International Journal on Cybernetics & Informatics (IJCI) Vol.13, No.2, April 2024

Enhancing Time Series Anomaly Detection: A Hybrid Model Fusion Approach

Paul Addai[1] and Tauheed Khan Mohd

[1] Dept of Math and Computer Science, Augustana College, Rock Island, Illinois, USA

[2] School of Information Security and Applied Computing, Eastern Michigan University, Ypsilanti, Michigan, USA

Abstract. Exploring and Identifying Anomalies in time-series data is very crucial in today's world revolve around data. These data are being used to make important decisions; hence, an efficient and reliable anomaly detection system should be involved in this process to ensure that the best decisions are being made.

The paper explores other types of anomalies and proposes efficient detection methods which can be used. Anomalies are patterns that deviate from usual expected behavior. These can come from system failures or unexpected activity. This research paper explores the vulnerabilities of commonly used anomaly detection algorithms such as the Z-Score and static threshold approach. Each method used in this paper has its unique capabilities and limitations. These methods range from using statistical methods and machine learning approaches to detecting anomalies in a time-series dataset. Furthermore, this paper explores other open-source libraries that can be used to detect anomalies, such as Greykite and Prophet Python library.

This paper serves as a good source for anyone new to anomaly detection and willing to explore.

Keywords: anomaly · forecasting · time series · data · libraries · machine learning

1 Introduction

Finding anomalous events or behaviors is an important responsibility in a variety of industries, such as manufacturing, banking, healthcare, security, and the healthcare industry. Doing so can significantly increase operational effectiveness, risk management, and customer satisfaction. Since time series data capture the temporal dependencies and patterns in the underlying system, they are especially helpful for spotting anomalies. Time series data are sequential observations of a variable across time. Data is all around us today due to the number of new technologies invented which allows data collection to be easy. Making decisions from this data can help companies save or gain a lot of money based on the kind of situation they are dealing with. Due to the increasing use of time series data in

Bibhu Dash et al: NLPAI, IOTBC, CITE -2024
pp. 135-147, 2024. IJCI – 2024
DOI:10.5121/ijci.2024.130210

many applications and the demand for scalable and precise solutions, there has been an increase in interest in the development of automated anomaly detection systems. The efficiency of these strategies, which range from basic statistical methods to complex machine learning algorithms, depends on the properties of the data and the particular use case.

This paper focuses specifically on time series data and the numerous ways to detect anomalies in this data. Time series data is also referred to as time-stamped data which is a sequence of data points indexed in time order. Examples of this include; stock prices, CPU temperature/usage, monthly subscribers, and memory usage. Today, a lot of techniques such as machine learning and statistical models have been created to tackle this problem. Time series data can be very dynamic and thus change over time, which makes it hard to find a "silver bullet" algorithm to detect every anomaly. These algorithms or models also make false decisions which can be very annoying. There are a lot of Python libraries for time series anomaly and forecasting or prediction.

The remainder of this document is organized as follows: Section 2 presents an overview of the related work. Section 3 details some of the most common anomaly detection models or techniques. Section 4 reveals the method used to detect and enhance the overall performance of the anomaly detection model. Section 5 discusses how well the proposed approach performed under a tricky anomaly situations. Section 6 presents the conclusions.

2 Related Work

Microsoft uses a new algorithm for time series anomaly detection which helps customers to monitor the time series continuously and alert for potential incidents on time [1]. Their algorithm pipeline consists of three main modules including the data ingestion module, experimentation platform, and online compute module as shown in 1. The online compute module is responsible for performing anomaly detection after receiving data from the data ingestion module (ingested in Kafka and fluxDB). The experimentation platform evaluates the performance of anomaly detection models after being processed in the online compute module. The algorithm is based on Spectral Residual (SR) and Convolutional Neural Network (CNN). Spectral Residual (SR) is an efficient unsupervised algorithm, which demonstrates outstanding performance and robustness in the visual saliency detection tasks. The Spectral Residual (SR) algorithm consists of three major steps: (1) Fourier Transform to get the log amplitude spectrum; (2) calculation of spectral residual; and (3) Inverse Fourier Transform that transforms the sequence back to the spatial domain. This is the first attempt to apply an SR model from the visual saliency detection domain to time-series anomaly detection. Anomaly can come in different forms. These include Global, Contextual, and Collective outliers.

