Event Monitoring in Modern Public Health Data Streams

Ananya Joshi

CMU-CS-25-103

March 2025

Computer Science Department
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee:
Roni Rosenfeld and Bryan Wilder, Co-Chairs
Carnegie Mellon University Dept. of Machine Learning
Rayid Ghani - Carnegie Mellon University Dept. of Machine Learning
Matt Biggerstaff - Centers for Disease Control and Prevention, Flu Division

Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

This work was supported by the Centers for Disease Control and Prevention of the U.S. Department of Health and Human Services (HHS) as part of a cooperative agreement funded solely by CDC/HHS under federal award identification number U01IP001121, "Delphi Influenza Forecasting Center of Excellence" and by the National Science Foundation Graduate Research Fellowship under Grant No. DGE1745016 and DGE2140739.

Any opinion, findings, and con- clusions or recommendations expressed in this material are those of the author and do not neces- sarily reflect the views of the National Science Foundation. The contents are those of the authors and do not necessarily represent the official views of, nor an endorsement by, CDC/HHS or the U.S. Government.

Keywords: Anomaly Detection, Computational Epidemiology, Human Computing Interaction, Online Learning, Time Series Analysis, Visualization, Data Monitoring

Dedication

I had a great doctoral experience, and I am grateful to my advisors, Roni Rosenfeld and Bryan Wilder, as well as the Delphi and LASI groups for providing a supportive research community. Nolan Gormley, Richa Gadgil, Catalina Vajiac, Tina Townes, Luke Neureiter, and Katie Mazaitis—thank you for believing in our FlaSH team. Collaborating with you all to build a deployed monitoring system has been the most rewarding part of my doctoral research. Outside of research, I loved being part of the CMU and Pittsburgh community, especially my friends — whomst I am grateful to for keeping my dreams afloat.

My research has been shaped by the insightful collaborators I've had. I want to thank my committee members, Rayid Ghani and Matt Biggerstaff, for their input in refining the implementation aspects of my thesis. Beyond my thesis, working with ACHD, ACDHS, PADOH, participating in InsightNet, interning at IBM Research, and mentoring public health departments through the CSTE network has broadened my understanding of computational tools and their real-world impact. I am particularly grateful to Deb Bogen, Chris Scott, and Skyler Speakman for their insight and encouragement.

Finally, I want to acknowledge those who laid the foundation for my academic journey [12, 60, 61, 63, 93, 99, 100], my teachers, both academic and otherwise, and my family and parents.

Public health surveillance and computational epidemiology are impactful and challenging research areas. I am hopeful that researchers will continue to explore innovative computational approaches that strengthen and support critical domains like public health and healthcare.

Abstract

Detecting individual data sequences corresponding to actionable events in large-scale, dynamic data streams, also known as data monitoring, is a challenging computational problem with applications across multiple domains. Specifically in public health, these data sequences can correspond to events like outbreaks or quality issues directly impacting downstream decision-making and outbreak response efforts. However, as the volume of public health-related data continues to grow, traditional machine learning algorithms for anomaly or event detection, designed for smaller datasets, become increasingly ineffective – for example, by outputting tens of thousands of uninformative alerts that lead to reviewer fatigue. These challenges are exacerbated by the noise, non-stationarity, and incompleteness of public health data and hinder the ability of domain experts to perform data monitoring.

My thesis enables domain experts to monitor large-scale data streams via novel ranked-list based algorithms that address the question, "Which data should be examined first, and why?" In contrast to traditional approaches that use statistical alerts, the output list of the top-ranked data prioritizes data reviewers' attention so that they remain engaged with the algorithmic outputs. These underlying algorithms, designed to be simple, scalable, and generalizable, include (1) ranking outliers from limited-history, nonstationary, noisy data streams with weekday effects, (2) reranking extreme outlier data points across large streams, and (3) ranking top anomalous subsequences of any length from dynamic, partially observed data without sampling.

Evaluating these algorithms and the overall approach in offline and deployed settings show strong results. For instance, when paired with custom user interfaces, the approach enabled a 53-fold increase in monitoring efficiency for data reviewers performing data monitoring at the Delphi Group at Carnegie Mellon University for over two years, allowing them to detect over 200 noteworthy data issues from 15 million new data points each week. This monitoring approach directly supports efficient and accurate public health surveillance and can readily be deployed at the state, national, or international level to enhance the effectiveness of public health data-driven decision-making and the core algorithms can be relevant to other critical monitoring domains.

Contents

Introduction

1.1 Contributions

This thesis introduces a novel approach and algorithms for monitoring large-scale data streams in critical settings designed to better align with the needs and workflows of human data reviewers. Existing monitoring systems often rely on human data reviewers inspecting alerts generated using rigid statistical thresholds or heuristics. When applied to modern data volumes with complex statistical properties, these approaches tend to fail. Instead, this thesis develops:

- a human-in-the-loop approach and fully deployed system built atop
- 3 novel interpretable and efficient algorithms that scale expert input for event detection:
    1. **FlaSH (Flexible outlier ranking method):** A customizable algorithmic ranking approach that scales expert feedback and constraints, allowing for context-aware outlier detection despite the data noise, nonstationarity, and incompleteness that are common in real-world data.
    2. **OutsHiNes (Reranking extreme outliers):** Identifies a new machine learning problem, multi-stream outlier ranking, and a solution algorithm rooted in extreme value theory, adapted for nonstationary and noisy data.
    3. **Enlighten (Anomalous subsequence detection & system development):** An algorithm and complete system that data reviewers use to identify and analyze anomalous data subsequences of any length.

Data monitoring is important in public health settings. It is used to detect important events, such as disease outbreaks and data quality issues, that are important for downstream data users like data scientists and public health decision-makers. For example, data scientists can build better models that exclude outliers or uncover informative disease dynamics that might otherwise

go unnoticed. Nevertheless, the known theoretical and statistical limitations of data monitoring at scale become apparent and pressing in public health due to the prevalence of real-world statistical properties of the data and the critical nature of the events. Progress forward for data monitoring at scale requires new computational approaches.

The strongest validation of the practicality of this approach is that it has been deployed in practice for two years and counting. To ensure practical utility in this setting, the above algorithms were designed, developed, and evaluated over several months in deployment with real-world public health data reviewers at the Delphi Group at Carnegie Mellon University. By working directly with data reviewers and users, the limitations of traditional alert-based monitoring systems became clear (see Appendix A) and inspired an approach that aligns with real-world data review practices. The final rank-based, human-in-the-loop anomaly triage system works by scaling reviewer intuition and prioritizing reviewer attention. Our evaluation includes surveys and longitudinal deployment studies that have shown that this system significantly improves public health data event detection processes, making it, as far as we know, the only open-source, statistically sound, scalable, and deployed approach for modern public health data monitoring. As the proposed methods also address computational challenges of data monitoring at scale in a statistically rigorous manner, they can also be applicable in other critical domains.

1.2 Motivation and Research Approach

> *Data saves lives. Better data saves more lives.* -United States Centers for Disease Control and Prevention (CDC) [40]

Background

Public health data curators, such as the CDC [40], UN [110], and WHO [2], regularly publish aggregated population-level time series data from various traditional and public health-adjacent sources. These data streams, referred to as indicators [41], are used for resource allocation, disease tracking, and identifying health disparities [30]. An advantage of these data curators is that they can monitor large volumes of data for critical events, or unexpected patterns in the data,

that may indicate system failures or meaningful shifts in public health dynamics [73].

Monitoring is particularly vital in public health, where curators are responsible for producing high-quality data streams [21, 85] and preventing costly attribution errors from downstream data users [87, 88]. Curators have unique advantages. They have aggregate data available at a scale larger than individual data providers. They also can have more data and compute their their downstream data users to detect critical events across multiple streams [2, 55, 112]. These events tend to belong to one of the following categories:

- *Data quality issues*, like shifts, errors, & delays in data collection and reporting [8, 11, 29]) which can have downstream impacts.
- Changes in *underlying disease dynamics*, like outbreaks [19].

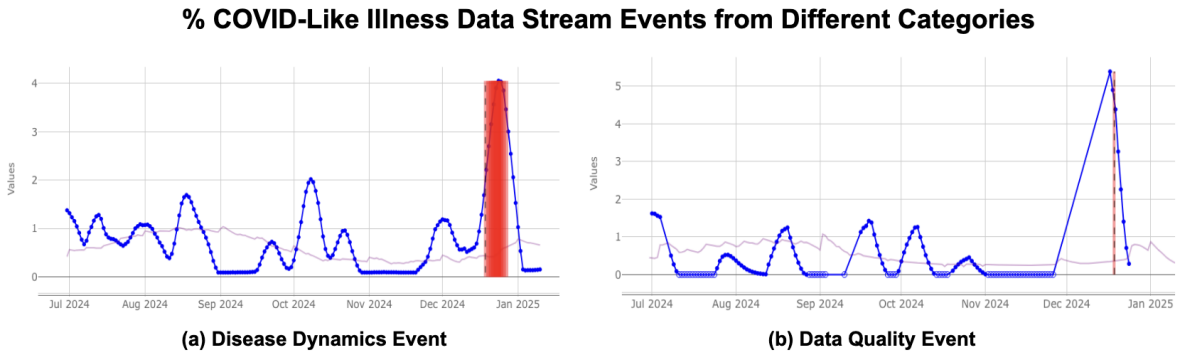


Figure 1.1: Examples of unexpected data points classified as events (highlighted in red) belonging to these different categories in public health data. Delphi's % COVID-like Illnesses indicator is calculated from Doctor's Visits data [96].

Plot (a) shows an increase in a respiratory illness indicator that is consistent with an outbreak, which represents a change in the *underlying disease dynamics*.

Plot (b) shows many days of reported data as 0 between November and December 2024, then a few weeks of missing data in December 2024, followed by a very high value for a respiratory illness signal.

According to one reviewer's analysis of this geospatial region, similar data streams, and external information, this data suggests that there was a *data quality* event, like delayed data reporting from the months of November and December 2024.

These events categories and how they manifest in the data correspond to different parts of how public health data is collected and reported (see Fig 1.1).

- *Data quality* issues typically stem from measurement/reporting errors in the data reporting pipeline, which can include imperfect human recorded data, breakdowns in reporting at multiple stages, changes in data definitions or aggregation, and data censored for individual privacy.
- *Disease dynamics* can represent changes in the ground truth of illnesses transmission.

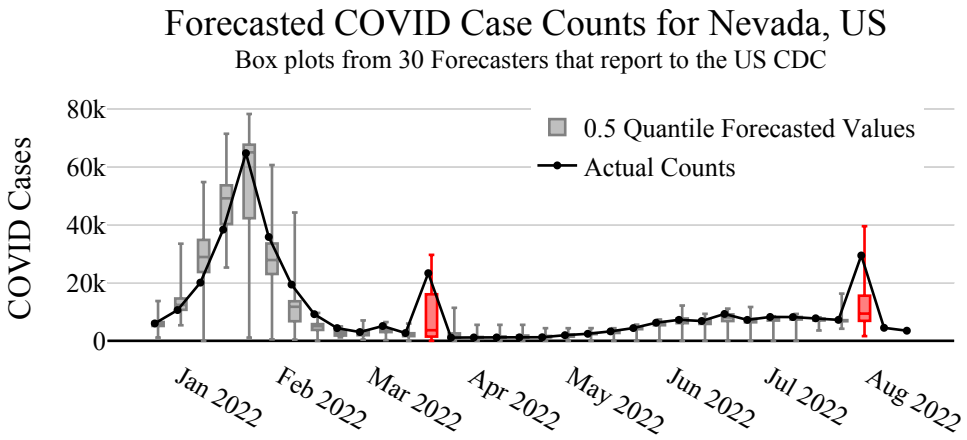


Figure 1.2: *Data quality* changes in case counts, shown by the large spikes in March and July 2022, when cases were trending down, resulted in similar spikes for predicted counts (red) from multiple forecasts that were then sent to the US CDC. This forecast is one of hundreds reported weekly and was likely missed by most in any downstream applications.

