Detecting real-time anomalies in stock time series data

Ho Thanh Duy Khanh[1,2], Nguyen Thi Nguyet[1,2], Nguyen Thanh Nhan[1,2], Nguyen Thi Phuong Thao[1,2], and Do Trong Hop[1,2]

[1] VNUHCM - University of Information Technology, Viet Nam
[2] 20521445, 20521689, 20521701, 20521936@gm.uit.edu.vn, hopdt@uit.edu.vn

Abstract. Detecting real-time anomalies in stock time series data poses a significant challenge in the financial domain due to the unavailability of labeled data for identifying trading anomalies or stock price movements. This study introduces a real-time anomaly detection system for stock data from Apple Inc., collected over five weeks using a financial API, recording minute-level stock price information during trading hours. Unsupervised anomaly detection models, including Isolation Forest, Spectral Residual, Local Outlier Factor, Random Cut Forest, and Luminol, are employed to create label functions for each data point. We construct a weakly supervised learning model based on the composite label functions to provide weak labels. Subsequently, we train a classification model using LightGBM and evaluate five basic machine learning models (SVM, XGBoost, KNN, Random Forest, Gradient Boosting Machine) to detect anomalies in the data. The results show that data balancing through Synthetic Minority Over-sampling Technique (SMOTE) significantly improves the anomaly detection capability, particularly for the Random Forest model. For the online component, real-time data is collected from the Binance platform and processed through Kafka and Spark Streaming. The trained offline model is then utilized to predict anomalies in the streaming data. The proposed system offers a comprehensive approach to real-time anomaly detection in stock data, providing crucial support for investors and financial experts in making timely and informed decisions.

Keywords: Real-time anomaly detection · Stock time series data · Weak supervision · Active learning · Kafka · Spark Streaming

1 Introduction

In the dynamic world of finance, making informed investment decisions and seizing profitable opportunities are crucial goals for every investor. However, the stock market is becoming increasingly complex and ever-changing, with millions of transactions taking place every second. Unpredictable fluctuations in stock data can lead to significant shifts in stock prices. Therefore, detecting anomalies in real-time stock data has become a formidable and essential challenge in the financial domain.

To address this challenge, we embark on the research topic of "Detecting real-time anomalies in stock time series data". Our main objective is to develop an automated and efficient system for detecting unusual patterns in real-time stock data. To achieve this, we employ a combination of unsupervised machine learning, weakly supervised learning, active learning, and experimentation with supervised machine learning models. This approach allows us to identify significant fluctuations in stock prices without the need for pre-labeled data. By automatically pinpointing and evaluating noteworthy changes in stock data, our system empowers investors to timely recognize potential opportunities and risks in the market.

Moreover, we employ a range of machine learning models and advanced data preprocessing methods to construct a reliable classification system. By integrating real-time stock data with advanced data processing techniques, our research promises to offer a comprehensive and effective solution for anomaly detection in real-time stock data. The anticipated outcomes of this study hold significant potential benefits for both investors and financial experts, empowering them to make swift and intelligent investment decisions in the face of a rapidly evolving market environment.

To tackle this challenge, our research introduces a real-time anomaly detection system, which consists of two main components:

1. **Offline**: In this component, we develop and train an anomaly detection model using historical data with labeled anomalies. This offline model serves as the foundation for identifying abnormal patterns and establishing the basis for anomaly detection.
2. **Online**: The second component involves the implementation of a real-time anomaly detection system. This system is designed to continuously process incoming data in real-time, making it capable of detecting anomalies as they occur in livestock time series data.

2 Components offline

2.1 Dataset

a. Collect Data

We collected the stock price data of Apple (stock symbol: AAPL) within the timeframe from June 12, 2023, to July 14, 2023 (5 weeks) using a financial API. The data was collected at a 1-minute frequency (interval='1min') and only during trading hours, from 4:00 AM to 7:59 PM. No data was collected on non-trading days, such as Saturdays and Sundays.

Through the financial API, we obtained 21,972 rows of data with 6 columns, providing detailed information about the real-time trading timestamps, opening price, highest price, lowest price, closing price, and trading volume of the Apple stock in each trading session. This data was delivered with accuracy and real-time precision, enabling us to capture significant fluctuations and patterns in Apple's stock price during the research period.

b. Preprocessing data

After the data collection phase, a meticulous data preprocessing process was executed to ensure data integrity and facilitate further analysis. This essential preprocessing involved several crucial steps, including column renaming, temporal data sorting, conversion of the 'Date' column to the appropriate datetime format, the creation of a coherent time series, and a meticulous filtering procedure to retain only the trading days and hours.

