POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Executive Summary of the Thesis

Exploring Data Preparation Strategies for Data Stream Analysis

Laurea Magistrale in Computer Science and Engineering - Ingegneria Informatica

Author: Giovanni Siracusa

Advisor: Prof. Cinzia Cappiello

Co-advisor: Camilla Sancricca

Academic year: 2022-2023

1. Introduction

Data stream analysis has been a prominent trend in recent years, applied in fields such as IoT, Finance, and sensor scenarios like Industry 4.0. For an accurate data analysis, an indispensable prerequisite is a high-quality dataset, to ensure that the errors present in the data do not propagate throughout the analysis results. Achieving good data quality necessitates the implementation of effective data preparation actions.

Data preparation includes all the methods and techniques that facilitate the collection, organization, and cleaning of data. Given the natural velocity of data streams and their unbounded volume, it is necessary to employ methods that manage them promptly and effectively.

This thesis aims to enhance an existing data preparation framework originally crafted for tabular datasets, introducing support for data streams. The undertaken tasks encompass data profiling, data quality assessment, and data preparation, with the objective of implementing them in an incremental way, employing methods tailored to the dynamic nature of data streams.

This summary is organized as follows. Section 2 briefly introduces the topic of this thesis, examining the present literature and introducing the gaps that this work aims to fill. Section 3 presents the architecture of the framework and how it is enhanced. Section 4 presents how the methodology is practically implemented and lists all the employed methods. Section 5 illustrates the findings obtained after the experimental stage.

2. Background

Data streams are defined as a countably infinite sequence of elements. Different models of data streams exist. They can be represented as an unbounded vector of elements, in which is possible to add, modify, and delete elements. Another possibility is to represent structured streams with the cash register model, in which only additions to the underlying vector are permitted. These streams are processed considering both volume and velocity and should be processed with low latency [2]. Data quality (DQ) becomes a crucial focal point when analyzing data in all of its forms, and it involves every single step of the pipeline. Low-quality data affects data management, machine learning models, data visualization, etc. DQ is defined as "fitness for use" and can be assessed across various dimensions. The most common DQ dimensions in literature are: [1].

- **Accuracy** - Closeness between a data value v and its real-world representation v'
- **Consistency** - Captures the violation of semantic rules defined over data stream
- **Completeness** - Extent to which data are of sufficient breadth, depth and scope
- **Timeliness** - How current are the data for the task at hand

Data profiling is the process of examining the data available from an existing information source, which, in our case, is a data stream. The objective of data profiling is to gather statistics and information about the data and generate related metadata. Data profiling can be categorized into single source and multiple source analysis. Illustrative tasks include analyzing data distribution and discovering functional dependencies.

Data preparation aims to enhance data quality before any analysis. These activities encompass not only those improving the aforementioned data quality dimensions but also other actions such as data structuring or data fusion. The primary tasks considered in this thesis are outlier detection and missing values imputation, aimed at enhancing accuracy and completeness, respectively.

The gaps identified in the existing literature are related to continuous profiling, addressed by Naumann in 2013, but not significantly addressed in the subsequent literature [3]. Another vacancy regards data quality; in fact, there exists a description for many DQ dimensions specific to data streams, and frameworks for data quality assessment have been designed, but the development of a ready-to-use tool that allows the representation of data quality in a real-time dashboard has not been faced yet. In conclusion, several studies are related to evaluating the impact of data preparation on tabular data analysis, but no significant paper was found about data stream analysis.

3. Methodology

This thesis aims to enhance an existing framework designed to recommend data preparation actions to end users, originally developed for tabular datasets. The improvement involves adapting the framework's functionality to address the complexities of data streams, thereby expanding its applicability to a more diverse range of data sources.

The main tasks carried out involve the implementation of techniques for continuous data profiling, the data quality assessment performed in real-time, and finally, an experimental stage to enhance the knowledge base, which is used for the suggestion of data preparation actions, testing which methods are more adapt for the challenges posed by data streams.