Once the data is identified and triaged as events by a data reviewer, they can be sent to stakeholders. However, these events are not equally relevant to all stakeholders. For example, modeling teams may care about data quality events relevant to data preprocessing, like single point outliers, while public health departments may care about the disease dynamics events and do not want to misattribute a data quality issue for an outbreak, like in Fig. 1.2.

Nevertheless, these events and their relative severity to different downstream data users are difficult to distinguish statistically through the data alone because different events can manifest similarly in these data streams. Instead, domain experts (e.g., epidemiologists, policymakers) need to use some type of external context (e.g., policy changes, public health reports) to triage these unexpected data sequences as events with different severity for downstream users. Then, these data users can readily use the classified, contextualized, and annotated data events they care

about. For example, during the COVID-19 pandemic, the modeling community used human-reported data quality events to revise forecasts [117], and data users also wanted to use events identify regions experiencing unexpected disease transmission, particularly in smaller geospatial regions without local public health departments or resources to monitor newly collected data streams.

Despite the demand and importance of event detection through data monitoring, for decades [55], reviewers have increasingly struggled to monitor incoming public health data thoroughly. One issue is that the data sequences corresponding to important events can be subtle [69], and detecting them requires careful attention that quickly exhausts limited data reviewing resources. In the current big-data setting for public health data, this constraint clarifies several limitations in existing approaches [27]. For example, existing methods can be uninformative because their output is highly sensitive to parameter tuning that needs to be continuously performed as the frequency of events change over time. Or, these methods produce too many alerts to be usable in practice. Monitoring challenges like these are likely to become more widespread as data modernization initiatives from curators [89, 118], although necessary, have consistently increased the volume, variety, and velocity of public health data [10] (modern data).

These factors culminated in an emergency at the Delphi Group at Carnegie Mellon University (Delphi). Delphi is a public health data curator for indicators related to respiratory illnesses in the United States [1]. As part of its curation process to provide various aggregate data sources free and open source to the public, Delphi monitors its data for events relevant to its users – including public health authorities, researchers, and the public. In line with large scale public health modernization initiatives [40], Delphi's daily data intake has increased over 1000× in the past three years, enabling more varied, timely, and high resolution data streams to be published [72, 96, 119]. While curators like Delphi only work with aggregate data—meaning they cannot detect events by analyzing individual records as some providers might, they are uniquely poised to identify and analyze unexpected data given that they can centrally access data from multiple

[1] In this thesis, the data used is from the Delphi Group and can be accessed through their open-source APIs [95].

data sources. Historically, they've used standard state-of-the-art anomaly or event detection approaches on smaller data volumes. However, as multiple stakeholders have pointed out, these existing methods are flawed. They are unable to find important 'needles' that users can take action over from a very large haystack of data. This need informs the thesis research question:

Research Question: How can computational limitations in monitoring large volumes of heterogeneous (modern) public health data streams be addressed?

This thesis presents a computational approach that involves designing both a novel system for data monitoring and developing new monitoring algorithms that support the system.

Preliminaries

The data monitoring **system** design must account for several nuances. For example, the monitoring process must provide flexibility to accommodate user and context preferences regarding "unexpected" or anomalous data [101] that correspond to data events. In fact, rigid systems, which fail to adapt to evolving user needs for event detection over time, are a primary limitation of prior approaches [15, 17], as described by public health experts [50], because they fail to adapt to evolving user needs for event detection over time without considerable effort. The proposed approaches should also focus on unexpected data *patterns* instead of data validation or algorithms that need external labels/metadata that are quickly outdated given the nonstationarity of public health data [105][2]. Additionally, all incoming data should be processed directly, rather than using sampling techniques, to prevent a common issue where regions with lower populations have fewer of their events detected due to sample size and noise. Beyond regional variability, the framework must be robust to changes in indicator sets, regional data quality [74], and shifting correlation structures as public health conditions evolve [43, 57]. This ensures the system remains useful to a large set of public health data curators, including USAFacts, JHU CSSE, The New York Times COVID Data, and the CDC. Finally, practical constraints—such

[2]Labeled data is also highly subjective and tends to only available in small samples [103].

as finite reviewer attention, computational resources, and data update cycles that limit real-time processing—must be addressed in the approach design.

Research Approach: Continuous, human-in-the-loop, anomalous subsequence monitoring system across all data streams to identify, "Which events should be examined first, and why?"

Continuous data monitoring involves (1) running a detection algorithm per data stream, [20, 59], like a control chart method, that (2) produces an alert, like when the method's resulting p-value falls below a threshold [18], that (3) reviewers inspect [126]. Continuous approaches are used in a number of domains to find real time anomalies in data streams (instead of identifying anomalous times or historical anomalies) [25, 36] . However, these alerting algorithms break down in modern public health data settings, as supported by attempts detailed in [19] and discussed in the Acute Phase Approach section.

Prior works are fundamentally limited or not applicable in this setting. First, similar approaches in networks, systems, and IoT literature rely on assumptions that do not necessarily hold for public health data. Public health data is incomplete and often an estimate, making it incompatible with many root cause analysis or formal verification methods. Further, other approaches focus on edge/node computing or sampling strategies because of limitations on communication protocols and privacy constraints [123, 124, 127]. In contrast, public health curators receive all their data over the course of a day and have no such limitations. These curators also need to process all their data, including streams aggregated at higher resolutions, because differently aggregated streams may include additional individuals that were not available or accounted for at more granular resolutions. Finally, by processing all the data they receive and not sampling, curators can be sure that all available data points are considered and identify problems in the data curation pipelines that are geography independent [2, 35, 55, 102]. So, popular methods that rely on dimension reduction [43, 57] or identify entire streams as anomalous are not relevant for this type of data-level monitoring.

Still, approaches for data-level monitoring do not scale with or perform well on modern pub-

lic health data. These approaches, including those embedded in public health monitoring systems [3], like **ESSENCE** [18, 79] system[4], are variations of the standard outlier detection method process. Take the World Health Organization's District Health Information System (DHIS) [3]. It produces an alert when a summary statistic (min, max, z-score) exceeds a baseline (determined by a stream's own history, the national stream, or streams from similar indicators at that geography) by a static value, like 10 %. While this approach is mathematically straightforward [121], using it in practice requires considerable manual effort for parameter tuning and data review as it is only intended for use across a limited number of streams (e.g. limited its intended use case in districts or counties). At scale across millions of heterogeneous and nonstationary streams, these approaches introduce multiple hypothesis testing errors [51]and necessitate manually updating thousands of thresholds and rules before new data arrives for the continuous setting. Even more statistically sophisticated alerting approaches, such as RAMMIE [83] and its extensions[5], can generate an extremely high and highly variable number of alerts when deployed across a large geography.

The challenges of these approaches, both statistically and in practice, have been documented by public health practitioners in papers like "What can you really do with 35,000 alerts a week anyway" [27, 50] , and tutorials/research from the International Society For Disease Surveillance through efforts from Michael Coletta, Wayne Loschen, and Howard Burkom. Yet, a statistically sound solution for data-level monitoring is needed.

Acute Phase Approach

The above limitations of existing approaches were validated after implementing and testing historical monitoring and surveillance algorithms on Delphi's data.

[3][24] overviews biosurveillance systems, including ESSENCE, RODS, INFERNO, BioSense, BioPortal, and NYC Syndromic Surveillance Systems. Others, including [44, 48], WHO's DHIS2, SAGES, and EARS, are also notable

[4]ESSENCE is the premier tool for Syndromic Surveillance (public health monitoring) in the United States [38, 39], and is widely used, even in local public health organizations.

[5]These methods are used by Public Health England to monitor national public health data.

During the acute phase of the COVID-19 pandemic, Delphi's data users wanted to surface unexpected data corresponding to important events ("finding a needle in a haystack.") Over two years, I served as a developer, engineer, and data reviewer to adapt existing approaches for monitoring with no sustained success. For example, a straightforward approach involves adding outlier scores directly to the data [107] using the standard Gaussian outlier detection equation. Yet, this approach not flexible enough to accommodate users' varying preferences for identifying underlying events and it was unclear what these scores meant in practice– an important requirement to stakeholders [19, 55]. It would have also doubled the size of our database. Another approach was similar to the DHIS approach, where our implementation set *alerts* for z-scores calculated using a rolling window that were above an adaptive threshold. This resulted in tens of thousands of daily alerts that, when reviewed, tended to contain few events and take up considerable human reviewer time. While we used the aggregate total number of alerts per provider as a heuristic of processes gone awry (similar to [27]), these alerts were eventually turned off after several months of parameter tuning, and other reviewers reverted to sporadic, manual inspection of the data.

The fundamental failures of existing monitoring approaches for modern public health data highlighted methodological gaps, as previously noted by practitioners and biostatisticians [19, 20]. These challenges include statistical issues, such as excessive false positives; computational issues, such as processing time and storage constraints; and practical issues, such as the inability of data reviewers to prioritize data for event detection, leading to delayed responses.

Approach Design

Based on the background and preliminaries, the three challenges this approach needed to address are:

C1: Informative Detection Algorithm: The underlying algorithm for ranking is central to separating random variation from substantive changes, but finding approaches flexible enough to meet the needs of different users is challenging. In addition, the "correct" event detection algorithm is dependent on the current state of public health (e.g., at the start of an influenza wave) [17]. Because updating the algorithms manually is costly, and parameter tuning requires long

data history that may not be available, the current best approaches are manual review or review based on multiple generic outlier detection algorithms [18].

C2: Addressing Overwhelming Alerts: These occur when multiple possible temporal, spatial, and value-range rules (explicitly or via a model) are applied to millions of data points, resulting in tens of thousands of alerts. Reviewing these alerts is taxing for reviewers because they are difficult to prioritize and focus on, which may undermine trust in the alerting algorithm itself. Manually tuning thresholds is labor-intensive, and increasing the thresholds using multiple hypothesis corrections [20] or waiting for consecutive days of alerting [1] produces data point outliers independent of relevant context, which rarely require human review. Moreover, the subtle anomalies that indicate important events are often missed [18, 91].

C3: Identifying Anomalous Subsequences: Data reviewers need relevant context (situational awareness) to identify consistent and actionable changes across and within multiple data streams. Specifically for anomalous subsequence detection, existing approaches include trying to find consensus for the anomaly across different data sources by combining p-values [19] or splitting a sequence into distinct subsequences based on data drift. However, these methods and their parameters (1) can be meaningless to domain-expert data reviewers, (2) do not necessarily work on non-iid streaming data, (3) may rely on consistent relationships between different streams, and (4) are susceptible to the data quality issues and unique statistical properties that characterize public health data.

These challenges are the root of the growing disconnect between public health surveillance systems and those they are meant to serve. For nearly two decades, practitioners have found that these monitoring systems fail to meet the needs of the individuals they were built for , from [55] to a recent survey from over 90 health practitioners [50]. Further, as many of these systems were built from a statistician's point of view, over time, they were not able to retrospectively account for engineering challenges in data processing and computational complexity, or for the system users' needs, such as transparency through clearly declared thresholds, parameters, and a desire to have an understanding of their data (situational awareness) [6, 36]. In fact many practitioners [19, 126] share sentiments like:

The current disconnect among algorithm developers, implementers, and users has ... foster[ed] distrust in statistical monitoring and in biosurveillance itself [103].