Furthermore, to address the issue of missing data points, we employed two prominent data imputation methods, namely the "Linear Interpolation" and "Effective Dynamic Time Warping Based Imputation" techniques, as referenced in the scholarly work titled "An Empirical Study of Imputation Methods for Univariate Time Series" [13]. These methods were skillfully applied to the collected Apple stock price dataset, which exhibited varying degrees of missing values on different dates.

Percent of missing values	Method	Sim	FSD	RMSE
3%	na,interp	0.93	1.29	63,647,034.23
	eDTWBI	0.9	0.2	91,136,047.83
5%	na,interp	0.95	1.33	60,257,050.62
	eDTWBI	0.92	0.18	97,770,975.79
10%	na,interp	0.98	0.06	34,553,816.07
	eDTWBI	0.83	1.13	238,341,687.25
20%	na,interp	0.97	1.07	75,588,405.84
	eDTWBI	0.88	0.43	247,539,223.19
30%	na,interp	0.96	0.07	89,243,836.36
	eDTWBI	0.95	0.76	143,058,174.64

Table 1: The result of imputing missing values from 3/1/2005 to 14/7/2023

Based on the imputation results mentioned in Table 1, Linear Interpolation yielded more favorable outcomes than the eDTWBI method across various levels of missing data. Consequently, this method closely approximates reality and proves to be more effective in reconstructing real-time temporal data for the Apple stock price dataset. Building upon this foundation, we have made the decision to employ the Linear Interpolation method for filling in the missing data points in the minute-level stock price dataset. Following the data preprocessing, we obtained a dataset with 24,000 samples and 6 features.

2.2 Abnormal detection

a. Labeling data

In the absence of labeled anomaly data, we employed a well-defined pipeline outlined in the scholarly work titled "Label-Efficient Interactive Time-Series Anomaly Detection"[7] to construct a sophisticated model capable of effectively detecting anomalies within the stock price data of Apple.

The procedure entailed the following steps:

Step 1: In the initial phase, we generated distinct label functions (LFs) leveraging five prominent unsupervised anomaly detection models: Isolation Forest[10][9], SR (Spectral Residual)[16], LOF (Local Outlier Factor)[2], RC-Forest (Random Cut Forest)[5], and Luminol. Each model provided predicted labels of {-1, 0, 1} for every data point in the time series. Here, {-1, 0, 1} denoted abstain, normal, and anomalous, respectively.

Step 2: Considering that the labels acquired from the unsupervised models might not be entirely precise, we proceeded with the development of a weak supervision model based on the renowned snorkel[15] theory. This novel model synthesized weak labels by aggregating the predictions from the LFs obtained in the preceding step. The weak labels were inferred by calculating probability thresholds for positive class membership for each data sample within the LFs, resulting in predicted labels also being classified as {-1, 0, 1}.

Step 3:

We constructed a classification model using LightGBM[8] based on the binary labels 0, 1 obtained in Step 2. LightGBM relies on GBDT (Gradient Boosting Decision Tree) and iteratively trains weak classifiers, i.e., decision trees, to obtain an optimized model. LightGBM introduces optimization techniques to improve time and space efficiency compared to XGBoost without compromising accuracy, making it a popular choice in large-scale data competitions.

The feature extraction techniques and methods were derived from the SR-DNN[17] time-series anomaly detection model, which is a state-of-the-art supervised model for univariate anomaly detection. Features were computed from the dimensionally-reduced data using Principal Component Analysis (PCA) [11]. This process involved calculating statistical parameters such as mean, standard deviation, percentile ranks, and transformations like logarithm, moving average, and exponential weighted moving average. The aim was to generate features that represent the characteristics and trends of the data within various time windows. These extracted features were combined with the prediction results from the 5 standards Unsupervised Anomaly Detection models to create a comprehensive feature set used as input for our model.

Step 4: We constructed an active learning model to propose positions for human labeling to replace the initial labels synthesized from the UAD models. These positions were determined based on uncertainty calculations from the LightGBM model's results and information from the Label Functions. For the human labeling part, our team lacked sufficient expertise in the stock market domain to determine which data points are anomalous or not. Therefore, the labels assigned by humans were based on the Three-Sigma Limits method [14] using data from the most recent 1-week period to identify anomalous data points.

This method involves statistical calculations considering data points within 3 standard deviations from the mean value.