The objective of the existing framework is to explore and profile a dataset, assess its quality and, depending on the results of the data profiling and data quality assessment, provide the best possible data preparation actions to perform. The architecture of this framework includes an initial input phase in which the user can select the data source d to analyze and set user preferences related to d or the type of analysis to be conducted.

d has been explored and profiled, providing insights into its characteristics. The statistical information obtained from this process is used to assess the initial data quality, which is then conveyed to the user to provide a preliminary understanding of the dataset's quality level. Subsequently, the user can move on to the data preparation stage, in which a knowledge base is investigated to suggest the most suitable data preparation actions to perform before the analysis. The recommendations are based on the quality level of the dataset and on the type of analysis which is requested by the user. Finally, the user can execute the suggested actions before the analysis model is fed with the dataset. This process is intended to improve the quality of the analysis, eliminating errors and faulty data points to prevent them from influencing the analysis outcomes.

The additions made and discussed in the thesis concern the control of data streams, facilitating continuous profiling, real-time quality evaluation, and incorporation of the most appropriate data preparation measures for data streams into the knowledge base. When the user decides to analyze a data stream, the pipeline starts with data ingestion. Contrarily from the previous pipeline, d is ingested row by row, simulating a data stream. The initial change concerns the data profiling task, where some of the methods applied in batch processing are adapted to perform incremental and continuous profiling, describing data distributions and functional dependencies. As the streams flow row by row through the pipeline, the data profiling models are updated for each data point received.

Data quality assessment uses a windowed approach, where each selected DQ dimension is assessed using a metric that is calculated at the end of each window using information from the data profiling phase. The stream processing employed for stream profiling enables real-time monitoring of the stream's quality.

The final stage of the data preparation and analysis process employs a windowed approach. A fixed-sized window is established, and the initial window is populated with the initial stream elements. Subsequently, when the window reaches its maximum capacity, the tasks are executed, and the window slides to create space for new samples. This allows for considering only the most recent data points.

The other addition realized involves enriching the knowledge base by introducing data preparation techniques appropriate for the dynamic nature of data streams previously tested, and their impact on machine learning predictions is evaluated. This step is executed using the same windowed approach described above. A dataset d is intentionally injected with various errors at different levels of quality, resulting in the creation of distinct datasets $d_1, d_2, \ldots$. On each of these datasets d_i, once the first window composed of x samples is complete, a data preparation action dpa_j is executed. Subsequently, a machine learning model is trained and tested on a test set. The window then slides, accommodating an additional y samples. Once the window reaches again x samples, data preparation and analysis are re-executed. This process is repeated until the stream concludes. At the end of the stream, all the predictions returned by the machine learning model are evaluated by generating a metric m_{ij}. All the m_{ij} values are compared to comprehend the behaviour of dpa and how its influence on analysis changes according to data quality. For the sake of completeness, an additional test is executed without executing any data preparation action. This helps to evaluate the impact of data preparation on the analysis.

All the implemented techniques are outlined in Section 4.

Six different experiments have been conducted to evaluate techniques for outlier detection and missing values imputation thoroughly. Five of these experiments follow the methodology described above.

The first experiment involves the injection of missing values into the original dataset. Various methods for data imputation are executed, and their impact on the analysis is subsequently assessed.

For three additional experiments, outliers are injected into the original dataset. The first test, which differs from the pipeline described above, does not include any machine learning model and prediction. The test is designed to evaluate the effectiveness of outlier detection techniques by comparing the detected anomalies with the injected ones. The comparison is done at the end of the process, while the outliers are detected with the same windowed approach described above.

In the second trial, outliers are identified and substituted with a standard value before the machine learning phase. In the last of these trials, detected outliers are substituted using data imputation techniques and subsequently analyzed with a machine learning model.

In the last two experiments, the original dataset is modified by inserting both missing values and outliers. The goal is to comprehend if a change in the sequence of operations yields different outcomes. In the first assessment, outliers are corrected, and subsequently, missing values are imputed, while in the second, the opposite occurs.

All these experiments have been replicated on a tabular dataset, allowing for a comparison of the behaviour of the tested methods in both a stream and a batch process.

4. Experimental Setup

In this section, all the technologies and specifications used for practical implementation are outlined. The source code has been written entirely in Python, utilizing widely used data science libraries such as NumPy, Scikit-Learn, and Pandas. Additional packages include River for streaming machine learning and PyOD for implementing outlier detection techniques. The simulation of the streams occurs with an Apache Kafka server, where a Kafka producer sends messages, and a Kafka consumer reads them.

Three different datasets are employed, with two being data streams, called AirQuality and NEWeather. The associated analysis tasks are regression for the first and classification for the second. The last is a tabular dataset that allows testing for both regression and classification, called Electrical.