When looking to the future of data monitoring systems, building one that supports triaging events that correspond to changes in public health dynamics and data quality changes while implicitly accounting for statistical, engineering, and reviewer perspectives may be able to address the widening gap between theory and practice for public health monitoring [126].

Outside of these methodological challenges, there is little consensus on **evaluating** monitoring systems or methods [126]. Most public health monitoring system *guidelines* [6, 17, 19, 20, 113, 126] emphasize evaluation using realistic data and practical metrics. Yet, many *evaluations*, especially related to respiratory illness, have started to rely on synthetic data. In fact, because identifying outliers in large volumes of data requires substantial domain expert effort it is often *assisted by the very algorithms that are meant to be evaluated* (circularity problem) [17, 126]. Practical evaluation metrics also differ based on the perspectives of developers, engineers, and reviewers, who each have different needs from a public health data monitoring system. Finally, some types of evaluation are not possible or can be difficult in big data settings, where humans are not able to review and label millions of data streams [86] (e.g. generating recall bounds). Accordingly, metrics and evaluation strategies meaningful for modern public health data surveillance were designed with domain experts, which are synthesized into the following evaluation criteria: [37, 103, 126]:

E1: Correctness How valuable is the approach and its evaluation [50, 126]?
E2: Feasibility Can it run over all recent production data for data curation organizations in a timely manner [21, 50]?

The primary metric reviewers care about is **efficiency**, or the number of events detected per minute. Given the large volume of data and the limited time/attention of human reviewers, surfacing more events for reviewers to triage and publish than the status quo is a considerable

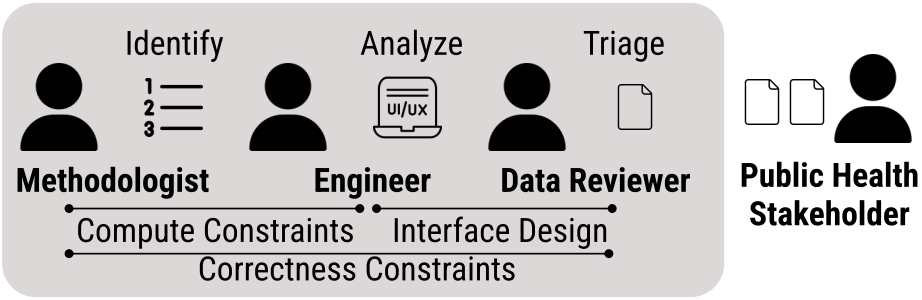


Figure 1.3: The redesigned monitoring approach takes advantage of method, engineering, and reviewer experts strengths. This design necessitates novel algorithms described in this thesis.

improvement in increasing the data utility. This metric is tied to precision and directly informs the method's design and evaluation criteria.

1.3 System and Algorithms Overview

This approach and underlying methods are inspired from the failures of the Acute Phase Approach, data characteristics from other completed projects in Sec. 5 and, weekly interactions with domain experts in public health – especially observing, surveying, and interacting with Delphi staff.

The redesigned system is a human-in-the-loop data monitoring workflow/system (Fig. 1.3), that combines novel analytic methods, engineering design, visualization techniques, and insights from human-computer interaction to help reviewers analyze unexpected data patterns (anomalies) in public health-related data. Instead of an alerting system, where users are only shown values that cross a p-value threshold (alerts), this approach **ranks the most extreme of all recently updated data** by how much it warrants reviewer attention. Reviewers can then decide how many data points they want to investigate, which prioritizes a reviewer's time and does not require them to review a certain number of alerts flagged via a (somewhat arbitrary) threshold.

Then, novel algorithms for each of the listed challenges [**C1-3**] were developed and evaluated across relevant metrics as described next.

FlaSH: Flexible Outlier Detection Method

Full paper in IJCAI '23 [65]

Most outlier detection methods (e.g. Gaussian outlier detection) have a recombinant structure [14, 84] with the following steps: data processing, baseline creation, test-statistic/residual generation, threshold setting, and alert generation. The differences in the vast majority of these methods can be attributed to changes in the baseline creation step (e.g. underlying forecasting method) than any other step. Additionally, domain experts are tasked with setting thresholds, which can be more unintuitive at scale than knowing the appropriate baseline to use. This discrepancy provides a unique opportunity to rely on experts to inform baseline creation, and instead develop new methods *to scale expert intuition*.

FlaSH is a customizable outlier detection method that incorporates reviewer data expectations in the form of a model and difference metric for the baseline creation and test-statistic generation step. Then, these distances and expectations are used to quantify 'unexpected' data.

The core method is about scaling expert intuition – which needed a new approach. For example, model-based residual thresholding methods can be useful to generate alerts and rankings. Standard approaches use parametric models of the residual distribution to return p-values sensitive enough to meet thresholds like < 0.01. However, using parametrized models for residuals across millions of data streams likely results in many streams where the model is inaccurate. On the other hand, empirical residual distributions can be more robust across millions of data streams. Still, the empirical residual models are limited by the data history available in nonstationary public health settings.

FlaSH overcomes this data history limitation to scale intuition, and makes a model-based approach feasible in public health data streams, by pooling together historical empirical test statistic distributions from streams that share the same indicator, geographic level (e.g., county, state, nation), and geographic parent (e.g., all states in a country). Then, new test statistics (ϕ) are ranked in comparison to their respective sibling-stream empirical test statistic distributions.

I conducted two rounds of evaluation using a custom evaluation interface In this setting, all data points were evaluated from chosen data streams over time. I started with a binary classification evaluation (IRB 1; Appendix) but, the thresholds for outlier classification varied by

reviewer, which inspired the ranked list (vs. alert classification) approach. These studies were preregistered (Preregistration 1, IRB 2) and evaluated using a revised FlaSH approach. FlaSH met targets for feasibility, outperformed 13+ outlier detection methods on standard binary and ranking metrics, as well as metrics important to Delphi members, and, crucially – like how many additional points were identified using the algorithm over what would have been obvious to a human in the first place (Assistive Rank).

OutsHiNes: Addressing Overwhelming Top-Ranked Outliers

Full Paper at AAAI '24

FlaSH was initially deployed on only a handful of indicators, so only a few points were tied as top-ranked outliers. However, when we expanded FlaSH over all indicators and all recently updated data (which includes recently updated historical data), reviewers were suddenly overwhelmed by the thousands of tied, top-ranked outliers that resulted from the FlaSH approach. To address this problem, I first formalized the problem as a new machine learning task, which we called the multi-stream outlier ranking task. In this task, an algorithm takes as input values from univariate outlier detection methods and outputs rankable scores over all recently updated data per day. Because this is a new problem, there are no directly related existing works, and existing approaches adapted to this setting performed poorly.

OutsHiNes tackles the multi-stream outlier ranking task and identifies the most extreme outliers given test statistics from a univariate outlier detection algorithm, like FlaSH. In this setting, the top-ranked data points were evaluated for precision per day, over time. I conducted ablation, correctness, and deployment evaluations [IRB 3 and Preregistration 2] and OutsHiNes led to a 9.2x speedup over the manual investigation in identifying data quality changes.

Enlighten: Surfacing Anomalous Subsequences

After OutsHiNes was deployed, reviewers, for the first time, could find the point data quality and public health changes they most cared about from all of Delphi's data. Yet, investigating changes, especially if they occurred over multiple days, and understanding any higher-level insights from the outliers still required a lot of effort from reviewers. To more directly support

process assurance for reviewers, (1) with the data monitoring time, we designed a novel interface for data review using participatory design, and (2) I developed new complementary methods to detect and aggregate anomalous sequences.

For (1), the interface and visualizations alone improved the efficiency of reviewers by **6x** over the **9.2x** previously documented using the OutsHiNes algorithm, as detailed by a 3-month longitudinal study and survey [IRB 4 and Preregistration 3]. Additionally, the novel Enlighten method to identify anomalous subsequences was evaluated in an (a) offline survey, (b) in online reviewer performance, and (c) against the outputs from OutsHiNes. The final results support a 1.7x improvement in the number of higher-level events detected across multiple geographies, indicators, or time, and an overall *288x* increase in reviewer efficiency over the deployed manual baseline as calculated using data points reviewed/minute. In another evaluation focused on precision and recall, scores were evaluated considering the a) top-k listed rows b) random sampling conditioned on output Enlighten scores, and c) a uniform random sampling (given that the scores are zero-inflated by design).

Summary and Impact: This thesis' system and methods considerably improved data monitoring processes in theory and practice. The overarching system re-envisioned how engineers, method developers, and public health data experts interact for data monitoring, and the resulting novel methods are applicable to monitoring and surveillance processes across domains. As a testament to its practicality, this system has been deployed for at least two years for the Delphi Group at Carnegie Mellon University.

FlaSH

This section is adapted from [65], which appeared in IJCAI '23.

Summmary: FlaSH (**Fla**gging **S**treams in public **H**ealth) ranks the most recent real-time outliers from data streams corresponding to a public health indicator (appx. 3000 streams) that

are relevant to data quality reviewers. FlaSH accomplishes this through simple, scalable, and intuitive models that explicitly capture the statistical properties of public health data, like nonstationarity, noise, and weekday effects. To address challenges in evaluating unsupervised outlier detection methods in time series data, I also developed and conducted a classification and ranking evaluation of FlaSH's performance using input from several data reviewers. In these evaluations, FlaSH matches or outperforms standard outlier detection methods, including recent deep learning baselines, using only a lightweight autoregressive (AR) model for forecasting.

Inputs:

1. Geospatial streams from one indicator at different granularities (county, state, national).
2. Univariate point prediction method (that may have multivariate inputs)

Outputs: Ranking outlier detection score, ϕ as an intermediate output

Evaluation: Binary and ranking expert feedback using a custom user interface on Delphi's data streams.