A research by Aphichit Hanbanchong; Krerk Piromsopa [2] used SARIMA to detect anomalies in network bandwidth which makes a significant addition to the field anomaly detection. To maintain good network performance and avail-

ability, the authors expand on previous research that highlights the necessity of robust network monitoring and anomaly detection. The suggested method makes use of a seasonal autoregressive integrated moving average (SARIMA) model to find anomalies in network bandwidth, which can assist network administrators in spotting and resolving network performance degradation before it results in system failure or downtime. Additionally, they compared it with machine learning-based techniques like support vector machines and artificial neural networks (ANNs) (SVMs). The findings demonstrate the SARIMA-based method's superior accuracy and speed over the competition, highlighting its potential for real-time network anomaly identification. The authors discuss related work in the area of anomaly detection, including statistical-based, machine learning-based, and hybrid approaches. They point out that as network traffic data frequently displays complicated and non-linear patterns, conventional statistical-based techniques like moving averages and autoregressive models would not be useful for spotting anomalies. They also draw attention to the drawbacks of machine learning-based methods, which may need a lot of labeled data and have problems with overfitting and weak generalization. Similar to this research, another research work titled "Comparing Prophet and Deep Learning to ARIMA in Forecasting Wholesale Food Prices" [3] talks about time-series forecasting which can also be used to detect anomalies. This can be done by finding the mean square error of the actual and predicted data points.

The paper "An overview of anomaly detection techniques: Existing solutions and latest technological trends" by Animesh Patcha and Jung-Min Park [4] provides a comprehensive survey of the existing anomaly detection techniques and the latest trends in this area. The authors review various approaches for anomaly detection in different domains, including network intrusion detection, sensor data analysis, financial fraud detection, and healthcare monitoring. The paper starts by defining the concept of anomaly detection and discussing the key challenges and requirements for effective detection. It then categorizes the anomaly detection techniques into three main groups: statistical methods, machine learning methods, and hybrid methods that combine both. The authors describe the advantages and limitations of each group and provide examples of the most commonly used techniques, such as clustering, classification, regression, and neural networks.

One of the strengths of this paper is that it covers not only the traditional methods but also the latest technological trends in anomaly detection. The authors discuss the emerging techniques based on deep learning, such as autoencoders, recurrent neural networks, and generative adversarial networks, and highlight their potential for improving the accuracy and scalability of anomaly detection. The paper also discusses the challenges and open research issues in anomaly detection, such as handling imbalanced and incomplete data, dealing with concept drift and contextual information, and ensuring the interpretability and explainability of the models. The authors suggest some possible directions for future research and emphasize the importance of benchmarking and evaluating the performance of anomaly detection methods on real-world datasets.

Overall, this paper provides a valuable resource for researchers and practitioners interested in anomaly detection, by summarizing the state-of-the-art techniques and highlighting the latest trends and challenges in this field.

In their paper "Anomaly detection: A survey", Varun Chandola, Arindam Banerjee, and Vipin Kumar [5] provide a comprehensive survey of various techniques for anomaly detection across multiple domains. The authors highlight the importance of anomaly detection as a critical task for detecting outliers, identifying suspicious activities, and preventing fraud in various domains, including finance, healthcare, and cybersecurity. The paper begins by defining the problem of anomaly detection and discussing its significance in various applications. The authors then provide a taxonomy of various anomaly detection techniques, including statistical methods, machine learning-based approaches, and data mining techniques. They discuss the advantages and limitations of each approach and highlight the challenges in selecting an appropriate technique for a specific application. The paper also discusses various evaluation metrics for anomaly detection and provides a comparison of different techniques based on these metrics. Additionally, the authors highlight some of the emerging trends in anomaly detection, including deep learning-based approaches and techniques for handling high-dimensional and complex data. Overall, the paper provides a comprehensive review of the state-of-the-art techniques in anomaly detection and serves as a valuable resource for researchers and practitioners in the field. The authors' insights and recommendations can aid in the selection and implementation of effective anomaly detection techniques in various domains.