Step 5: We create a new Label Function (LF) based on the labels assigned by humans. This new LF is then added to the existing LFs and fed into the weak supervision model once again (Step 2). The stopping condition is set to terminate the process after two iterations if the human labelers fail to identify any anomalies predicted by LightGBM.

Ultimately, this iterative process allows us to obtain the final model - LightGBM, which demonstrates a robust capability to perform well on data where human intervention in labeling is present. Utilizing this model, we construct a comprehensive label set for the dataset, catering to the anomaly detection task in the Apple stock trading data.

Through this comprehensive approach, we have successfully created a model that effectively combines the strengths of both unsupervised and supervised learning, leveraging human-in-the-loop input to achieve high accuracy in detecting anomalies within the Apple stock trading data.

b. Model We utilized five fundamental binary classification models in our approach:

- ***Support Vector Machine (SVM)***[4]: A widely used machine learning model for classification, SVM is a powerful model capable of handling both linear and nonlinear data. It constructs hyperplanes or boundaries to separate data points belonging to different labels. SVM excels in solving nonlinear classification problems and can apply kernel functions to enhance model accuracy.
- ***eXtreme Gradient Boosting (XGBoost)***[3]: XGBoost is a robust gradient boosting algorithm that combines multiple weak decision trees into a stronger model. It employs weak decision trees as local learners to create a powerful ensemble model. The training process of XGBoost involves finding the optimal decision trees by minimizing the loss function. Additionally, XGBoost employs regularization techniques to avoid overfitting and improve the model's generalization ability. Once trained, the XGBoost model can be used to predict labels for new data points.
- ***Random Forest***[1]: A type of ensemble model based on the concept of decision trees, Random Forest combines multiple simple decision trees into a forest. Each decision tree in the Random Forest is trained on a random subset of data and a random set of features from all available features. This process helps reduce overfitting and creates a more generalized and stable model.
- ***K-nearest neighbors (KNN)***[6]: A simple and popular machine learning algorithm for classification and prediction tasks. KNN classifies new data points by finding and using the nearest data points from the training set. The basic idea of KNN is based on the "nearest neighbor principle." This means that if a new data point has nearby data points in its feature space

that belong to a specific class, there is a high probability that the new data point also belongs to that class.
- ***Gradient Boosting Machine***[12]: One of the powerful and popular machine learning algorithms used for prediction and classification tasks. GBM is an ensemble model, meaning it combines multiple weak models to create a stronger one. The fundamental idea behind GBM is to build decision trees iteratively and use the residuals from the previous tree to improve the subsequent trees.

c. Result

Measure

Using the evaluation metrics Accuracy, Precision, Recall, and F1-score to assess the classification performance of the models, the expressions are as follows:
Accuracy:

$$\text{Accuracy} = \frac{\text{Num of correct predictions}}{\text{Total num of predictions}}$$

Precision:

$$\text{Precision} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Positives (FP)}}$$

Recall:

$$\text{Recall} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Negative (FN)}}$$

F1-score:

$$F1\text{-score} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

Results

The experimental results on the dataset before and after data balancing are as follows

Based on tabel 2 and tabel 3, the following observations can be made:

Before applying data balancing, all models yielded relatively high Accuracy and F1-score for the "Normal" class, but they encountered challenges in classifying anomaly points (the "Anomaly" class) with low Recall and F1-score. This indicates that addressing imbalanced data is essential to enhance the models' performance in anomaly detection. The team employed the Synthetic Minority Over-sampling Technique (SMOTE) [3.15] for data balancing.

After data balancing, significant improvements were observed in the models' ability to detect the "Anomaly" class. For instance, the recall of the "Anomaly" class increased notably, indicating the ratio of correctly predicted anomalies over the total actual anomalies. The precision of the "Anomaly" class also showed a considerable increase, representing the ratio of correctly predicted anomalies

		SVM	XGBoost	R.Forest	KNN	GBM
Precison	Normal	0.9876	0.9895	0.9895	0.9873	0.9894
	Anomaly	1.0	0.5714	0.6809	0.7692	0.8158
Recall	Normal	1.0	0.9975	0.9984	0.9997	0.9993
	Anomaly	0.0985	0.2424	0.2424	0.0758	0.2348
F1-Score	Normal	0.9938	0.9935	0.9940	0.9934	0.9943
	Anomaly	0.1793	0.3404	0.3575	0.1379	0.3647
	Accuracy	0.9876	0.9871	0.9880	0.9870	0.9888