Open Source Code, Preregistration[67]

2.1 Background

Prior Work: Detecting data irregularities that correspond to events across many sources is uniquely challenging for typical outlier detection methods, leading to a range of failure modes observed in our experiments. First, deep learning outlier detection methods struggle with the large number of time series, each with a short history and rapid distribution shifts [90] typical in public health settings. Moreover, high computational costs mean these methods scale poorly to real-time operation over thousands of distinct time series. Second, simpler statistical methods [15] are not attuned to the specific structure of public health data and struggle to accurately identify events [120]. Third, neither class can leverage features of public health data streams that could assist with diagnosing events. Some source-specific public health outlier detection methods [34] that operate on data streams before Delphi receives them do not have publicly available methods, but the continued presence of important events in those streams that impacts downstream data stakeholders, especially related to data quality or errors from the curator, highlights

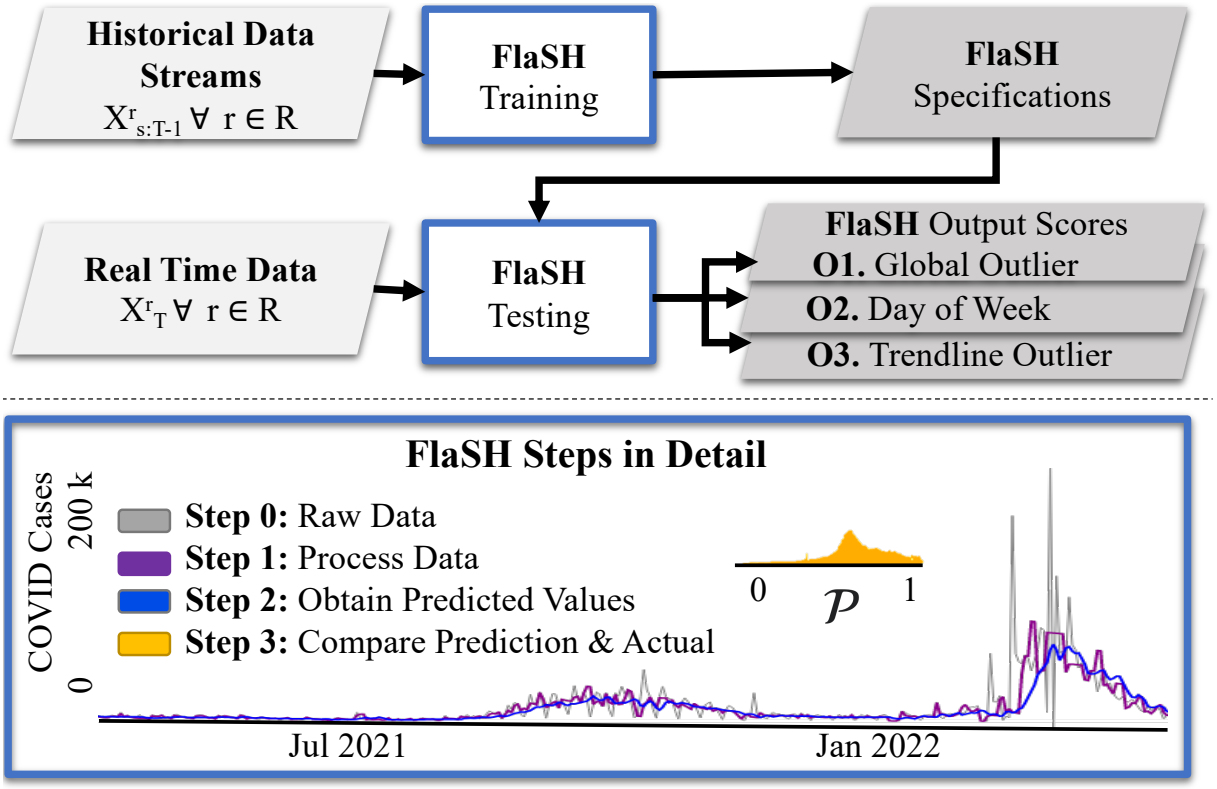


Figure 2.1: In the FlaSH outlier ranking method, data stream inputs are processed through FlaSH to generate informational outlier scores. FlaSH itself has three steps. The raw data (gray) is processed [S1] (purple), and model m is used to predict future values [S2] (blue). Then, the historical performance of model m is captured with the test statistic distribution (gold), and this distribution is used to compare predicted and actual values [S3].

their limitations.

User Preferences Based on our exploratory analysis on data streams[1] from different sources at different geographic resolutions until December 2021, I identified that Delphi's stakeholders care about data that deviate strongly from the recent trends (e.g. case counts were rising last week, but today's count is low) or from the recent trends of close geographic regions. These phenomena, which we call **trendline outliers**, are the most difficult for humans to detect and can indicate critical irregularities in the context of recent data.

[1]The streams were from National, Texas, New York, LA County (CA), and Loving County (TX) sourced from JHU CSSE, Department of Health and Human Services, Google, and USA Facts.

2.2 Formulation and Method

Problem Formulation We denote a single data stream as a time series X_t, $t = s...T$. Here, s is the starting time for the stream analysis[2], and T is the current time. When discussing multiple geographic regions, we use X^r to denote the stream for a given quantity in geographic region r (e.g. the stream of COVID cases in a given US county).

Suppose that $X_{s:T-1} \sim m$ for some $m \in \mathcal{M}$, where $\mathcal{M}$ is a set of models. We test the hypothesis that the most recent point in the stream is drawn from the same model ($H_0 : X_T \sim m$). If the observed data has a low probability under this hypothesis, it means that X_T was likely not generated from the same model m as the historical data. This sudden shift from the data-generating distribution indicates a potential irregularity that signifies a notable event (e.g. disease dynamics change or data quality issue). We conduct the hypothesis test by first calculating a test statistic measuring the discrepancy between observed values and values predicted by m. We then obtain a p-value by comparing the real-time test statistic value to a historical distribution of test statistics $\mathcal{P}$. FlaSH instantiates this entire method via 3 steps: data processing, obtaining predicted values, and comparing predicted and observed values (Fig. 2.1).

S1: Process Data. We want to fit a model m such that points with irregularities appear in the most extreme tails of the m's predictive distribution. However, training m on less subtle outliers both distorts the model and inflates the tails of the distribution of prediction error so that more subtle deviations no longer stand out. We process the data to identify and impute these outliers before training. The key challenge in this step is to accommodate the statistical properties of public health data (see paper for more details).

S2: Obtain Predicted Values. After processing, we fit a parametric model m from a model class $\mathcal{M}$ that uses the history of the stream to predict future values. Choosing an appropriate $\mathcal{M}$ is nontrivial. Heavily parameterized models are unsuitable because of the limited data history available to tune the model and the rapid distribution shifts in the data. FlaSH uses $\mathcal{M}$: Linear Autoregressive (AR) models (lag=7) , where m is characterized by the linear weights, $\hat{\beta}$, fit

[2]Often, there is a ramp-up period before streams report reliable measurements, so we do not start at t=0.

during training. This class of models is preferred in public health applications for its simplicity and performance with *limited historical data* [81]. The remaining processed historical data (not used to fit the model) is used to generate predictions $\hat{X}_t$.

S3: Compare Predicted and Observed Values. Finally, FlaSH compares the observed and predicted values to test if X_T could have been generated from m given the historical performance of observed and predicted values. The critical decision in this step is the choice of the test statistic and construction of its distribution under the null hypothesis, which are complicated by short training histories and the resulting need to share information across geographic regions.

Test Statistic: To quantify the discrepancy between predicted and observed values, let N^r denote the total population of geographic region r. The day of week corrected observed values ($w(X_t^r)$, corrected to be comparable to the predicted values) and the predicted values ($\hat{X}_t^r = \hat{\beta} * w(X_{t-1:t-7}^r)$) are used to calculate the test statistic ϕ_t:

$$\phi_t = (P(w(X_t^r) < D))$$
$$D \sim \text{Bin}\left(n = N^r, p = \frac{\hat{X}_t^r}{N^r}\right)$$

Extreme values of the test statistic indicate that the observations were much bigger or smaller than expected, given the predictions.

Comparison Distribution: Each stream model's typical performance discrepancy is specified by a distribution $\mathcal{P}^r$, composed of test statistics $k_{30:T-1}^r$, that compares observed values and the predicted values for the out-of-sample historical data $X_{30:T-1}^r$. However, there is often too little history to approximate the null distribution of an individual stream effectively. Accordingly, we define the pooled test statistic distribution $\mathcal{P}$, specified by $\bigcup_{r \in R} k_{30:T-1}^r$, where R is all the counties in a state if r is a county, else R is all states and territories in a nation because these streams share geographic context and tier (**sibling streams**). Note that pooling is enabled by the design of our test statistic, which is chosen to ensure comparable distributions across regions (e.g. via normalizing by the population).

The final output is a list of real-time points ranked by how extreme their test statistic is via

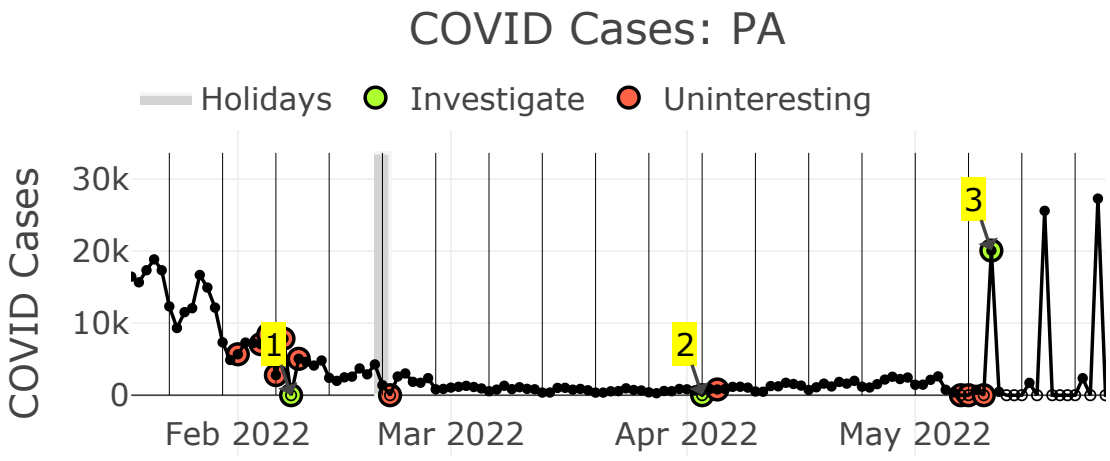


Figure 2.2: Example of a Survey Task. Respondents click on the time series plot to mark points as unevaluated, uninteresting, or warrants investigation. They also rank points that warrant investigation, and these rankings appear on the plot in yellow. Respondents could zoom, pan, and see a 7-day average per graph.

the transformation $|2p - 1|$, where p is the p-value for the real-time test statistic in the pooled historical test statistic distribution $\mathcal{P}$. This transformation ensures that the most outlying points (from either distribution tail) will top the ranked list.

2.3 Survey and Analysis

To understand FlaSH's performance on empirical data, I designed an interactive web survey (Fig. 2.2) for Delphi's engineers and researchers to evaluate data points with enough context to triage and categorize data as events. First, respondents classified candidate data points from a public health stream as 'warrants human investigation' or 'uninteresting.' Then, they ranked (with possible ties) the subset of these candidates they think would warrant additional human inspection. They were also asked how likely they would have flagged each point for human review had it not been identified by an algorithm ('unlikely,' 'somewhat unlikely,' 'neither,' 'somewhat likely,' or 'likely'). This allowed us to measure the value added by the algorithm over what would have been obvious to a human (Assistive Rank).

To form a candidate set of evaluation points, I took the union of the top outlying points output

by both FlaSH and 8 previously proposed outlier detection methods [3] given all historical data so that the candidate set is limited to points that are considered anomalous by some method. This empirically meant the candidate set comprised of points that were at least interesting enough to classify and rank. I evaluated the algorithm's performances in a realistic setting of only 60 days of history for training (12/21/2021-1/31/2022). Our test set was the following 100 days (2/1-5/12/2022).

We compare FlaSH off-the-shelf outlier detection algorithm baselines implemented in TODS, which have in-built data processing [S1] and prediction comparison [S3] steps, just like FlaSH. Additionally, for an ablation study, I compared results from the TODS AR model implementation, which has the same model class $\mathcal{M}$ as FlaSH, to a mixed implementation (Mixed), where the processing step [S1] is the same as FlaSH, and the prediction comparison step [S3] is from TODS.

FlaSH is computationally scalable per indicator. Each algorithm was trained on the full 3341 JHU CSSE COVID-19 case streams with 60 days of history on our deployment hardware. Deep learning algorithms did not finish training within one day (DNF), which is when the data would be updated. Training time can only increase for these deep learning implementations as historical data increases, and while GPU acceleration may benefit deep learning models, such specialty hardware may not be available in many public health settings.