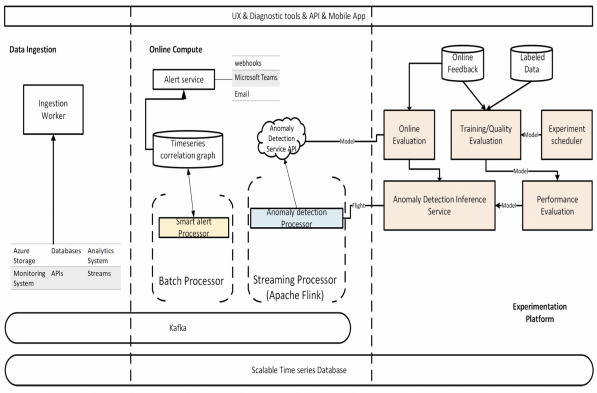

Fig. 1. Microsoft's Anomaly System Overview

3 Anamoly Detection Techniques

Please note that for the forecasting method, the data was first filtered using percentiles to remove any anomalies from the data before training. This is to help the model learn the actual pattern or structure of the data in its normal form.

3.1 LSTM Neural Network (Long Short-Term Memory)

The LSTM neural network is one of the common deep-learning models for anomaly detection. LSTM stands for Long Short-Term Memory, which is a type of recurrent neural network (RNN) architecture designed to address the problem of vanishing gradients in traditional RNNs. These neural networks are effective in capturing long-term dependencies in the data, which is difficult for other types of neural networks to do.

3.2 Z-Score (Statistics)

The Z-Score equation can also be used to detect anomalies using the average and standard deviation of a data point. The equation is given as $(x - avg)/standard_deviation$. This technique works quite well, but a static threshold needs to be set to help classify a data point as an anomaly. The Z-score is able to detect anomalies by identifying data points that deviate significantly from the mean of the data distribution. It measures the number of standard deviations a data point is from the mean of the data distribution. Z-score is a quick and efficient way to find anomalies, although it might not work for all kinds of anomalies or distributions of data. Also, it assumes that the data are regularly distributed, which may not always be the case. Figure 2 and 3 show the results when we applied the Z-Score algorithm on a time series dataset.

3.3 Prophet (Python Library)

Facebook researchers created this Python microframework library for financial markets. The Prophet library is able to forecast data which this feature can also use to detect anomalies by calculating the delta/change in actual data and forecasted data. However, this would require setting a fixed value which we can use to classify the calculated delta value as an anomaly or not. A simpler way is to use the $yhat_lower$ and $yhat_upper$ provided in the forecast data frame and classify any value above this as an anomaly. Figure 4 shows an example of an anomaly detected using the Prophet library.

3.4 GreyKite (Python Library)

Greykite is a Python library for time series forecasting and analysis, developed by LinkedIn. It provides a suite of powerful algorithms and techniques that can be used to analyze and predict time series data in a variety of industries, from finance to retail to healthcare.