Table 2: Anomaly detection results before processing data imbalance

		SVM	XGBoost	R.Forest	KNN	GBM
Precison	Normal	0.9952	0.9954	0.9964	0.9922	0.9981
	Anomaly	0.0483	0.1118	0.1161	0.0397	0.0940
Recall	Normal	0.8024	0.9227	0.9188	0.8185	0.8819
	Anomaly	0.7197	0.6970	0.7652	0.5379	0.8788
F1-Score	Normal	0.8884	0.9577	0.9560	0.8970	0.9364
	Anomaly	0.0906	0.1927	0.2016	0.0739	0.1698
	Accuracy	0.8013	0.9197	0.9167	0.8147	0.8819

Table 3: Anomaly detection results after processing data imbalance by SMOTE

over the total predicted anomalies as anomalies. Despite the improvements in recall and precision for the "Anomaly" class after data balancing, the models still exhibited deficiencies in detecting this class, particularly in the SVM and K-nearest neighbors (KNN) models. The F1-score for the "Anomaly" class in these models remained low, indicating that the trade-off between recall and precision has not reached an optimal level.

Among the considered models after data balancing, the Random Forest model demonstrated the best performance in detecting the "Anomaly" class, with higher recall and F1-score for the "Anomaly" class compared to the other models after data balancing. Considering the primary goal of detecting anomalies, the team chose to use the Random Forest model after data balancing to apply it to real-world data for anomaly detection.

3 Components online

We perform real-time data retrieval from Binance's website, followed by anomaly detection using the pre-trained model from the offline phase. Subsequently, we visualize the results on Streamlit using the Python programming language. The pipeline for our process is as follows:

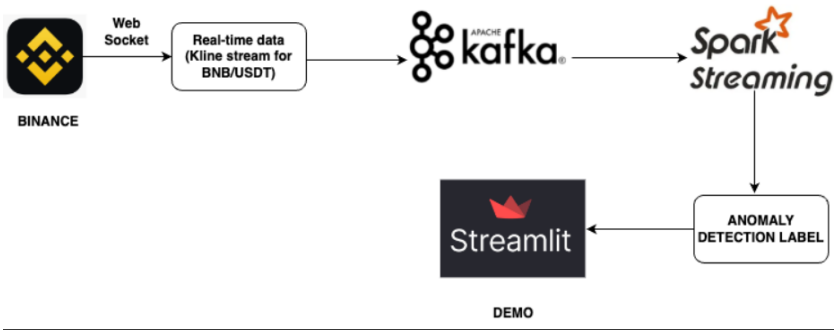


Fig. 1: Pipeline for system implementation

3.1 Real-time data collection

Using Binance's API, we fetch the WebSocket link for the BNB/USDT trading pair with a 3-minute interval for Kline (candlestick) data. After obtaining real-time data, we proceed to filter out the necessary attributes for the problem, including "Time", "Open", "High", "Low", "Close", and "Volume".

3.2 Processing stream data

We set up Kafka in the local environment, including one Zookeeper, one topic, and one broker. We proceed to configure some fields:

bootstrap.servers='localhost:9092'
listeners=PLAINTEXT://localhost:9092
advertised.listeners=PLAINTEXT://localhost:9092

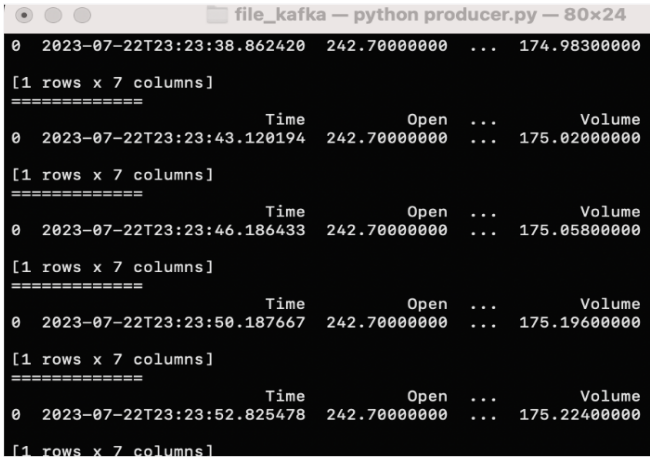
- "listeners" refers to the configuration for Kafka Broker, where the Kafka Broker listens for connections from producers and consumers.
- "advertised.listeners" is the configuration for Kafka Broker to advertise the address and port that producers and consumers can connect to.

When feeding the data stream into the producer, we convert the data type to JSON format using 'json.encode'. Subsequently, when sending the data to the consumer, the team performs the conversion back to its original format using 'json.decode'.

