Platform Design for IoT Data Quality Improvement through Flexible Preprocessing Pipeline Building

Jaewon Moon
Information Media Research Center
Korea Electronics Technology Institute
Seoul, Republic of Korea
jwmoon@keti.re.kr

Seungwoo Kum
Information Media Research Center
Korea Electronics Technology Institute
Seoul, Republic of Korea
swkum@keti.re.kr

Seungtaek Oh
Information Media Research Center
Korea Electornics Technology Institute
Seoul, Republic of Korea
stoh@keti.re.kr

***Abstract*—With the proliferation of IoT devices, real-time data collection and monitoring services are being offered to various industries. However, time series data, due to its temporal nature, presents numerous unforeseen challenges over time, often leading to data contamination. Such contaminated data make it difficult to effectively utilize in practical applications. This paper investigates the issues with time series data collected from sensors and introduces preprocessing techniques and a corresponding platform to overcome these challenges. In addition to established preprocessing methods, novel approaches for processing and transforming multiple time series data sets are applied, followed by performance testing within the framework of the proposed service platform.**

Keywords—time series data, preprocessing, analytics platform, data pipeline

I. Introduction

There are two primary motivations for processing time series data collected from IoT devices. First and foremost, it aims to enhance the suitability of data for application by addressing both a multitude of faulty data points and the presence of gaps of varying lengths. In numerous applications, the presence of flawed data complicates processing and adversely affects the quality of analysis and learning outcomes. However, when collecting data over extended periods from diverse locations without meticulous control, the inclusion of lower-quality data becomes inevitable [1]. Therefore, a suitable preprocessing method becomes imperative to mitigate this.

Additionally, there is a need to transform time series data based on its intended purpose. Time series data finds application in diverse contexts such as analysis, learning, and monitoring. To cater to these multifaceted applications, it is essential to provide the data in formats that are apt for each specific use. Thus, due to these two primary reasons, data preprocessing before its application in various contexts is crucial. Of particular note is the fact that time series data, unlike other data, exhibits a strong dependence on the passage of time. This aspect must be carefully considered during preprocessing.

II. Backgrounds

A. Related Researches

In this section, we introduce some of the representative preprocessing methods for time series data. The objective of these methods is to enhance the quality of time series data. This involves detecting and rectifying inaccuracies or errors within the data, typically employing domain knowledge or statistical techniques. The primary aim is to identify outliers that deviate from the normal range, preventing potential distortions caused by such anomalies [2]. Additionally, addressing missing data resulting from the removal of identified erroneous values is crucial. When duplicated data is present, steps are taken to process or eliminate duplicates to maintain data accuracy. Furthermore, considering the inherent temporal nature of time series data, timestamps are managed, and data is aligned chronologically.

Moreover, ensuring the reliability and accuracy of time series data entails validating data format and range, as well as identifying and removing inappropriate data. This process contributes significantly to improve the quality of analysis and modeling results.

Furthermore, technologies are applied to enhance the performance of analysis and learning endeavors. Smoothing and filtering methods involve applying filters such as moving averages or exponential smoothing to mitigate noise and volatility in time series data. This effectively accentuates underlying patterns and trends. To address missing data, interpolation techniques encompassing linear, spline, and autoregressive methods are employed. Contemporary methodologies, including deep learning approaches, are also incorporated. Normalization of time series data enhances performance by achieving uniformity in variable scales. The application of feature engineering involves expanding or compressing raw time series data to extract meaningful features. In addition, transformation and dimensionality reduction techniques are pivotal in reconfiguring and transposing time series data into alternate dimensions, thereby enriching the depth and insights of analysis. Furthermore, various deep learning techniques are being researched for time series data preprocessing, and their performance is also improving, extending beyond traditional machine learning methods [3].

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.2021-0-00034, Clustering technologies of fragmented data for time-based data analysis).

B. *Challenges and issues*

Applying various high-performance time series data preprocessing techniques does not necessarily improve data quality. The effect of each technique may depend on the order in which one or several preprocessing methods are applied to a particular segment of time series data. This variability arises because the collected data has various statistical distributions, each contaminating the data with its own problems. However, despite these differences, preprocessing methods are universally applied to most time series data and can unintentionally reduce data quality.