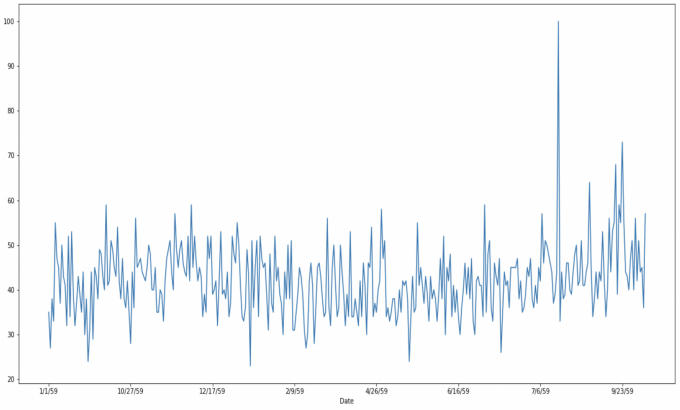

Fig. 2. Data before Z-Score algorithm was applied

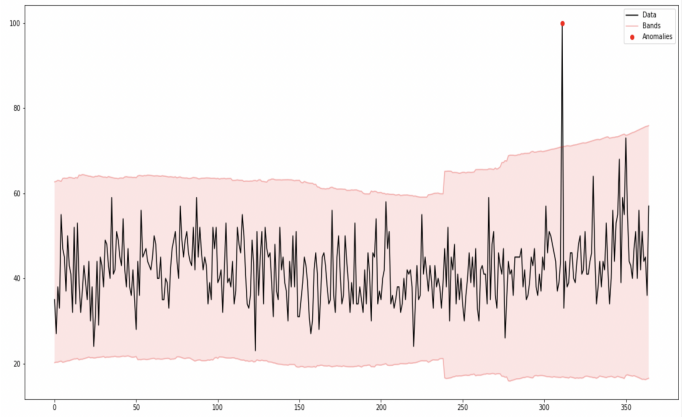

Fig. 3. Data after Z-Score algorithm was applied

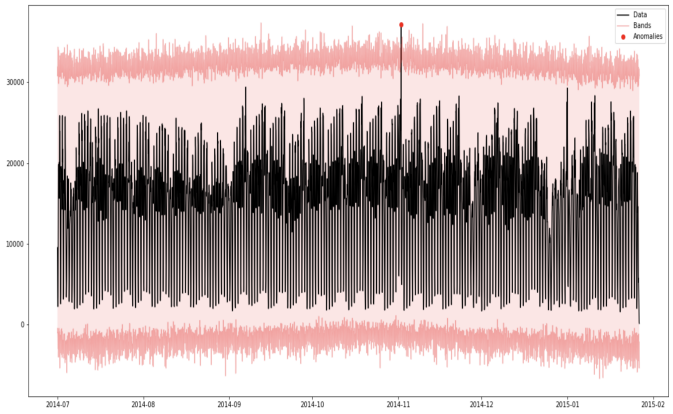

Fig. 4. Detecting anomaly using prophet

4 Method

Anomaly aggregation refers to the process of combining multiple signals or data streams to detect and analyze anomalies that may not be apparent when looking at individual data points. Some potential approaches to anomaly aggregation include Statistical aggregation [6], which involves using statistical methods such as mean, median, mode, standard deviation, or correlation to aggregate multiple data streams and identify anomalies. For example, if you are monitoring network traffic from multiple sources, you could aggregate the traffic volumes and look for spikes or dips that are statistically significant. Another example is Machine learning-based aggregation which involves training a machine learning model to identify anomalies in individual data streams and then using an ensemble method to combine the results from multiple models. This can be particularly useful in cases where the data streams are complex and exhibit nonlinear patterns. Also, Rule-based aggregation could create a set of rules or heuristics based on domain knowledge to identify anomalies across multiple data streams. For example, if you are monitoring the performance of a fleet of vehicles, you could create rules that flag anomalies when multiple vehicles show similar symptoms, such as sudden drops in fuel efficiency or engine temperature. Event-driven aggregation is also used for event-driven architecture to trigger an aggregation process when certain conditions are met. For example, if you are monitoring a pipeline for leaks, you could trigger an aggregation process when multiple sensors detect an increase in pressure or a drop in flow rate. Lastly, Hybrid aggregation [7] involves combining multiple approaches, such as statistical and rule-based or machine learning-based and event-driven, to create a more robust anomaly detection system. This can help you achieve higher accuracy and reduce false positives.

