Metro-Viz: Black-Box Analysis of Time Series Anomaly Detectors

Philipp Eichmann
Brown University
philipp_eichmann@brown.edu

Franco Solleza
Brown University
franco_solleza@brown.edu

Junjay Tan
Brown University
junjay_tan@brown.edu

Nesime Tatbul
Intel Labs and MIT
tatbul@csail.mit.edu

Stan Zdonik
Brown University
sbz@cs.brown.edu

ABSTRACT

Millions of time-based data streams (a.k.a., time series) are being recorded every day in a wide-range of industrial and scientific domains, from healthcare and finance to autonomous driving. Detecting anomalous behavior in such streams has become a common analysis task for which data scientists employ complex machine learning models. Analyzing the behavior and performance of these models is a challenge on its own. While traditional accuracy metrics (e.g., precision/recall) are often used in practice to measure and compare the performance of different anomaly detectors, such statistics alone are insufficient to characterize and compare the algorithms in a systematic, human-interpretable way. In this extended abstract, we present Metro-Viz, a visual analysis tool to help data scientists and domain experts reason about commonalities and differences among anomaly detectors, and to identify their strengths and weaknesses.

CHI, May 2019, Glasgow, UK
 The definitive Version of Record was published in *CHI Conference on Human Factors in Computing Systems Extended Abstracts (CHI'19 Extended Abstracts), May 4–9, 2019, Glasgow, Scotland Uk*, https://doi.org/10.1145/3290607.3312912.

KEYWORDS
Time Series; Anomaly Detection; Visual Analysis

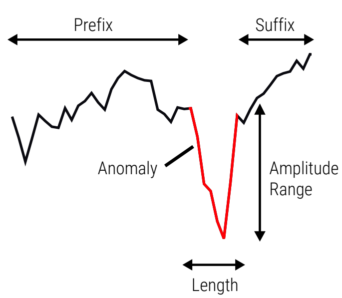

Figure 1: Example of implicit time series anomaly properties.

MOTIVATION

When tasked to perform time series anomaly detection (i.e., finding unusual data points or patterns) on a new data source [3], data scientists see themselves confronted with questions like: which anomaly detection systems work best for this specific task, where do these algorithms fall short, and why. Similar questions arise in virtually any classification domain. In machine learning, the performance of classifiers is typically evaluated using summary statistics such as precision, recall, or F-score. In a recent paper, we introduced a scoring model that is based on these statistics, specifically tailored to time series classification systems [11]. Yet, while these metrics are useful for benchmarking and to get a sense of overall model performance, more involved questions that would lead to actionable insights for model improvements, such as the ones above, cannot be answered by any of these summary statistics directly [4].

To address this issue for tabular and image data, researchers have built tools like ModelTracker [2], Squares[9], and Google's What-If Tool [7], which allow users to inspect and understand correct and false predictions of a single machine learning model. However, these tools lack support for time series classification tasks. In this extended abstract, we introduce Metro-Viz, a system that aims to narrow this gap by providing a set of tools that are specifically designed for users who want to analyze the results of time series anomaly detectors in a systematic way. Through our work on building practical tools and systems for enabling next-generation time series anomaly detection solutions [10], as well as through closely collaborating with machine learning experts, we identified a set of functional requirements. We summarize these requirements in terms of four core challenges that we aim to address with Metro-Viz:

- *Challenge 1: Ambiguity.* What constitutes an anomaly depends on the application domain and context. For example, whether or not a sudden change in value or an irregular pattern should be considered as an anomaly might depend on factors such as the dataset and the task, the time of occurrence, periodicity, what happened before and after the anomaly occurred, as well as global trends (see Figure 1 for an example). Furthermore, anomalous patterns may evolve over time; what may have been anomalous a year ago might not be now. Exposing meta-data of anomalies helps data scientists to characterize how anomaly detectors behave under such conditions.
- *Challenge 2: Comparing Multiple Models.* We learned that many anomaly detection system developers struggle to compare multiple different models in a rigorous and systematic way, which goes beyond using simple summary statistics. This is mostly due to the overhead of having to create plots of success and failure cases, creating a mapping between the same anomalies detected by different detectors, or even more pragmatically, "seeing the data". Streamlining this process is crucial for a more efficient analysis.

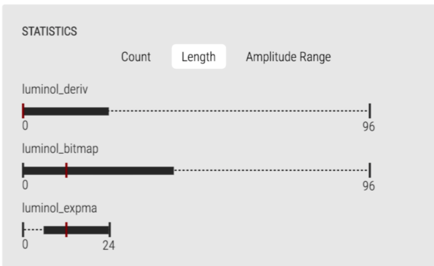

Figure 2: Users can inspect statistics, e.g., the length of all anomalies, in a box plot, showing the minimum, maximum, median, and interquartile range (IQR) per detector.