FlaSH performs well on outlier detection metrics. Table 2.1 shows the 95% CI of various traditional binary and ranking outlier detection metrics across all participants per algorithm. In the binary analysis, points identified by the majority of respondents as to-investigate were marked as outliers (ground truth). To calculate binary labels from each algorithm, I took the top k points per algorithm, where k denotes the number of human-identified outliers for a stream, ranked according to the algorithm's outlier scores, as the predicted outliers for binary classification tasks and compared these results to the ground truth labels. On average, FlaSH meets or exceeds the performance of all baselines in the binary analysis. FlaSH performs slightly bet-

[3] DeepLog [35], Telemanom (Telem.) [54], Variational Autoencoder (VAE) [7], Local Outlier Factor (LOF) [16], Lightweight Online Detector of Anomalies (LODA) [92], Isolation Forest (IF) [78], k-Nearest Neighbors (KNN) [9], and Linear AR Model [47]

Model Class		**AR**		
Implementation		TODS	Mixed†	**FlaSH**
Training (s)		10.1±0.3		169±0.8
Binary	Accuracy	0.78±0.02	0.71±0.04	0.8±0.03 ✓
	Bal.Acc.	0.68±0.02	0.59±0.06	0.73±0.05✓
	F1	0.54±0.05	0.43±0.09	0.64±0.08✓
	ROCAUC	0.79±0.02	0.73±0.06	0.75±0.06
Ranking	Distance	0.66±0.39	1±0	0.62±0.39✓
	RBO	0.84±0.1	0.89±0.08	0.84±0.1
	Corr.	0.2±0.63	0.42±0.45	0.37±0.57
Assistive Rank*		8.00±6	3.66±1	1.33±0.7 ✓

Model Class		**DeepLog**	**Telem.**	**VAE**	**LOF**	**LODA**	**IF**	**KNN**
Implementation					TODS			
Training (s)		DNF	DNF	DNF	8±0.2	71±0.1	DNF	7±0.08 ✓
Binary	Accuracy	0.8±0.04✓	0.6±0.04	0.76±0.04	0.69±0.01	0.68±0.04	0.79±0.04	0.74±0.03
	Bal.Acc.	0.72±0.05	0.42±0.03	0.67±0.07	0.55±0.03	0.54±0.05	0.7±0.07	0.62±0.05
	F1	0.63±0.07	0.19±0.07	0.53±0.12	0.33±0.08	0.34±0.09	0.56±0.11	0.42±0.09
	ROCAUC	0.82±0.05✓	0.42±0.07	0.68±0.06	0.62±0.04	0.44±0.07	0.66±0.08	0.65±0.07
Ranking	Distance	0.63±0.36	0.83±0.24	0.66±0.37	0.66±0.39	0.7±0.39	0.67±0.39	0.66±0.39
	RBO	0.84±0.1	0.84±0.1	0.89±0.07	0.88±0.08	0.93±0.06✓	0.91±0.11	0.88±0.08
	Corr.	0.43±0.54✓	−0.13±0.71	0.18±0.64	0.21±0.67	0.24±0.69	0.17±0.68	0.22±0.66
Assistive Rank*		2.33±0.7	41.33±38	32.00±57	24.00±40	70.67±51	47.33±39	5.33±5

* Mean rank of points somewhat unlikely or unlikely to be caught by human
† Mixed model with FlaSH data processing [S1] and TODS comparison of predicted and observed values [S3].

Table 2.1: Summary of algorithm comparison with 60 days of historical data. ✓marks the best algorithm in each row.

ter than DeepLog, an unusable but performant deep learning method. Some model classes like Telemanom and LODA performed poorly on the ROC-AUC score because while they identified global outliers very clearly, they failed to capture trendline outliers. For the ranking analysis, each algorithm's ranking of the subset points available that a majority of participants marked as warrants suspicion was compared to each respondent's rankings using Hamming distance (lower is better), Ranked-Biased Overlap (RBO) [115], and swap correlation (corr). Once again, FlaSH performs comparably to DeepLog and is competitive with the other algorithms. Finally, FlaSH shows strong improvements over the TODS AR implementation. By using data processed using FlaSH's first step (Mixed) [S1], the AR model can better build a null model of the data. Still, because the TODS outlier scoring uses the absolute difference between the predicted and observed values to rank points, the mixed approach performs poorly on streams with small case counts [S3], as reflected in the results.

FlaSH can complement human judgment. FlaSH ranks useful points that were unlikely to have been inspected without computational assistance (via an algorithm identifying the point), as shown in the Assistive Rank row of Table 2.1. This metric is computed from the set of points that (a) the majority of humans rated as warranting investigation after a full examination, and (b) at least 40% of such respondents said that they were "unlikely" or "somewhat unlikely" to have identified the point without algorithmic assistance. We reported the mean rank assigned to such points, where a smaller rank indicates that the algorithm would prioritize those points more for human inspection. FlaSH consistently ranks these points near the top of its list (more so than other methods), indicating that FlaSH can usefully direct human attention to points that would have been missed otherwise. This is a result of FlaSH's emphasis on discovering trendline outliers, which our prototyping showed are difficult for humans to recognize in public health data streams.

Conclusion: FlaSH can scale to the data streams per indicator required, perform well on traditional outlier detection metrics, especially compared to the best-performing deep learning models, and crucially, prioritize points for human review that would not have been discovered otherwise. Based on FlaSH's empirical performance and design, it was deployed as part of Delphi's

daily workflow in February 2023. It ran on selected streams, and a data reviewer inspected the ranked, outlying points. As reviewers prioritized different events, I modified the point prediction method to FlaSH to detect those respective outliers as seen next in the OutsHiNes work, where the predictions came from an Exponentially-Weighted Moving Average model.

The next challenge was to generalize this approach to the 100+ indicators with real-time updates to historical data that Delphi receives per day via **OutsHiNes**.

OutsHiNes

This section is adapted from [68], which appeared in AAAI '24.

Summary: Outlier detection methods typically return < 0.01 of the data. When they are are applied to hundreds of indicators (data streams) with hundreds of recently revised points across thousands of geographies, they output too many maximum-priority outliers to review (e.g. tens of thousands ranked as number), in addition to mathematically less meaningful outputs, like an overall ranking based on a naive combination of smaller ranked lists (with different contexts and granularity dependent on the amount of historical data available). This task is formalized as multi-stream outlier ranking, where algorithms rank the outputs of univariate outlier detection methods applied to each of a large number of data streams. Our approach for this task (OutsHiNes) uses a combination of hierarchical networks and extreme value analysis to rank outliers across multiple streams. In expert evaluations, the best-performing approach (across all metrics considered) used OutsHiNes, and data reviewers report identifying noteworthy data points **9.1x** faster while using OutsHiNes than baselines.

Inputs: Test statistics for all recently acquired data from univariate outlier detection methods.

Outputs: Ranked list of Outlier Scores

Evaluations: Interactive human binary and ranking evaluation on Delphi's data streams and deployed performance.

Open Source Code, Preregistration [66]

3.1 Background

Univariate outlier detection methods, like the previously described FlaSH, identify outlier data points in individual streams as needed, [15, 53, 94], can operate over data streams with different

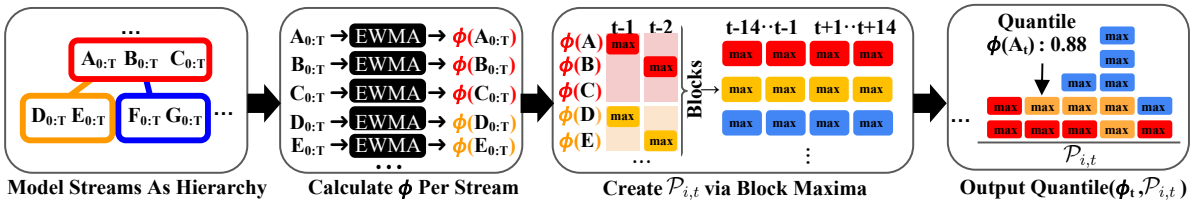


Figure 3.1: OutsHiNes models streams as a hierarchy, applies a univariate method across all streams to calculate ϕs, creates $\mathcal{P}_{i,t}$ via block maxima per indicator i and day t across sibling streams, and finally, outputs the quantile of new ϕ_t to $\mathcal{P}_{i,t}$ to rank.

properties (e.g. scale[1], noise, and outlier patterns) [28], and are fully parallelizable over large sets of streams. Yet, as they are currently used, these methods return too many alerts (14k-20k from 3-4m points for FlaSH in July 2023).

Motivated by this setting, we introduce a new task called **multi-stream outlier ranking**, where the goal is to rank the overall highest-priority outliers across all data streams, thus prioritizing expert time. Algorithms for this task take values produced by any univariate outlier detection method applied independently to each of a large number of data streams as input and must rank them in a way that leverages the historical behavior of the underlying univariate outlier detection method on each stream.

As this is a new task, prior work has not explicitly considered this task. However, existing algorithms can be adapted as baselines because they prioritize outliers within a single stream or a small set of similar streams. Generally, these ranking algorithms are already baked into outlier detection methods and score new data points by comparing them to an empirical reference distribution, $\mathcal{P}$, formed from historical data. As the size of $\mathcal{P}$ increases, so too does the resolution of quantiles possible that determine empirical scores used for outlier ranking. The most common approach, which we call **threshold ranking**, identifies a dynamic outlier threshold per stream and returns a binary classification based on values that exceed the threshold [53, 75]. The other approach, which we call **sibling ranking**, from FlaSH, considers data from multiple, similar streams that share a parent. Both rankings return too many alerts, among other issues, because (1) their $\mathcal{P}$ is generated from only one or only a few streams, (2) they use the whole stream's

[1]E.g., the raw COVID case count in a rural county (0-10) will be much differently scaled than that in a city (0-1000)

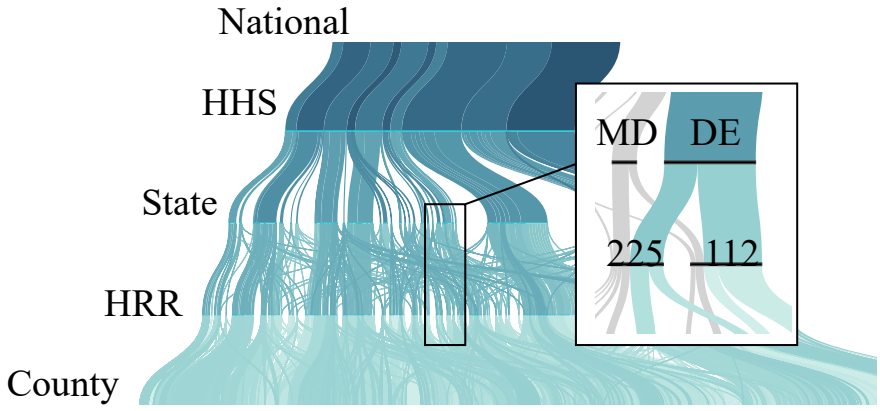


Figure 3.2: The geospatial hierarchy for public health streams covers 4270 regions. HRRs may serve multiple states like HRR 112 and 225 serve both D.E. and M.D. residents.

history instead of values that share temporal context, and (3) they have only a small and varying number of values in $\mathcal{P}$ because data is limited per stream and some streams have a number of missing days.

3.2 Notation and Method

In this section, we use the following notations for clarity. Each day, T, a curator receives data $d_{i,r}(t)$, where i is an indicator in the set of curated indicators $\mathcal{I}$, r is a region in $\mathcal{R}$, where $\mathcal{R}$ contains all the regions in Fig. 3.2 from different geospatial tiers (e.g. county, state or national), and t is a historical day $0 \leq t \leq T$. This geospatial-temporal data forms data streams, where each stream, identified by i, r, consists of $d_{i,r}(t) \, \forall \, t \in [0, T]$. In public health data, the number of data streams is large ($|\mathcal{I}| \times |\mathcal{R}|$) and far exceeds the history (T) available per stream. Higher-tiered regions (e.g., states) may include data from more people than expected by combining information from county-level sub-regions due to data privacy reasons, so these streams must be analyzed separately. Regions in $\mathcal{R}$ form a **geospatial hierarchy** (see Fig. 3.2). Two context-specific tiers that capture rich geospatial and epidemiological relationships are hospital referral regions (HRR), which are regions that share a hospital system [31], and HHS groups, which contain nearby states [49]. Experts use these hierarchical relationships (e.g., parent, sibling (which share a parent), and child streams) when ranking outliers by how different they are from similar points

in sibling streams.