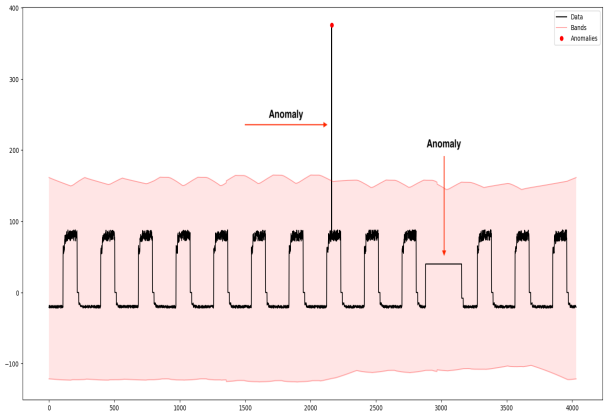

Fig. 5. Static/Dynamic Threshold with ZScore

4.1 Static/Dynamic Threshold Method

The concept of using thresholds is widely employed in multiple fields such as signal processing, anomaly detection, and quality control. The static threshold method is a simple approach where a fixed value is set based on pre-existing knowledge, statistical analysis, or domain expertise to differentiate between normal and abnormal behavior or categorize data. While the static threshold approach is easy to implement and comprehend, it may not be suitable for situations where the data distribution varies over time or when there are changes in the underlying characteristics of the data. To address these limitations, dynamic threshold methods are used which incorporate adaptability and responsiveness to changing conditions. Dynamic thresholds can be implemented using various techniques such as moving averages, standard deviations, or control charts that estimate the threshold based on the current system state or observed data. These statistical measures can be updated periodically or continuously to capture shifts, trends, and anomalies in the data. Dynamic thresholds are particularly useful when dealing with non-stationary data distributions that exhibit significant variations over time. By adjusting the threshold based on current data characteristics, they can provide more accurate detection of anomalies or changes in system behavior. However, dynamic threshold methods require careful parameter tuning and computational considerations. Appropriate update frequency must also be chosen while balancing sensitivity with false positives/negatives.

4.2 Forecasting Method

Forecasting involves predicting the future values of a time series data. To ensure accuracy, it's crucial to train the model using historical data that's been filtered to remove any anomalies - data points that deviate significantly from the expected pattern. By excluding these anomalies, we can obtain the true underlying pattern of the data and avoid negatively impacting forecasted accuracy.