For instance, the impact of each preprocessing technique can differ based on the order of their application to various segments and their influence on specific problematic regions within the data. Addressing these challenges requires a nuanced approach. It requires tailoring preprocessing strategies by considering the characteristics of each segment and the intended use of the data. By recognizing the idiosyncrasies of each dataset and sequentially applying preprocessing while accounting for existing challenges, optimal data quality can be achieved.

This paper aims to overcome the limitations of conventional preprocessing methods and adopt an adaptable approach based on context to effectively enhance data quality. Such an approach could ultimately result in improved analysis and more refined modeling outcomes.

III. Proposed Platform

A. *Overall Architecture*

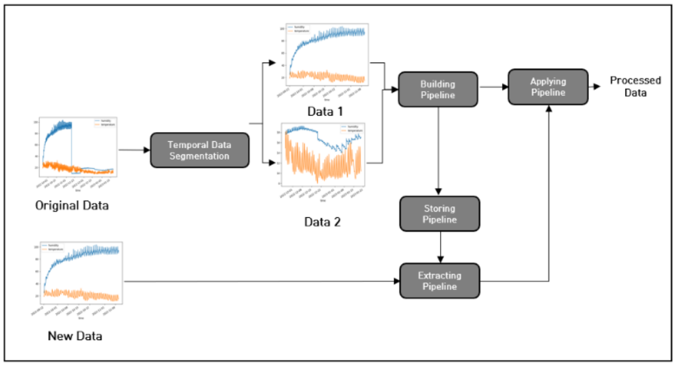


Fig. 1. Time series data segmentation preprocessing framework

This chapter describes the overall structure of the framework. The proposed approach is founded on the premise that applying the same preprocessing methods to the entire span of a single time series data can deteriorate data quality. Specifically, recognizing that time series data accumulated over extended periods encompasses anomalous data with diverse patterns attributed to distinct factors over time, it becomes imperative to address this heterogeneity appropriately. Thus, the time series data is partitioned into temporal intervals, and the most suitable preprocessing pipeline is constructed for each segment. The constructed pipelines undergo an evaluation to identify the most fitting pipeline for the test data. Furthermore, the application of the constructed pipelines enhances the quality of the segmented time series data. These stored pipelines can be uniformly applied to similar new data inputs, even as time series data can be segmented manually or automatically.

Figure 1 describes the utilization flow of such a data pipeline. The original data for constructing the pipeline is divided into Data 1 and Data 2 according to characteristics, and a data preprocessing pipeline is created and stored adaptively for each data. Afterward, when data with similar patterns, or similar problems such as outliers and noise is received, the optimal data preprocessing pipeline is retrieved and pre-processed by applying it.