The machine learning model employed is Random Forest, a tree-based model that constructs a predictor by creating an ensemble of decision trees. It supports both regression and classification tasks, and the implementation used is provided by the Scikit-Learn package.

For each window, the dataset is split into training and test sets, utilizing the first 67% for training and the remaining 33% for testing. This approach is crucial in a data stream as it avoids random shuffling of samples, allowing the model to be trained on preceding values and used for subsequent ones.

The metrics employed to evaluate the performance of the machine learning models include F1-Score for classification and R2-Score for regression. Additionally, F1-Score has been used to assess the effectiveness of outlier detection techniques by comparing the detected outliers with the injected ones.

4.1. Employed Techniques

The techniques employed for data profiling are adapted from those used in a batch scenario. Data distributions are computed in an incremental way. Mean and standard deviation are computed with an easy formula, while quantiles require the use of the KLL data structure. Functional dependencies are detected using the Apriori algorithm, which is regularly employed in associative rule mining. In our case, this algorithm has been implemented using a windowed approach.

The tested methods for outlier detection are:

- **Z-Score** - Statistical method to detect outliers according to the mean and the standard deviation of the dataset.
- **Local Outlier Factor** - Density-based method that retrieves outliers using the concept of local density, computed on the distance of the k-nearest neighbors.
- **Isolation Forest** - Tree-based method that builds an ensemble of Isolation Trees, designed to isolate every instance effectively.
- **Half Space Trees** - Incremental tree-based method. It builds a model and updates it one sample at a time. It requires a constant amount of memory.

Local Outlier Factor (LOF) and Isolation Forest (IForest) are used in a windowed fashion, while the other two techniques are computed in an incremental way.

The applied techniques for outlier correction and missing values imputation are:

- **Dropping** - Elimination of all the rows with missing values.
- **Last Observation Carried Forward** - Missing values are filled, propagating the previous values.
- **Mean** - Impute the missing values using the mean of the actual window.
- **Linear Interpolation** - Technique that computes missing values according to adjacent points, assuming a linear relationship between data points

5. Results

This section describes the results obtained from the experiments outlined in Section 3 with the presence of selected graphs. The presented graphs illustrate the performance achieved by machine learning models in relation to the dataset quality. The Y-axis shows the F1-Score or the R2-Score, while the X-axis represents the percentage of the dataset quality. The graphs consistently feature a solid line representing the performance achieved with the clean dataset and a dashed line representing the performance with the injected datasets without applying any interventions.

Concerning data imputation techniques, LOCF and interpolation yield comparable results, while the mean has the worst impact on performance metrics, even worse than not performing any imputation. This occurs because it causes a discontinuity in the stream.

In the second experiment, Local Outlier Factor, applied with a windowing approach, proved to be the most effective technique for detecting outliers, outperforming both Half Space Trees and IForest. Z-Score is the least effective method due to its sensitivity to high standard deviation, and in the NEWeather dataset, it fails to recognize any outliers.

The third experiment shows that substituting outliers with a standard value leads to generalized underfitting. In an unexpected turn of events, the results of the fourth experiment have shown that correcting outliers has led to a decline in performance, as opposed to the initial assumption that it would lead to improvement. Further investigations show that this is due to the model chosen. In fact, Random Forest proves to be very robust to outliers, while an additional experiment performed with K-Nearest Neighbors leads to a foreseen trend in which the correction of anomalies improves the performances. The sole outlier correction method that enhances performance is dropping anomalies. In certain cases, it even surpasses the performance achieved using the clean dataset—a sensible outcome as the prediction retains more of its informative power. Figure 1 illustrates these results in the NEWeather dataset, when using LOF as outlier detection method.

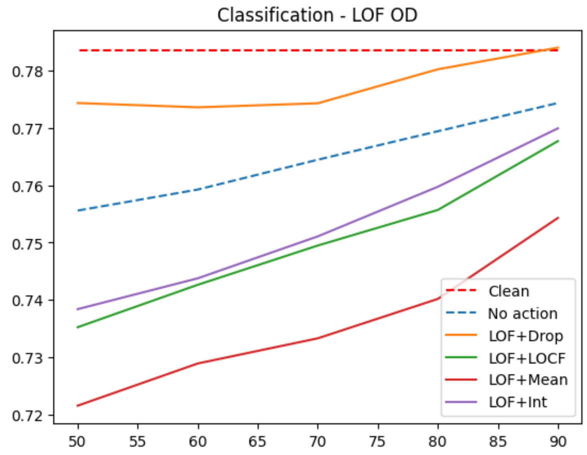

Figure 1: Experiment No.4 - NEWeather

The last two experiments aim to assess if the sequence of performed tasks influences performance. Executing outlier correction first and then data imputation appears to have no critical impact. Conversely, performing data imputation before outlier correction might introduce a bias toward anomalies, propagating them into clean data points with missing values. The obtained results do not indicate a significant difference in trends, showing only a slight advantage toward the results of the fifth experiment. Figure 2 illustrates this trend by comparing some of the most significant results in the AirQuality dataset.

