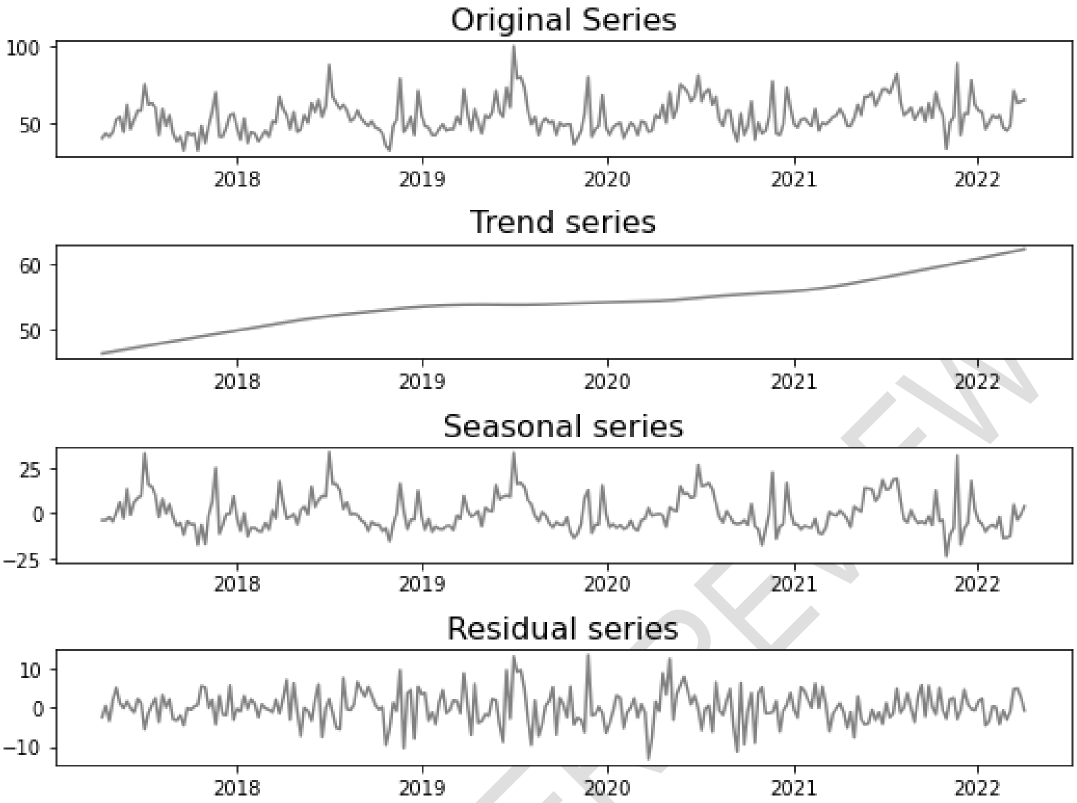


~~Fig~~Figure. 2. Visualization of Ice cream data

2.2 Discussion

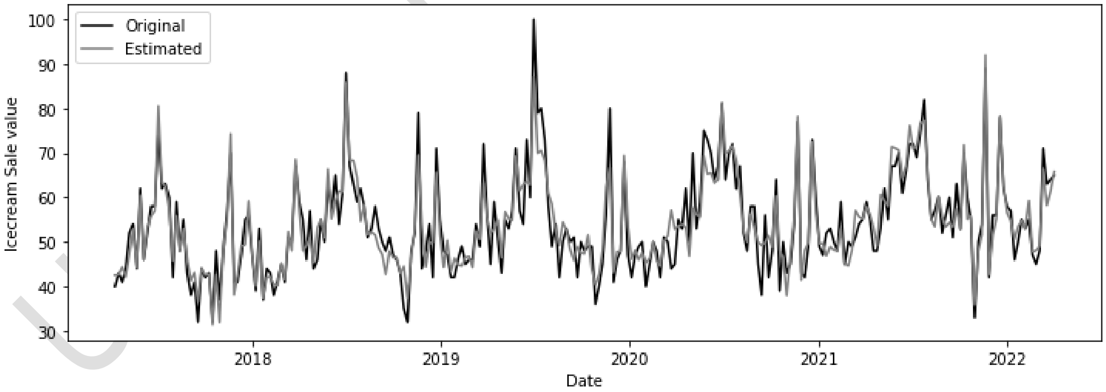
The ice cream data is loaded into the data frame and visualized. The visualization describes the data, and the presentations are mostly descriptive, focusing on "raw" data and concise summaries. Displays of changed data, often based on complex transformations, can be included, as shown in Fig. 2.

It is noticed from the visualized data that there are some surges in sales between mid-2019, late ~~-~~2019, and early 2020. The seasonal trend and the residual model were performed on the data to segregate all components, respectively, as shown in Fig. 3. Estimation was performed over the data.