Algorithm 1 OutsHiNes Ranking

Using Block Maxima to make $\mathcal{P}_{i,t}$ for indicator i and day t
Input:$\phi(d_{i,r}(t)) \, \forall \, r \in \mathcal{R}$
Output:$y(d_{i,r}(t)) \, \forall \, r \in \mathcal{R}$

l = regime length
for $\mathcal{R}_{sib} \in \mathcal{R}$: `#Stream Aggregation Dim.`
  $P_{i,\mathcal{R}_{sib},t} = \{\}$
  for $h \in [t - l/2, t) \cup (t, t + l/2]$: `#Temporal Dim.`
    `#Block Maxima`
    $P_{i,\mathcal{R}_{sib},t} = P_{i,\mathcal{R}_{sib},t} \cup \max(\phi(d_{i,r}(h) | r \in \mathcal{R}_{sib}))$
$P_{i,t} = \cup_{\mathcal{R}_{sib} \in \mathcal{R}} P_{i,\mathcal{R}_{sib},t}$
$y(d_{i,r}(t)) \leftarrow q(\phi(d_{i,r}(t)), P_{i,t}) * \frac{\log(|P_{i,t}|)}{\log(\max |P_{i,t}|)} \, \forall \, r \in \mathcal{R}$

Method: First, OutsHiNes (shown in Fig. 3.1) models the hierarchical relationships in data streams per indicator i. The resulting *hierarchical network* implicitly captures contextual relationships across all data streams. Then, OutsHiNes inputs the test statistics output from any univariate outlier detection method applied to recently updated values in these hierarchical streams ($\phi(d) \, \forall \, d \in \{d_{i,r}(t) \, \forall \, r \in \mathcal{R}, t \in [0, T]\}$). These test statistics (ϕ) measure the degree of agreement between predicted and observed values, and extreme ϕ indicate a potential outlier. The univariate outlier detection method must ensure that, per stream, the ranking of each ϕ is expected. Yet, because ϕ is computed per stream, and some streams ill-suited to the chosen univariate outlier detection method may consistently return extreme ϕ, ϕ alone does not provide an informative ranking across many data streams and must be contextualized. Finally, OutsHiNes outputs the real-valued quantile of test statistics from all regions in the hierarchy at time t from an empirical reference distribution generated per day, per indicator $\mathcal{P}_{i,t}$.

OutsHiNes creates $\mathcal{P}_{i,t}$ by using hierarchical relationships and extreme value analysis on data from time close to t. We adapt the block maxima approach from extreme value analysis, which traditionally splits a data stream into equally-sized non-overlapping data blocks (e.g. one block per month) and calculates the maximum value in each block to form $\mathcal{P}$ [42]. Highly-ranked outliers are points d for which $\phi(d)$, the test statistic, is large even with respect to the reference distribution $\mathcal{P}$. The intuition is that if ϕ is miscalibrated for a particular data stream and regularly

returns ϕ with large values, $\mathcal{P}$ will contain many such examples, and a new data point must have even more extreme ϕ to stand out. Yet, traditional block maxima does not apply to streams with limited, nonstationary, non-i.i.d data [98] like public health streams.

Instead, our approach changes the block sizes, to address both limited data history and nonstationarity in streams of ϕ. Each block has a temporal dimension (day, week, month, etc.) and a stream aggregation dimension (typically one stream). Aggregating homogenous, or similar, streams per block is a known way to calculate block maxima over more data, but identifying an appropriate homogeneity test is difficult [77]. Instead, we designate homogenous streams as those that share a parent $r \in \mathcal{R}_{sib}$. Aggregating streams across $\mathcal{R}_{sib}$ creates blocks of similar regions. Then, to address nonstationarity, we limit the range of block maxima calculations to data that is temporally similar to the time being evaluated (a regime of length l) so that $\mathcal{P}_{i,t}$ is generated from days that are the most similar to the time considered. The block maxima calculated over these blocks define $\mathcal{P}_{i,\mathcal{R}_{sib},t}$, which contains the maximum ϕ per indicator, per R_{sib} per t in the regime. To make an empirical reference distribution with many observations that capture extreme ϕ from all $\mathcal{R}_{sib}$, these $\mathcal{P}_{i,\mathcal{R}_{sib},t}$ for $\mathcal{R}_{sib} \in \mathcal{R}$ can be pooled together to create $\mathcal{P}_{i,t}$, which represents the distribution of recent extreme ϕ equally weighted from each set of geospatial regions, as shown in Alg. 1, lines 5 & 6.

If OutsHiNes is applied separately across different indicators, as it is for Delphi's data, scores from $\mathcal{P}_{i,t}$ with more observations should be weighted higher because these scores are more refined. Thus, OutsHiNes scales each quantile by $|\mathcal{P}_{i,t}|$ divided by the log of the maximum possible observations, (e.g. $\log(|\mathcal{R}_{sib} \in \mathcal{R}| * \text{regime})$) to return a score ($y$) in [0, 1], as shown in Alg. 1, line 7.

OutsHiNes is preferable to other ranking algorithms because it ensures that $\mathcal{P}_{i,t}$ does not over-represent any region or time, it compares every $\phi(d_{i,r}(t)) \, \forall \, r \in \mathcal{R}$ to the same $\mathcal{P}_{i,t}$, unlike sibling or threshold ranking, and finally it has more granular output scores because it has more observations that characterize $\mathcal{P}$ than alternatives.

	Task		*Ranking Method*	
			Thresh.	Opt. Thresh.
Timing/Indicator (s)		Generate ϕ		
UOD	Delphi-Deployed	57.9 ± 35.17	* 6.71 ± 2.59	* 6.6 ± 2.51
	FlaSH	458.81 ± 146.21	* 5.33 ± 2.32	* 5.2 ± 2.12
	AR	36.13 ± 19.85	5.61 ± 3.92	4.32 ± 3.02
	Isolation Forest	420.59 ± 270.15	65.04 ± 43.16	61.84 ± 41.92
	DeepLog	6.52k ± 4.398k	53.2 ± 36.13	52.79 ± 35.95
	Telemanom	6.16k ± 4.449k	64.65 ± 47.08	68.26 ± 52.38
# **Ties/Indicators**				
UOD	Delphi-Deployed	-	*15.81k ± 1.97k	* 6.05k ± 2.54k
	FlaSH	-	* 22.02k ± 1.97k	* 8.08k± 2.13k
	AR	-	42.11k ± 18.84k	7.02k ± 3.714k
	Isolation Forest	-	89.87k ± 67.23k	39.80k ± 21.93k
	DeepLog	-	32.29k ± 26.07k	14.24k ± 3.44k
	Telemanom	-	78.78k ± 39.10k	53.31k ± 32.42k

	Task	*Ranking Method*	
		Sibling	OutsHiNes
Timing/Indicator (s)			
UOD	Delphi-Deployed	319.67 ± 172.55	50.58 ± 46.40
	FlaSH	326.54 ± 160.64	45.85 ± 46.84
	AR	270.61 ± 156.14	58.97 ± 47.10
	Isolation Forest	190.58 ± 104.08	39.06 ± 29.10
	DeepLog	188.78 ± 102.44	38.97 ± 28.97
	Telemanom	293.36 ± 159.5	57.75 ± 43.44
# **Ties/Indicators**			
UOD	Delphi-Deployed	585.33 ± 549.13	6.67 ± 0.65
	FlaSH	159.33 ± 23.37	7.67 ± 2.85
	AR	127.67 ± 18.29	11.67 ± 10.51
	Isolation Forest	3.87k ± 2.23k	20.67 ± 20.88
	DeepLog	260.33 ± 63.78	18.0 ± 2.99
	Telemanom	215.0 ± 89.44	14.0 ± 2.26

Table 3.1: Baseline Comparisons with the blue highlighted deployed combination.

3.3 Evaluation and Results

We calculate standard outlier detection metrics comparing the expert-labeled data to outputs of different univariate x ranking methods we collected from a sample of 6383 streams. All streams are from 1. Outpatient doctor visits for COVID-related symptoms, 2. % COVID-positive antigen tests, and 3. The univariate methods (UOD) were 1. Delphi-Deployed (Described in paper): an exponentially weighted moving average forecasting model (EWMA) we tailored for Delphi using the FlaSH process, 2. the original FlaSH method, 3. Telemanom, 4. DeepLog, 5. Isolation Forest (IF) and Linear Autoregressive Models (AR), with 2-6 using the TODS implementation [75].Estimated % of new COVID hospital admissions based on claims data. The four comparison ranking methods are: Threshold ranking [75], Optimized Threshold ranking[2], Sibling ranking [65], and OutsHiNes ranking.

Approach Feasibility As shown in Table 3.1[3], all tested combinations are theoretically feasible as they executed in under a day. Of the ranking methods, OutsHiNes is more than 4.5x faster than sibling ranking and generally faster than TODS threshold ranking. Because OutsHiNes computes $P_{i,t}$ across a l-day window (parallel) of ϕ instead of all historical ϕ in a stream (serial), there may be more pronounced performance gains as T increases. OutsHiNes is also the *only* ranking method that produced a number of maximum tied points that reviewers could investigate daily.

Expert Evaluation In our offline expert evaluation, experts interactively inspected, classified, and ranked a subset of points in 8 streams that were tied using Sibling Ranking (14k+) but differentiated when using OutsHiNes. These selected streams allow us to test if OutsHiNes ranking matches that from experts on points that would otherwise have been tied according to sibling ranking, the previously best method. We display mean metrics per stream per person with a 95 % CI error bar for each combination of {outlier detection x ranking method} in Fig. 3.3. Binary

[2]The TODS threshold=0.9. The optimized threshold ranking is set to 0.99 to match the frequency outliers are expected [121]

[3]Because Delphi-Deployed and FlaSH are not in TODS, we implemented a comparison ranking method (indicated by *).