B. *Flexible Selection and Linkage of Preprocessing Modules*

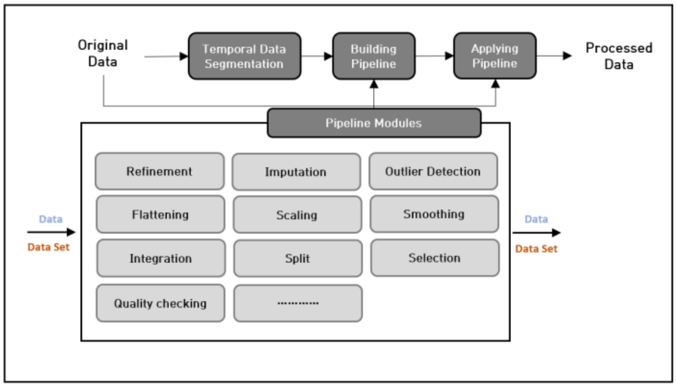


Fig. 2. Utilizing preprocessing pipelines based on submodule selection

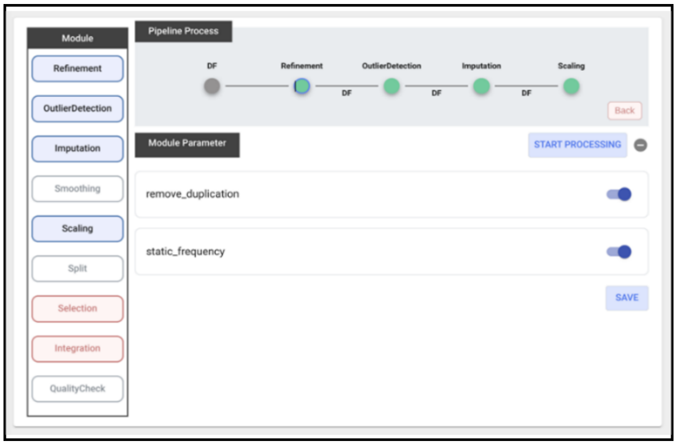
In this chapter, the method of preprocessing using a data pipeline consisting of multiple preprocessing sub-modules for segmented data is described. Each time series data encompasses multiple complex problem scenarios, often requiring the applying multiple preprocessing modules simultaneously to achieve the desired quality outcome. Appropriate preprocessing modules need to be applied to the data. Furthermore, it is assumed that the sequence and application order of each module will affect the quality of the data. Based on this premise, this paper defines a data processing pipeline as a sequential set of preprocessing modules connected in a specific order, with the autonomous selection of sub-modules forming its foundation. In this paper, the following 10 preprocessing sub-modules are considered.

Figure 2 illustrates the flow of data in the framework, depicting the construction of a pipeline based on sub modules and the preprocessing process applied to segmented data. In this paper, the following 10 actual preprocessing modules were considered and utilized:

- Refinement: Adjusts the technical intervals of time series data uniformly and removes unnecessary data to enhance consistency and accuracy.
- Imputation: Predicts and complements missing parts within time series data to maintain data integrity.
- Outlier Detection: Identifies and eliminates outlier data suspected of being problematic or noise, increasing data reliability.
- Flattening: Integrates multiple time series data into a single concise representation, facilitating streamlined data analysis.

- Scaling: Adjusts data according to the overall range of time series data, mitigating size discrepancies between data and maintaining analytical consistency.
- Smoothing: Reduces noise and fluctuations within data to reveal underlying trends and patterns, facilitating easier data interpretation.
- Integration: Combines various time series data into a singular dataset with unified time information, enhancing data utility.
- Split: Divides a single time series data into multiple distinct datasets based on specific characteristics, enabling diverse perspectives for analysis.
- Selection: Chooses time series data that meets given conditions from multiple datasets, precisely focusing analytical efforts.
- Quality Check: Selects time series data free from quality anomalies, ensuring reliable analytical outcomes.

C. *Case Study*



remove_duplication

static_frequency

Fig. 3. User interface for building data pipelines

In this chapter, the utilization scenarios of the proposed platform, taking into account these temporal characteristics, are introduced. The selected data can be subjected to the construction and application of preprocessing pipelines for one data. Figure 3 depicts the user interface for applying data processing using the platform introduced in the paper. In the presented example, the Refinement, Outlier Detection, Imputation, and Scaling are sequentially applied to data input. Fine-tuning of parameters for each preprocessing submodule is also feasible.

Figure 4 illustrates two configurations of pipeline settings. The first pipeline is structured with Refinement, Outlier Detection, Imputation, and Scaling consecutively applied to the data. The second pipeline utilizes Outlier Detection, Refinement, Imputation, Scaling, and Smoothing.

Figure 5 illustrates the outcomes of applying different pipelines. (a) represents the original graph without undergoing preprocessing, (b) is the result after passing through the Refinement-Outlier Detection pipeline, and (c) is the outcome after passing through the Outlier Detection-Refinement pipeline. Refinement involves down-sampling to adjust data to uniform intervals. The initial data contained periodic and significant outlier values. Consequently, it's evident that proper outlier detection processing should precede Refinement to eliminate these severe errors.