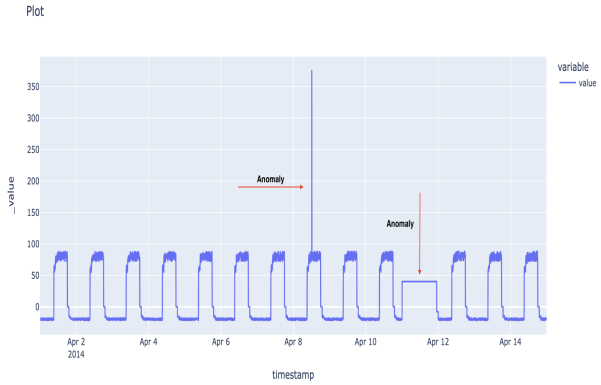

Fig. 6. Data before anomaly filtering

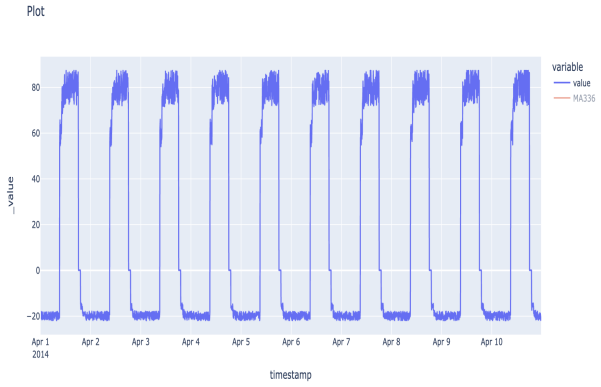

Fig. 7. Data after anomaly filtering

After filtering and preparing the data for forecasting, forecast data frames with upper and lower bands are common. These bands represent ranges within which actual data points are likely to fall. The upper band indicates a higher boundary while the lower band represents a lower boundary. To improve forecasting accuracy, a combination of percentage difference as shown in Listing 1.2 and severity as shown in Listing 1.1 can be used when dealing with unusual data points that exceed forecast limits (upper and lower bands). Percentage difference measures the relative deviation from the forecasted range while severity expresses the degree of deviation as a standardized score. When selecting between upper or lower bands for calculation purposes, choose the value closest to actual data points in order to reduce differences between actual and forecasted values - this helps align them better leading to improved accuracy. By combining percentage difference and severity measures as shown in Listing 1.3 with appropriate band selection, forecasting becomes more reliable in accounting for potential anomalies and limits while refining the overall process for enhanced reliability in predicting time series data.

```
def getSeverity(actual, isLower):
    if not isLower:
        meanData = forecast.df['forecast_upper'].mean()
        stdData = df['actuals_unfiltered'].std()
        zScore = (actual - meanData) / stdData
    else:
        meanData = forecast.df['forecast_lower'].mean()
        stdData = df['actuals_unfiltered'].std()
        zScore = (meanData - actual) / stdData

    severity = zScore * 100 / 3

    return bound_between(0, severity, 100)
```

Listing 1.1. Calculate the Severity Based on Z-Score

```
percentageError = (abs(forecastValue-actual)/abs(forecastValue+actual/2))
     * 100
    percentageError = bound_between(0,percentageError,100)
```

Listing 1.2. Calculate the percentage error between actuals and forecast data

```
anomalySeverity = ((percentageError + severity)/200) * 100
if anomalySeverity >= 52:
      print('Anomaly')
    else:
      print('Not Anomaly')
```