The final assessment compares the impact of data preparation techniques between data stream and tabular datasets. While the tested techniques remain the same, batch processing is employed instead of a windowed approach.

The results are in line with the predictions for Z-score and mean imputation, which have demonstrated higher efficacy in tabular datasets compared to data streams. This was expected based on previous findings that these techniques are incompatible with the features of data streams. In most other instances, the windowed methods outperform batch processing. In these cases, thanks to the decay of data being taken into account and only the most recent samples being considered, the machine learning model exhibits greater accuracy. Figure 3 illustrates the performance of Z-Score in the outlier recognition task.

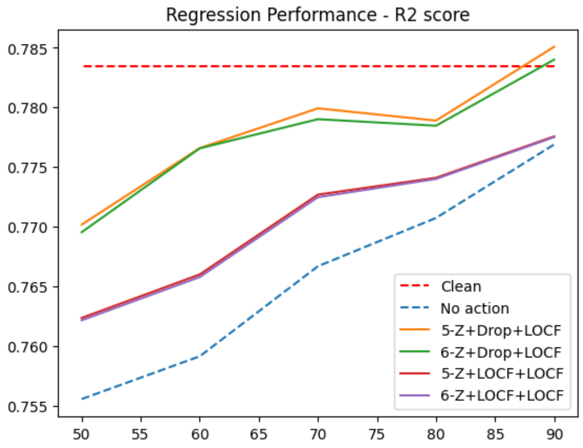

Figure 2: Comparison experiments No.5 and No.6

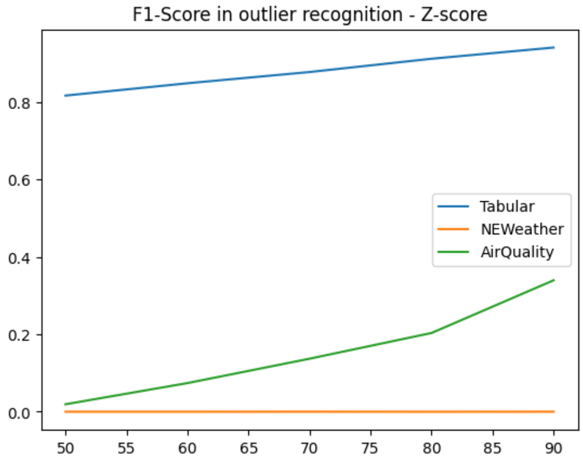

Figure 3: Experiment No.2 - Z-Score

6. Conclusions

The objective of this thesis was to enhance a data preparation framework, primarily by adapting the techniques used in the framework for data streams. The initial objective was to design and implement a continuous stream profiling procedure that facilitated real-time data quality assessment. This allowed for a comprehensive evaluation of the trends in data characteristics.

The second upgrade to the framework focuses on expanding the knowledge base employed to suggest data preparation operations by incorporating details pertinent to data streams. This augmented the knowledge base content by assimilating the results acquired in an experimental stage in which two distinct data streams were subjected to assessments of outlier detection and missing values imputation methods.

The conducted trials yielded valuable insights into the impact of data preparation on data stream analysis. These results were influenced by the choice of the machine learning approach. Further studies can delve into this field, testing other models and exploring additional data preparation actions. Potential future developments could involve the examination of deep learning techniques to identify outliers or impute missing values or the application of targeted methods for univariate time series, such as ARIMA models. Another potential addition involves incorporating adaptive window size adjustment methods.

References

[1] Carlo Batini and Monica Scannapieco. *Data and Information Quality*. Springer International Publishing, 2016.

[2] Alessandro Margara and Tilmann Rabl. *Definition of Data Streams*, page 1–4. Springer International Publishing, 2018.

[3] Felix Naumann. Data profiling revisited. *ACM SIGMOD Record*, 42:40–49, 02 2014.