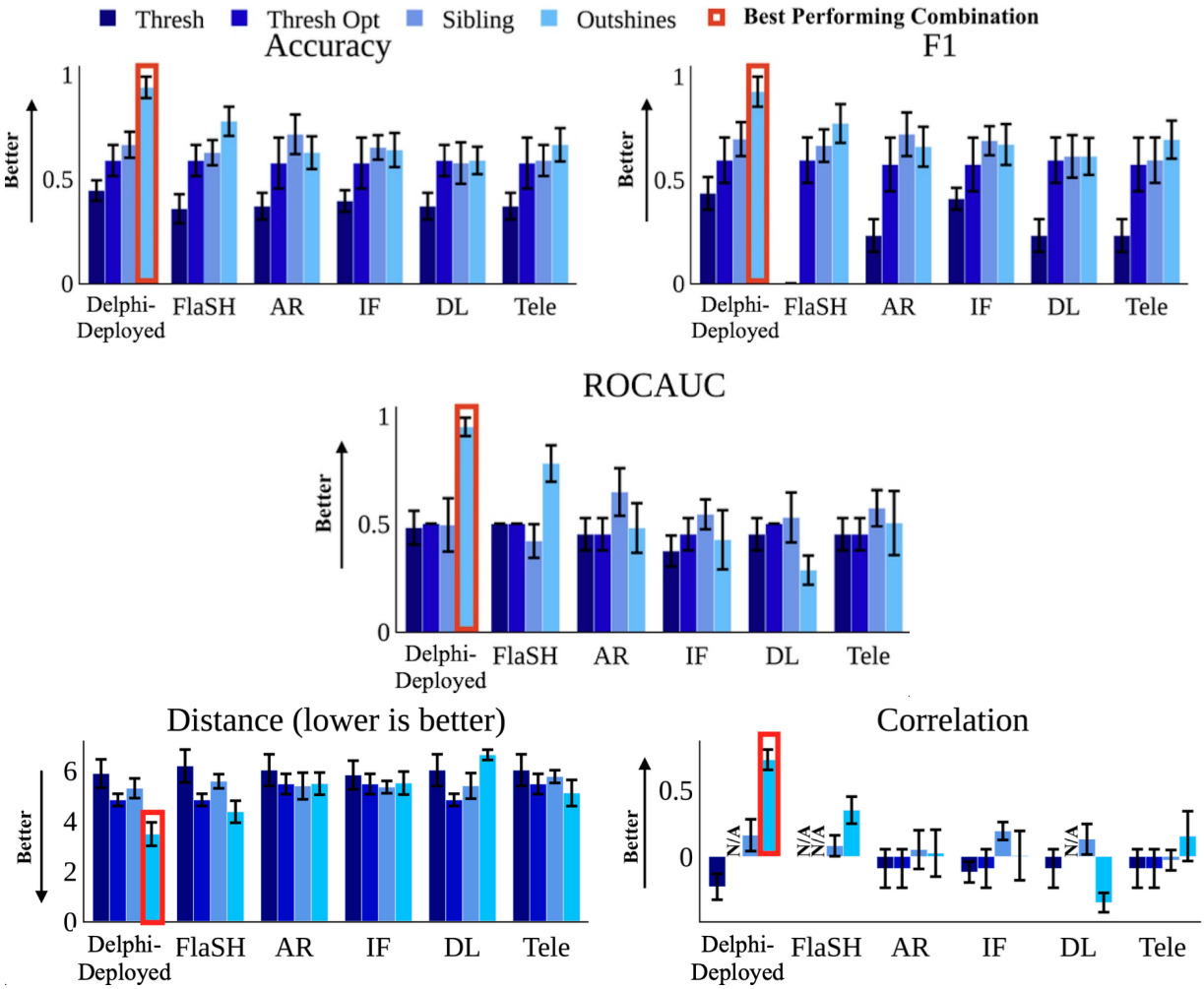


Figure 3.3: The red box highlights the best-performing combination of outlier detection x ranking method for standard binary and ranking outlier detection metrics. Correlations were N/A when the method returned all 0's or 1's.

metrics (Accuracy, F1, and ROCAUC) were calculated using the top-k points as the positive class, where k is the number of streams with a ranked outlier per person, and ranking metrics (Swap Correlation (higher is better) and Hamming Distance (lower is better)) were calculated using the respondent's absolute ranking. Our results show that OutsHiNes scores best match the expert ranking of all combinations tested, as per these standard metrics.

Deployed Performance Delphi's reviewers have used OutsHiNes in their daily outlier review process since April 2023. We report experts' performance metrics from using OutsHiNes from July 10th to August 5th. During this time, the daily data volume was 3.5 million $\pm$ 280k points, which OutsHiNes took 123.44 ± 189.16 minutes to process and produced 21 ± 5.5 ties/day (far fewer than sibling ranking on the same data: $14k \pm 1.7k$). OutsHiNes increased the rate of expert irregularity identification over the baseline of manual inspection, as shown in Fig. 3.4 by 9.13 ± 2.26**x**.

Comparing OutsHiNes to other ranking methods in deployment was not straightforward because experts felt that manual review was more fruitful than only analyzing a random sample of thousands of outliers, like those produced by sibling ranking. Still, for experimental completeness, for one week, experts split their time reviewing the top 10 data points from OutsHiNes and 10 random maximum-tied points from sibling ranking. In this direct comparison, experts found 4.02**x** as many outliers using the OutsHiNes points and at 3.96 ± 1.27**x** the rate.

Conclusion OutsHiNes provided the first-ever insight into outlier data points on a large scale for Delphi. Based on self-reports, experts valued analyzing alerts prioritized by using OutsHiNes over exploring a random subset of maximally tied outliers. Experts also felt that the feedback cycle with the research team was crucial in updating their FlaSH-input predictive model needs from an autoregressive model to the EWMA model.

However, it takes many individual point outliers to diagnose data quality issues, and often, multiple data points together are more anomalous than individual data points. To assist data evaluators with data diagnosis, the **Enlighten** approach combines visualization design and method generalization to improve monitoring outcomes.

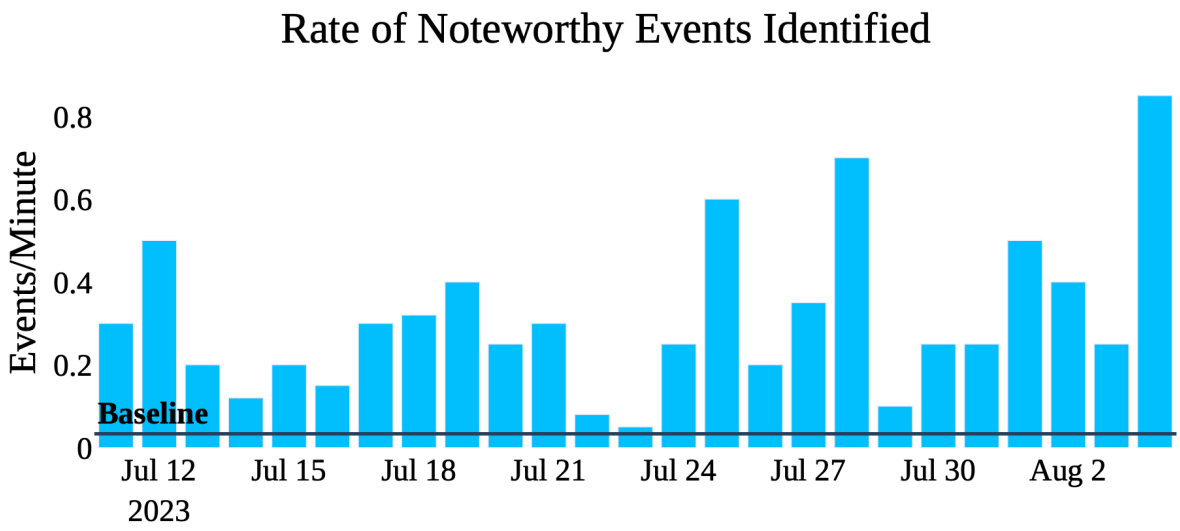


Figure 3.4: Experts can identify outliers of interest more quickly with OutsHiNes output than their alternative baseline at *scale* – manual review.

Enlighten

Summary: With OutHiNes, data reviewers needed to go through each ranked outlier using a basic review interface to triage the category and severity of issue (see Fig. 4.1). However, reviewers needed to go through hundreds of rows that often contain subsequent or similar types of outliers to gain situational awareness [1]. To enhance situational awareness, I (a) designed an overall incremental, participatory design approach to systematically improve the design interface with design choices made as part of a team of methodologists, engineers, and data reviewers, and (b) developed a method generalization to identify anomalous subsequences. Together, these approaches improve the efficiency and number of events detected.

Inputs: Test statistics (ϕ) for all recently acquired data from univariate outlier detection methods (e.g., from first steps of FlaSH).

Outputs: Ranked list of Anomalous Segment Scores in a Dashboard and System.

Evaluations: Interactive human evaluation on Delphi's data streams, deployed performance, and statistical comparison to outliers detected.

Preregistration[64]

Images in this section render best on a computer (multiple layers) and may not print properly. Please contact the author if you run into any issues.

Motivation: Providing data reviewers with situational awareness and relevant context during data review can support efficient and effective data monitoring. However, defining and providing relevant situational awareness is difficult. A good standard is based on how epidemiologists search for situational awareness in practice. A 2005 study had epidemiologists review 60 anomalous subsequences in public health data to find outbreaks (they did not find any). As part of their

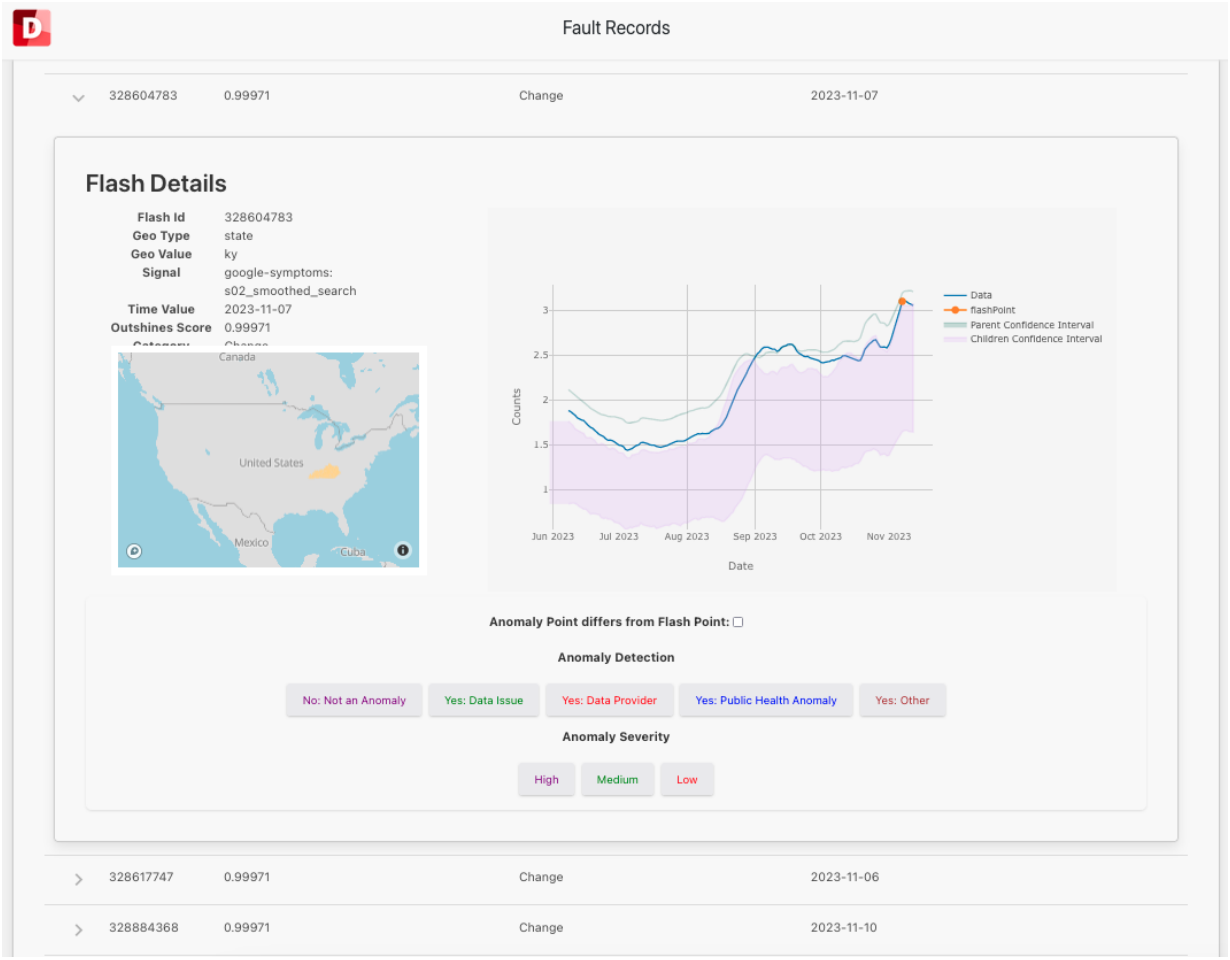
[1] Process-related issues impact multiple streams, from a handful to thousands

investigation, they needed to ask questions like, 'For how many days has the anomaly lasted?'" [106] and 'Are similar patterns found in adjacent regions?'. These are the types of higher-order questions we want data reviewers to understand because they correspond with situational awareness.