Listing 1.3. Combining both models to make decision

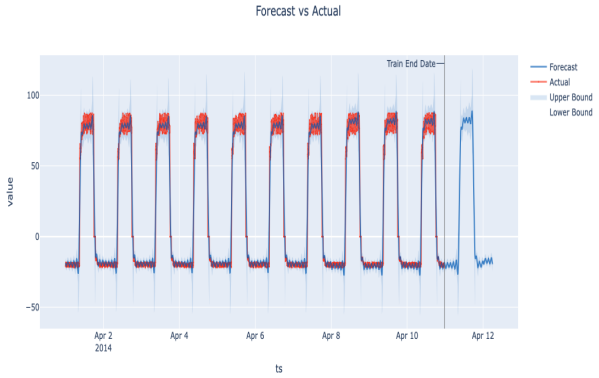

Fig. 8. Forecast data results

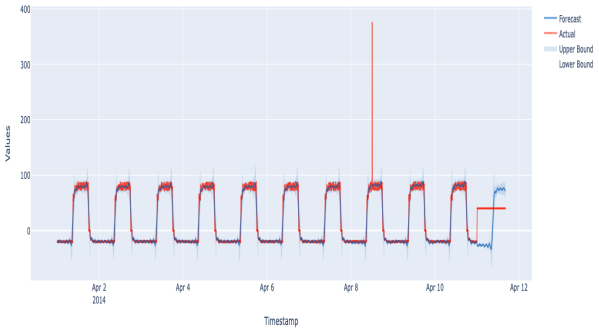

Fig. 9. Forecast vrs Actuals data results

5 Results

When examining the advantages of using the forecast method versus threshold-based anomaly detection, it becomes clear that the forecast method outperforms the threshold approach in terms of accuracy and effectiveness. Although suitable for capturing sudden spikes in data, the threshold method proves inadequate when it comes to detecting certain types of anomalies. To illustrate this point, consider an anomaly that occurred on April 12th (as shown in Figure 6). Despite its ability to detect sudden changes in data, the threshold method failed to identify this particular anomaly. This highlights a limitation of the threshold method - it struggles to detect anomalies that do not present themselves as immediate and noticeable outliers. During our research, we encountered an anomaly (shown in Figure 5) which resulted from an unusual change in underlying data patterns. Unfortunately, our use of a static/dynamic threshold approach proved insufficient for identifying this type of anomaly as shown in figure 5. This further emphasizes the downsides of using thresholds for detecting anomalies caused by unusual changes in data patterns. In contrast, by leveraging predictive models and analyzing historical data patterns, we found that the forecast method with statistical models(ZScore and Severity) demonstrated superior performance across various scenarios. It proved capable of identifying anomalies that would have gone unnoticed using threshold-based techniques. Furthermore, this approach offers a more comprehensive and robust solution for detecting different types of anomalies and adapting to changes in data patterns over time.

6 Conclusion

It is important to note that detection of time series anomalies in the end may not be easy considering its many types of anomalies. For efficient capture and detection of different anomaly types, specific modeling strategies are needed. Relying on only one anomaly detection model will be too risky because every model possesses certain unique properties that make it prone to various attacks. In an attempt to address these limitations, researchers and practitioners have identified the merits of integrating the various models. Such an approach would utilize different features of one method to cover another's weaknesses while benefiting from its strong sides to develop a stronger base system with more accurate and secure results. Nevertheless, one should emphasize that this topic is relatively fresh, so there are still many questions to be raised here. There is a need to be innovative with the available methods and ideas for improvement and efficiency of anomaly detection applied to time series data. Hence, future studies will aim at creating new and sophisticated models that offer solutions to different anomaly issues in a better way. The use of advanced machine learning algorithms, deep learning architectures, or other new technologies may be helpful. Moreover, research should focus on benchmarking different approaches on real-world databases to determine which methods are the most effective. Moreover, such forward movement in this field will bring about efficient and dependable

anomaly detection devices. Through constant innovation and research into different issues, we will also be able to recognize anomalies beforehand as well as draw beneficial insights that might guide us in making decisions, dealing with risks, or optimizing business processes.

References

1. H. Ren, B. Xu, Y. Wang, C. Yi, C. Huang, X. Kou, T. Xing, M. Yang, J. Tong, and Q. Zhang, "Time-series anomaly detection service at microsoft," in *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 3009–3017, 2019.
2. A. Hanbanchong and K. Piromsopa, "Sarima based network bandwidth anomaly detection," in *2012 Ninth International Conference on Computer Science and Software Engineering (JCSSE)*, pp. 104–108, 2012.
3. L. Menculini, A. Marini, M. Proietti, A. Garinei, A. Bozza, C. Moretti, and M. Marconi, "Comparing prophet and deep learning to arima in forecasting wholesale food prices," *Forecasting*, vol. 3, no. 3, pp. 644–662, 2021.
4. A. Patcha and J.-M. Park, "An overview of anomaly detection techniques: Existing solutions and latest technological trends," *Computer Networks*, vol. 51, no. 12, pp. 3448–3470, 2007.
5. V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM computing surveys (CSUR)*, vol. 41, no. 3, pp. 1–58, 2009.
6. S. Song, L. Ling, and C. Manikopoulo, "Flow-based statistical aggregation schemes for network anomaly detection," in *2006 IEEE International Conference on Networking, Sensing and Control*, pp. 786–791, IEEE, 2006.
7. D. Ohana, B. Wassermann, N. Dupuis, E. Kolodner, E. Raichstein, and M. Malka, "Hybrid anomaly detection and prioritization for network logs at cloud scale," in *Proceedings of the Seventeenth European Conference on Computer Systems*, EuroSys '22, (New York, NY, USA), p. 236–250, Association for Computing Machinery, 2022.