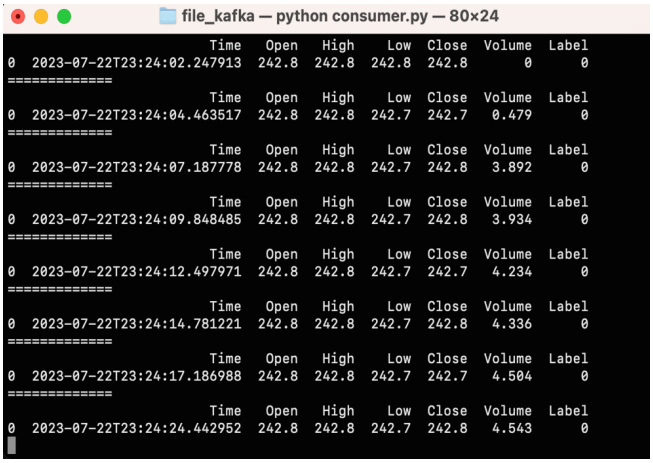
3.3 Spark Streaming

We read data from Kafka for analysis and processing. Firstly, we select the best-performing pre-trained model from the offline phase to use for anomaly detection on real-time data. Then, we preprocess the data to ensure it is suitable for the selected trained model. In this step, we noticed that Random Forest, after data

balancing using the SMOTE method, provided the best predictions for anomaly and non-anomaly labels. After predicting the labels for real-time data, we save the prediction results back to Kafka. Observe the real-time anomaly detection process through figure 2.



(a) Data Sent to Producer



(b) Data Received by Consumer

Fig. 2: Real-time anomaly detection

3.4 Web Apps

Retrieve the data stored in the Kafka topic and deploy the system on the Streamlit application, consisting of one DataFrame displaying the data fields and labels

for each real-time data point, along with a line chart illustrating the anomalous variations of the data over time. Observe this process through Figure 3.

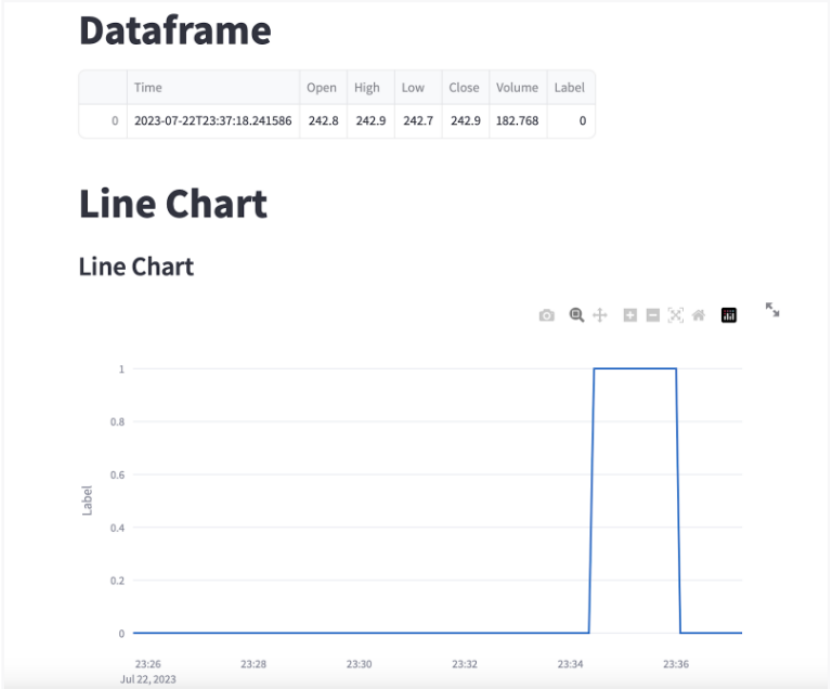


Fig. 3: Real-time Anomalous Data Demo

4 Conclusion and future work

We have successfully built a real-time anomaly detection system for stock market data, which includes data preprocessing, missing data imputation, research, and application of automated labeling methods. The system has been integrated into real-time processing, deployed on the Streamlit application, and utilizes several supervised learning models along with data balancing techniques. This work contributes to providing valuable information and functionalities for investors and financial experts, enabling them to analyze and make intelligent and effective decisions based on anomaly detection results.