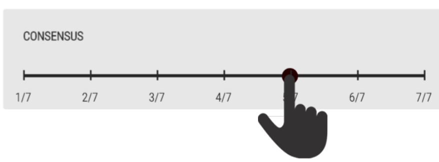

Figure 3: The consensus slider can be used to filter anomalies displayed in the time series viewer to ones that are detected by at least a certain number of detectors (5 out of 7 in this example).

- *Challenge 3: Time Granularity.* Anomalies might only become visible at certain time granularities or aggregation levels. Depending on how raw data is aggregated, e.g., by computing the average or maximum value for fixed time intervals like days or seconds, anomalous data points might get smoothed-out or accentuated. Understanding the impact of sampling and aggregation on time series anomaly detector performance is important when tweaking a detector for a specific use case.
- *Challenge 4: Missing and Unknown Labels.* Anomalies are, by definition, rare or unique events. This makes it difficult to determine ground-truth for certain problems. Captured training datasets therefore often only exhibit a small and incomplete set of possible anomalous patterns. As a result, classifiers might be misguided during the training process, requiring careful human testing and probing before deployment. Furthermore, it is infeasible to train anomaly detection algorithms for every possible scenario. Thus, functionality to modify input data and test hypothetical scenarios can help users to evaluate a model, even when training/test data is scarce.

METRO-VIZ

Overview

Metro-Viz is a web application that runs in any modern web-browser, with an accompanying backend written in Python. It features a set of default anomaly detectors and allows users to plug in custom detectors by implementing an API that Metro-Viz uses to communicate with the detector. This enables Metro-Viz to be used with a diverse set of off-the-shelf or newly developed anomaly detectors, irrespective of the programming language in which they are written. Data can be loaded into Metro-Viz either by specifying a CSV file or by implementing a database adapter.

Features

In the following we describe the core functionality we built into Metro-Viz to address the challenges outlined in the previous section.

Visual Anomaly Analysis. To address Challenge 1, we developed tools to browse through time series data and analyze the detected anomalies. In Metro-Viz, a time series is displayed as a line graph in the time series viewer (Figure 4B), when selected from the main menu (Figure 4A). The time series viewer shows a window of a customizable number of data points. Users can pan to slide the window across the time axis (Figure 4C), or alternatively zoom into a subsequence of interest by using a drag-and-drop gesture. To trigger anomaly detection on the current time series, one or more anomaly detectors can be selected from the menu on the left. Found anomalies are highlighted in the line graph, color-coded by set membership (see Figure 4F). Hovering over the line graph displays different anomaly properties in a separate view below (Figure 4D). Furthermore, users can identify anomalies

Figure 4: A screenshot of Metro-Viz's user interface: A) Main menu, B) Time series viewer, C) Time axis showing the full time range, D) Anomaly properties of a currently selected anomaly, E) The consensus slider that can be used to limit the shown anomalies to those detected by a minimum number of detectors, F) The set operation widget that can be used to define and query anomaly sets, G) Summary statistics for all anomaly detectors selected in the menu.

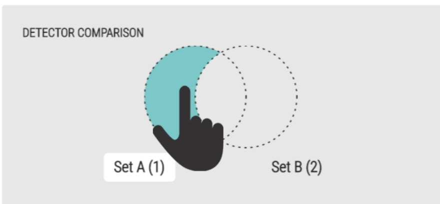

Figure 5: Anomaly detectors can be assigned to either set A or set B. This allows users to selectively display anomalies, e.g., only anomalies detected by detectors in set A but not in set B, by clicking on the corresponding portion in the Venn diagram.

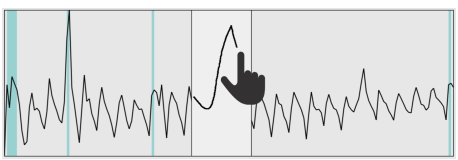

Figure 6: Metro-Viz allows users to erase arbitrary portions of a time series and replace them with a sketch in order to probe anomaly detectors with a pattern not present in the data.

that are more likely to be true anomalies (those that are detected by more than just a single detector), by adjusting the *consensus* value through a slider (see Figure 3 and Figure 4E).

Comparing Multiple Detectors. Metro-Viz provides functionality to answer questions like: which anomalies were detected by both algorithm A and algorithm B, by algorithm A but not by algorithm B, and vice-versa (Challenge 2). Users can assign detectors to either one of two sets, and click on the desired portions in a Venn Diagram (Figure 4F and Figure 5) to select the anomalies displayed in the time series viewer. By default, two anomalous ranges are defined to be equal if they share at least one data point. The amount of overlap required for the system to consider two subsequences as equal can be configured in the settings.