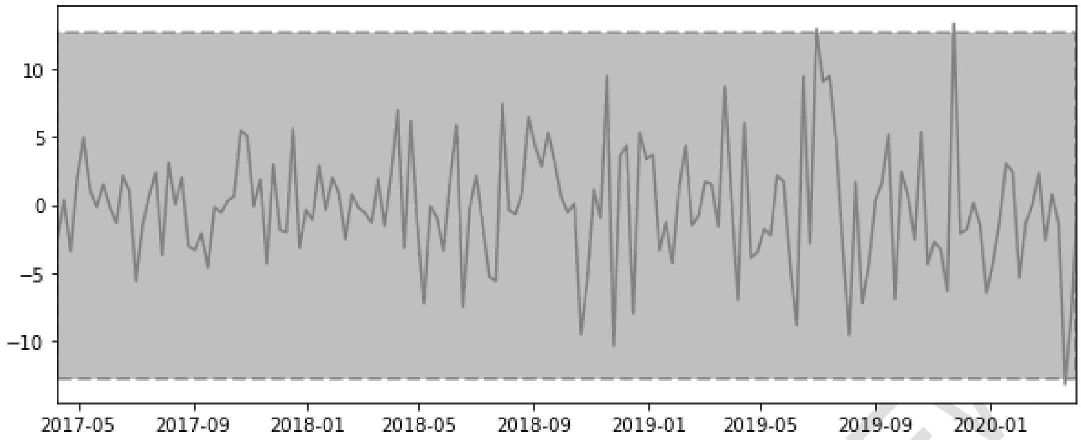
~~Fig.~~Figure 3. The Original, Seasonal, Trend, and Residual series of the data using STL

The estimator was to guess at a parameter that is not supposed to be part of the dataset. It is the sum of all the seasonal~~ and~~ trends in the data. Fig.4 shows the estimated points printed over the original data. This helps further to analyze the anomalous points in the data.



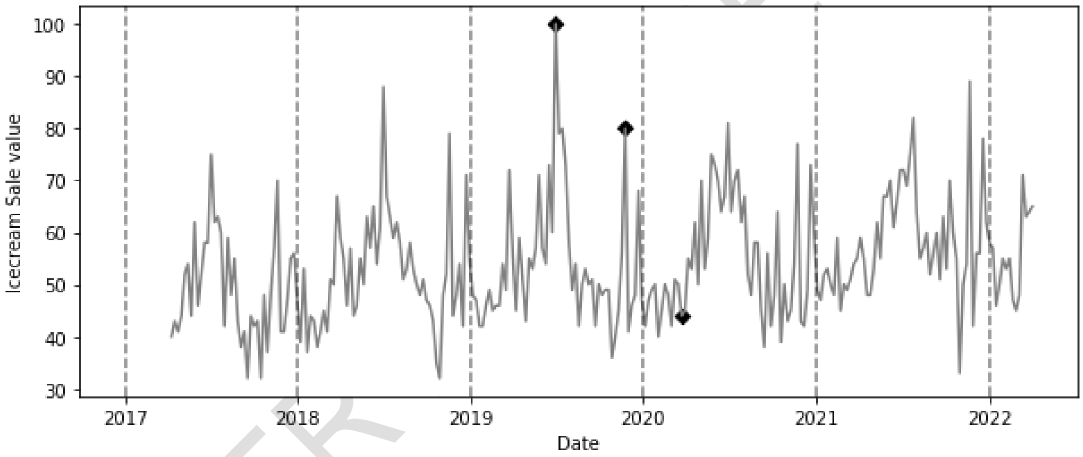
~~Fig.~~Figure 4. The original and the estimated points for the Ice cream sales data

We note that~~,~~ these abnormalities affect the model's performance as sale anomalies are seen between mid-2019, late ~~-~~2019, and early 2020. As the thumb rule in statistics, we set a lower limit threshold by subtracting three times the residual standard deviation from the residual mean. Again, we developed an upper limit threshold by adding three times the residual standard deviation to the residual mean. These steps were used to isolate outliers from the residual data, as shown in Fig.5 and the point well-identified in Fig.6.



~~Fig.~~Figure 5. Isolating abnormal points from the residual data

These three anomaly sales with their specific dates of occurrence are printed as shown in Table 1.



~~Fig.~~Figure 6. Plot showing the abnormal points

Table 1. Record of Sale anomalies

Date	Anomaly sales value
2019-06-30	100
2019-11-30	80
2020-03-22	44

4. CONCLUSION

This research proposes detecting sales anomalies using automated time series decomposition based on ice cream data from Google Trends. The first anomaly was discovered during the data visualization stage, which revealed some abnormal activity in mid-2019, late-2019, and early 2020. However, the precise date could not be determined due to probable human bias. Following the application of the ATSD, our results indicate that the proposed anomaly detection approach can reliably detect anomalies with the dates of these anomalies. Because this technique worked well with our data, we assume it will work with any time series data.