Although the real-time anomaly detection system for stock market data has achieved significant accomplishments, it still faces some limitations:

– The data used in this study was collected over a short period, lasting only 5 weeks. The short-term data collection may not fully reflect the overall

variations and trends of real-world stock market data. This could reduce the representativeness and accuracy of the anomaly detection model.
- Despite active learning methods reducing labeling efforts, it still requires a high level of domain expertise and faces challenges in labeling data in the financial domain, such as identifying causes and factors influencing prices. To ensure the accuracy and effectiveness of the model, labeling needs to be carefully done with support from financial experts.
- The use of only a few simple machine learning models may limit the diversity and detection capabilities of the system. Additionally, although we have experimented with the SMOTE data balancing method, testing only one method may not be sufficient to evaluate its effectiveness and pros and cons compared to other methods.

In the future, we hope to develop a Telegram bot that can send information to users when anomalies are detected, providing a convenient and efficient experience for users to monitor stock market fluctuations.

References

[1] L Breiman. "Random Forests". In: *Machine Learning* 45 (Oct. 2001), pp. 5–32. DOI: 10.1023/A:1010950718922.

[2] Markus M. Breunig et al. "LOF: Identifying Density-Based Local Outliers". In: *SIGMOD Rec.* 29.2 (2000), 93–104. ISSN: 0163-5808. DOI: 10.1145/335191.335388. URL: https://doi.org/10.1145/335191.335388.

[3] Tianqi Chen and Carlos Guestrin. "XGBoost". In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2016. DOI: 10.1145/2939672.2939785. URL: https://doi.org/10.1145%2F2939672.2939785.

[4] Theodoros Evgeniou and Massimiliano Pontil. "Support Vector Machines: Theory and Applications". In: vol. 2049. Sept. 2001, pp. 249–257. ISBN: 978-3-540-42490-1. DOI: 10.1007/3-540-44673-7_12.

[5] Sudipto Guha et al. "Robust random cut forest based anomaly detection on streams". In: *ICML 2016*. 2016. URL: https://www.amazon.science/publications/robust-random-cut-forest-based-anomaly-detection-on-streams.

[6] Gongde Guo et al. "KNN Model-Based Approach in Classification". In: (Aug. 2004).

[7] Hongmei Guo et al. "Label-Efficient Interactive Time-Series Anomaly Detection". In: *ArXiv* abs/2212.14621 (2022).

[8] Guolin Ke et al. "LightGBM: A Highly Efficient Gradient Boosting Decision Tree". In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon et al. Vol. 30. Curran Associates, Inc., 2017.

[9] Fei Tony Liu, Kai Ting, and Zhi-Hua Zhou. "Isolation-Based Anomaly Detection". In: *ACM Transactions on Knowledge Discovery From Data - TKDD* 6 (Mar. 2012), pp. 1–39. DOI: 10.1145/2133360.2133363.

[10] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. "Isolation Forest". In: *2008 Eighth IEEE International Conference on Data Mining*. 2008, pp. 413–422. DOI: 10.1109/ICDM.2008.17.

[11] Andrzej Maćkiewicz and Waldemar Ratajczak. "Principal components analysis (PCA)". In: *Computers Geosciences* 19.3 (1993), pp. 303–342. ISSN: 0098-3004. DOI: https://doi.org/10.1016/0098-3004(93)90090-R. URL: https://www.sciencedirect.com/science/article/pii/009830049390090R.

[12] Alexey Natekin and Alois Knoll. "Gradient Boosting Machines, A Tutorial". In: *Frontiers in neurorobotics* 7 (Dec. 2013), p. 21. DOI: 10.3389/fnbot.2013.00021.

[13] Thi-Thu-Hong Phan. "An Empirical Study of Imputation Methods for Univariate Time Series". In: 19 (Apr. 2021), pp. 452–461.

[14] Friedrich Pukelsheim. "The Three Sigma Rule". In: *The American Statistician* 48.2 (1994), pp. 88–91. URL: https://doi.org/10.2307/2684253 (visited on 07/23/2023).

[15] Alexander Ratner et al. "Snorkel: rapid training data creation with weak supervision". In: *The VLDB Journal* 29 (May 2020). DOI: 10.1007/s00778-019-00552-1.

[16] Hansheng Ren et al. "Time-Series Anomaly Detection Service at Microsoft". In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (2019).

[17] Hansheng Ren et al. "Time-Series Anomaly Detection Service at Microsoft". In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. KDD '19. Anchorage, AK, USA: Association for Computing Machinery, 2019, 3009–3017. ISBN: 9781450362016. DOI: 10.1145/3292500.3330680. URL: https://doi.org/10.1145/3292500.3330680.