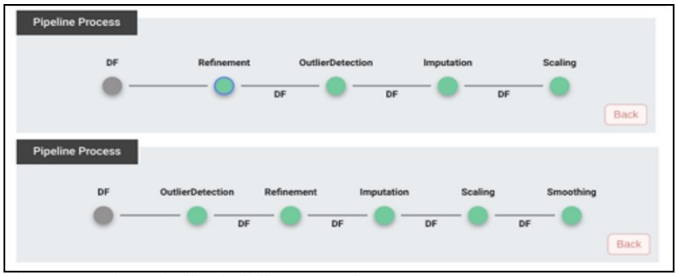


Fig. 4. Two data pipeline building examples

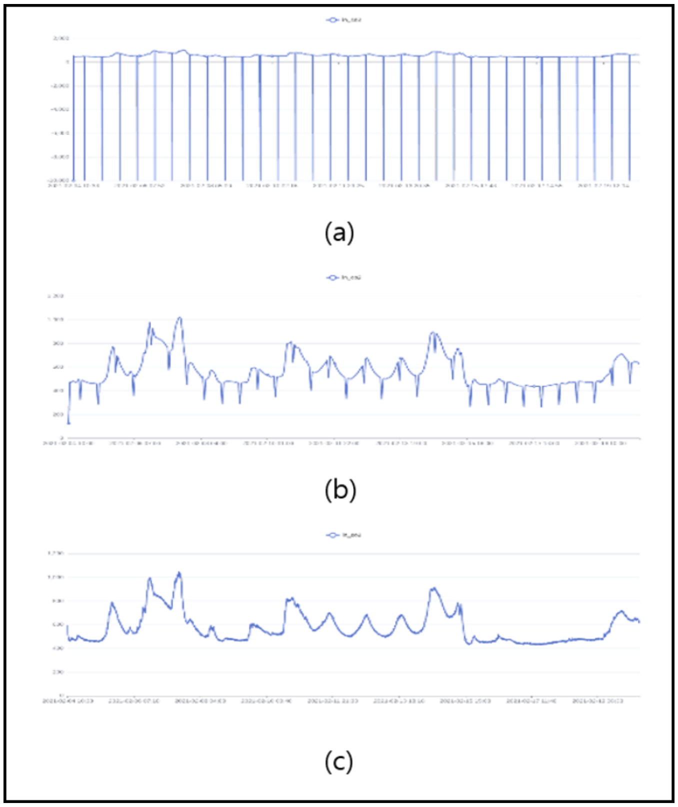


Fig. 5. Results of applying different preprocessing pipelines

Viewing the results graphs of (b) and (c) independently might lead to the misconception that both preprocessing steps were correctly executed. However, upon comparing the two graphs simultaneously, it becomes evident that the outliers present in the original graph have significantly influenced the post-preprocessing time series data due to the impact of Refinement.

IV. Conclusions

This paper examines the challenges arising from the long-term collection of time series data by IoT sensors and proposes preprocessing methods to address these issues. Despite the appropriate performance of individual existing preprocessing techniques, the quality of data may deteriorate depending on the order and combination of them. Based on this premise, this paper introduces a platform that utilizes such modules to construct and apply preprocessing pipelines. The proposed platform is able to offer the capability to seamlessly apply segmented processing of time series data and the integration of diverse preprocessing modules. Subsequently, this platform is employed to preprocess actual IoT big data through various pipelines, with plans to conduct further research to assess how these differences impact learning outcomes.

REFERENCES

[1] G. -O. Meritxell, B. Sierra and S. Ferreiro, "On the Evaluation, Management and Improvement of Data Quality in Streaming Time Series," in IEEE Access, vol. 10, pp. 81458-81475, 2022, doi: 10.1109/ACCESS.2022.3195338.

[2] A. A. Cook, G. Mısırlı and Z. Fan, "Anomaly Detection for IoT Time-Series Data: A Survey," in IEEE Internet of Things Journal, vol. 7, no. 7, pp. 6481-6494, July 2020.

[3] A. Y. Yıldız, E. Koç and A. Koç, "Multivariate Time Series Imputation With Transformers," in IEEE Signal Processing Letters, vol. 29, pp. 2517-2521, 2022