Aggregating across Time. To address Challenge 3, time series can be aggregated across different time scales/granularities (Figure 4A), e.g., day, hour, minute, using four different aggregation functions (`sum`, `avg`, `min`, `max`). Time series data that is not explicitly time-stamped, but has an implicit order can be aggregated using a set of configurable sampling methods.

Interactive Hypothesis/What-If Testing. We address Challenge 4 by enabling data scientists and domain experts to simulate and understand the behavior of anomaly detectors thorough patterns not present in the data. For example, users can test scenarios like: Would detector A still classify this peak as an anomaly, if its amplitude were lower or if its prefix were different? Metro-Viz provides features to modify a time series displayed in the time series viewer. Data points can be selected through a gestural interface, as proposed by [5]. Once selected, these points can be moved along the vertical axis, changing the amplitude of the selected subsequence. Furthermore, similar to functionality introduced in TimeSketch [6] or Qetch [8], users can select a subsequence in the time series viewer and replace it with a hand-drawn sketch (see Figure 6). A modification to the original time series will automatically re-run all selected detectors. Immediate responses from the backend allow the user to engage in a back-and-forth dialog with the system when probing detectors.

FUTURE WORK

Detector-Guided Labeling. With the increasing variety of streaming data, users become data curators in addition to analysts [1]. While Metro-Viz has proven useful within a small group of test subjects to analyze anomaly detection algorithms, our pilot users expressed that in many cases they would have liked to mark anomalies as false positives or false negatives, i.e., adding or refining ground-truth labels while analyzing the results. Based on this feedback, we are currently extending Metro-Viz with functionality to perform what we refer to as *detector-guided labeling*. This means that, instead of having users manually scan a time series to annotate anomalous ranges, Metro-Viz will guide users to anomalous time series ranges discovered by a set of pre-trained detectors, and ask users to classify them as false, true, or partially true by correcting the anomalous range. Given that multiple pre-trained detectors are used, every anomaly can be assigned a probability (confidence) of it being a true anomaly, analogous to the consensus feature outline above. Our hope is that users will experience the task of labeling time series as less daunting and will be more efficient, as anomaly verification can be done iteratively, starting from the ones with high confidence to the ones with low confidence.

Scalable Data Backend. Motivated by the scale of the data that we received from our collaborators in healthcare and computer networking sectors, the database experts on our team are currently architecting a data management system that will allow users to scalably perform Metro-Viz style interactive analysis on large datasets.

CONCLUSION

In this extended abstract, we presented an early design of Metro-Viz, a tool that allows data scientists and domain experts to analyze the behavior and results of time series anomaly detectors. Further information about this work is available on our project website [1].

[1]http://metronome.cs.brown.edu/

ACKNOWLEDGMENTS

This research has been funded in part by Intel and by NSF grant IIS-1514491.

REFERENCES

[1] Daniel J. Abadi et al. 2014. The Beckman Report on Database Research. *ACM SIGMOD Record* 43, 3 (2014), 61–70.
[2] Saleema Amershi et al. 2015. ModelTracker: Redesigning Performance Analysis Tools for Machine Learning. In *ACM CHI Conference*. 337–346.
[3] Varun Chandola et al. 2009. Anomaly Detection: A Survey. *ACM Computing Surveys* 41, 3 (2009), 15:1–15:58.
[4] Finale Doshi-Velez and Been Kim. 2017. Towards a Rigorous Science of Interpretable Machine Learning. *ArXiv e-prints* abs/1702.08608 (2017).
[5] Philipp Eichmann et al. 2017. Discrete Time Specifications in Temporal Queries. In *ACM CHI Conference*. 2536–2542.
[6] Philipp Eichmann and Emanuel Zgraggen. 2015. Evaluating Subjective Accuracy in Time Series Pattern-Matching using Human-Annotated Rankings. In *IUI Conference*. 28–37.
[7] Google. 2018. What-If Tool. http://pair-code.github.io/what-if-tool/.
[8] Miro Mannino and Azza Abouzied. 2018. Expressive Time Series Querying with Hand-Drawn Scale-Free Sketches. In *ACM CHI Conference*. 388:1–388:12.
[9] Donghao Ren et al. 2017. Squares: Supporting Interactive Performance Analysis for Multiclass Classifiers. *IEEE Transactions on Visualization and Computer Graphics* 23, 1 (2017), 61–70.
[10] Nesime Tatbul. 2019. Tools for Advanced Time Series Analytics: Enabling the Future. In *CIDR Conference*.
[11] Nesime Tatbul et al. 2018. Precision and Recall for Time Series. In *NeurIPS Conference*. 1924–1934.