REFERENCES

1. Edholm G. Anomaly Detection and Revenue Loss Estimation in Accounting Data. Published online 2020.

2. Oxford College. The Important Role of Sales In An Organisation. Oxford College. Published 2022. Accessed April 11, 2022. https://blog.oxfordcollegeofmarketing.com/2014/10/17/the-important-role-of-sales-in-an-organisation/

3. World Trade Organization. *E-Commerce, Trade and the Covid-19 Pandemic*.; 2020. file:///C:/Users/User/Downloads/fvm939e.pdf

4. Prasad NR, Almanza-Garcia S, Lu TT. Anomaly detection. *Comput Mater Contin*. 2009;14(1):1-22.

5. Bilgihan A, Kandampully J, Zhang T (Christina). Towards a unified customer experience in online shopping environments: Antecedents and outcomes. *Int J Qual Serv Sci*. 2016;8(1):102-119. doi:10.1108/IJQSS-07-2015-0054

6. Nathaniel Meyersohn. Some grocery stores are limiting purchases of toilet paper and disinfectant wipes again. CNN Business. Published 2020. Accessed April 11, 2022. https://edition.cnn.com/2020/11/06/business/grocery-stores-toilet-paper-cleaning-wipes/index.html

7. Prasad NR, Almanza-Garcia S, Lu TT. Anomaly detection. *Comput Mater Contin*. 2009;14(1):1-22. doi:10.1145/1541880.1541882

8. Goldstein M, Uchida S. A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data. *PLoS One*. 2016;11(4):1-31. doi:10.1371/journal.pone.0152173

9. Bozbura M, Tunc HC, Kusak ME, Sakar CO. Detection of e-commerce anomalies using LSTM-recurrent neural networks. *DATA 2019 - Proc 8th Int Conf Data Sci Technol Appl*. 2019;(October):217-224. doi:10.5220/0007924502170224

10. Yoseph F, Heikkila M. A clustering approach for outliers detection in a big point-of-sales database. *Proc - Int Conf Mach Learn Data Eng iCMLDE 2019*. 2019;(February 2020):65-71. doi:10.1109/iCMLDE49015.2019.00023

11. Sreenivasulu A, Karlsson A, Kourentzes N. Evaluation of Cluster Based Anomaly Detection. Published online 2019.

12. West J, Bhattacharya M. Intelligent financial fraud detection: A comprehensive review. *Comput Secur*. 2016;57:47-66. doi:10.1016/j.cose.2015.09.005

13. Akyildirim E, Gambara M, Teichmann J, Zhou S. Applications of Signature Methods to Market Anomaly Detection. *arXiv Prepr arXiv13026613*. 2022;2(February):1-32. http://arxiv.org/abs/2201.02441

14. Cream - Explore - Google Trends. Accessed April 28, 2022. https://trends.google.com/trends/explore?date=today 5-

y&geo=686&q=%2Fm%2F01tv9

15. Ouyang Z, Ravier P, Jabloun M. STL Decomposition of Time Series Can Benefit Forecasting Done by Statistical Methods but Not by Machine Learning Ones. Published online 2021:42. doi:10.3390/engproc2021005042

16. Dagum EB. TIME SERIES MODELING AND DECOMPOSITION. *Statistica*. 2010;1:1-8. https://www.researchgate.net/publication/307663962_Time_Series_Modelling_and_Decomposition

17. SAS Institute Inc. (1993), SAS/ETS User's Guide, Version 6, Second Edition, Cary, NC: SAS Institute Inc. Accessed June 7, 2022. https://support.sas.com/rnd/app/ets/examples/expa/index.htm

18. Makridakis S, Spiliotis E, Assimakopoulos V. The M4 Competition: 100,000 time series and 61 forecasting methods. *Int J Forecast*. 2020;36(1):54-74. doi:10.1016/j.ijforecast.2019.04.014

19. Oliveira DHL, Filho FMV, De Araújo TP, Celestino J, Gomes RL. Adaptive Model for Network Resources Prediction in Modern Internet Service Providers. In: *Proceedings - IEEE Symposium on Computers and Communications*. Vol 2020-July. ; 2020:1-7. doi:10.1109/ISCC50000.2020.9219550

20. Seabold S, Perktold J. statsmodels: Econometric and statistical modeling with python. 9th Python in Science Conference. Published 2010. Accessed June 8, 2022. https://www.statsmodels.org/stable/index.html

21. pandas.DataFrame. pandas 1.4.2 documentation. Accessed June 9, 2022. https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.html

22. Matplotlib. Visualization with Python. Accessed June 9, 2022. https://matplotlib.org/

23. D~~d~~atetime. Basic date and time types — Python 3.10.5 documentation. Accessed June 9, 2022. https://docs.python.org/3/library/datetime.html