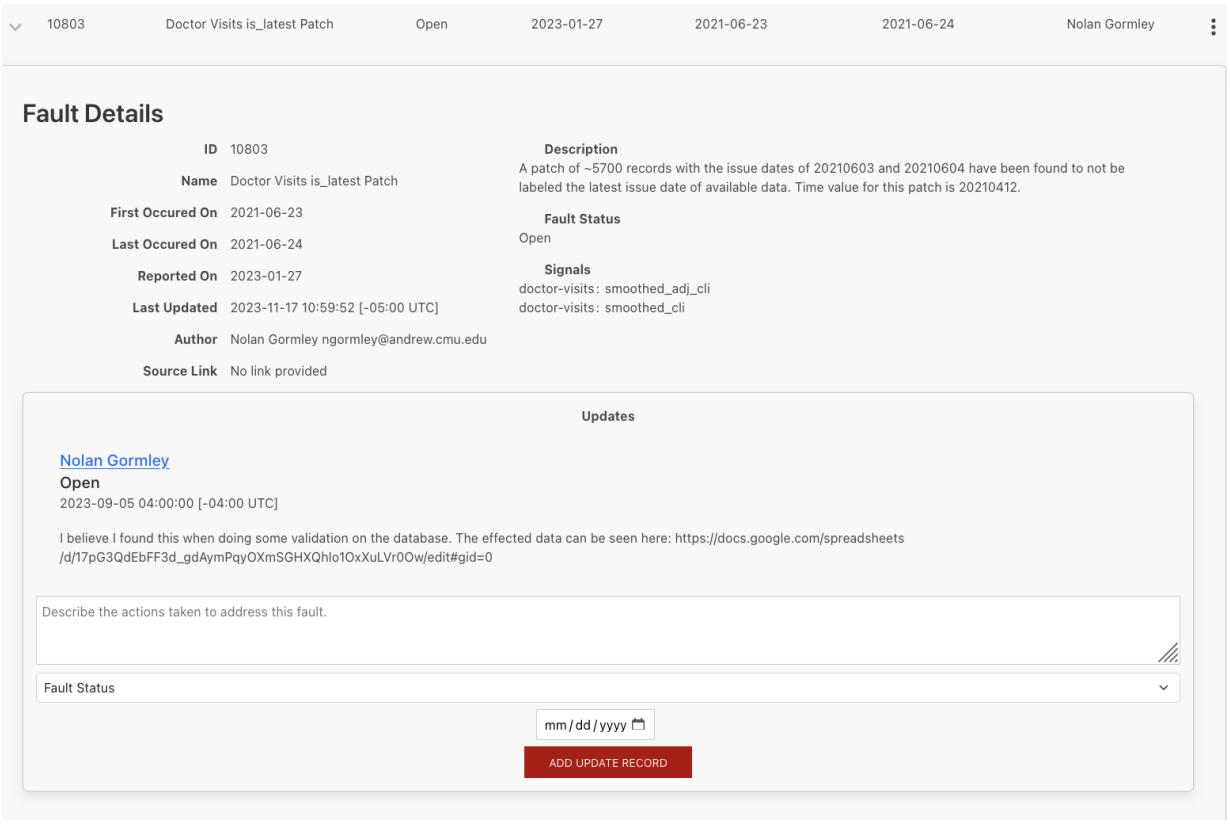
Towards this goal, we have two mechanisms to support situational awareness among reviewers triaging unexpected data:

(a) User Interface and Visualizations for Triaging : Many biosurveillance systems rely on highly customizable interfaces so that epidemiologists can quickly convey concerning aspects of the data [1]. I had implemented a Slack interface (Sec. A), where reviewers would be directed to an external site to view data streams, a dynamic HTML webpage that only showed the top 25 streams per day (OutsHiNes), and the final deployed version – a highly interactive, record-keeping web dashboard (implemented by Nolan Gormley and Richa Gadgil with design choices including Tina Townes and Catalina Vajiac). As part of this dashboard, we designed and evaluated the benefit of different visualization approaches on reviewer rates for completing data assurance tasks (via IRB 4).

(b) Methods for anomalous sequence detection within a stream: Many data quality issues span multiple days. Current systems address this challenge by providing an alert only if a p-value goes below the user-set threshold multiple days in a row [1]. This approach has two issues. First, sequential points alone may not be noteworthy, but their sequence can be concerning (anomalous sequences). Second, reviewers want to find these anomalous sequences as contextualized by other similar streams rather than trending subsequences within a stream (which is less indicative of a data quality issue or widespread outbreak). Other approaches, such as those which split the stream into different independent subsequences or that evaluate data subsequences of different lengths are limited in theory and practice, as explored in Sec 4.2.



(a) Dashboard



(b) Fault Record Keeper

Figure 4.1: Dashboard (a) that reviewers iterate through daily for triaging unexpected data as

4.1 User Interface Design Process and Evaluation

The key aspect of the UI is to support data reviewer triage. More generally, triage is a form of anomaly analysis that emphasizes a standard, structured classification. There are many cross-domain anomaly analysis systems that support data prioritization and contextualization from large volumes of data (e.g., [33, 82]). The core design features of these mechanisms prioritize data discovery algorithms and visualizations. Notably, they tend to treat different dimensions of data similarly. This dimension-agnostic segmentation is not appropriate for public health data, where the temporal and geospatial dimensions are the most important. There are other geospatial-temporal visualizations [22, 71]. In most of these systems, the geospatial segment of the data is the prominent dimension (like a map) with interactive panels for temporal data aspects. These methods have geospatial bias stemming from how smaller populations and regions are represented on maps, which can bias against regions with smaller geographical area or populations. Furthermore, these methods focus on the temporal dimension of the data more than the geospatial dimension, which is problematic for data reviewers. Regardless of the incompatibilities, the individual elements in these approaches, like maps, line plots, and filters [46] informed our initial approaches, which were limited in their speed and responsiveness.

To start, I documented and classified how data reviewers currently analyze data with and without a ranking. Reviewers typically performed four main actions:

- *Identify unexpected (outlier) data points.* These outliers might correspond to either a reporting error or to the early stages of an outbreak.
- *Contextualize outlier points.* This broader analysis ensures that significant events are not overlooked, even if they appear minor in isolation.
- *Structure and record findings.* These should be in a format that public health stakeholders can readily use.
- *Decide whether to continue reviewing.* This decision balances the reviewing urgency with

the reviewer's attention capacity and capability to perform high-quality event detection.

Identification was often based on recent news and exploring different data segments. Once an anomalous data point was identified, reviewers first prioritized the temporal dimensions of the data and analyzed individual data streams before expanding to other data dimensions. Then, they recorded their findings in paragraphs and, based on the severity of recent classified (triaged) data points, decided whether to keep reviewing. For reviewers to complete these actions, the system's UI and visualization needed to address the following considerations:

1. **Analysis Support.** Visualizations must provide context within a stream (including revisions) and about the event detection method, meeting stakeholders' need for transparency. Here, it is critical to balance avoiding oversimplification and cognitive overload to prevent misinterpretation. Some design questions we asked for this consideration are:
    - How should the system segment this data's dimensions for analysis to balance potential user overwhelm with providing the needed data for the reviewer to make an informed decision?
    - What combinations of visualization and interactivity techniques [46] can we use so that the reviewer is more aware of event detection choices (e.g., are they aware of changes in events due to data revisions)?
2. **Engagement:** Engagement is critical to sustained system use but can be affected by a few challenges. One such challenge we identified was *algorithmic fatigue*, where reviewers become less engaged with the system, for example, due to the volume of alerts. Another challenge, *algorithmic over-reliance*, happens when reviewers trust the system's results without conducting their own contextual analysis. One opportunity to address these challenges and increase engagement is to provide real-time situational awareness [128], which can help reviewers decide whether to spend longer reviewing data (e.g., there is an outbreak) or only check a few data points. Some design questions we asked for this consideration are:
    - How can we provide situational awareness to data reviewers?
    - How can we prioritize/filter the data we show to reviews so as not to cause algorithmic

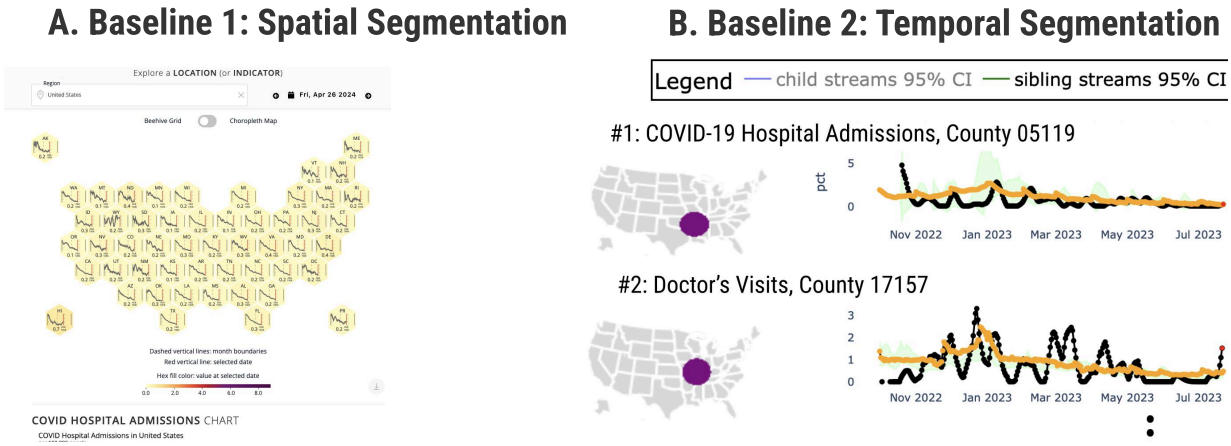


Figure 4.2: Previous iterations of the user interface for the reviewing system match general classes of visualization tools for geospatial data.

A. This system focused on geospatial data segmentation and was subject to the drill-down fallacy across dimensional layers.

B. This system focused on temporal data segmentation and displayed data streams with some context for similar regions (e.g., sibling streams come from regions that share a spatial parent, like states in a nation) and a map to orient the reviewer.

fatigue?

3. **Structured Event Detection Information** Reviewers need tools to correct or validate events they have flagged historically, even though data is processed in real time. Representing past events in the context in which they were viewed presents an ongoing challenge due to the revised nature of the data.
    - How can we standardize the unstructured triaging process?
    - How can we save past events in the context they were reviewed for quality assurance checks?

Baseline Approaches

Baseline 1: *Exploratory Interface*. Identifying events in exploratory systems via visual inspection (Fig. 4.2A)) requires drilling down several clicks, and reviewers have been shown to miss subtle but important public health data events when they rely on this type of visual inspection

[55, 65, 104]. Another challenge is that if there are several events with different strengths across different geographic tiers, any aggregation strategy could result in a drill-down fallacy [76], where reviewers could still end up needing to explore a combinatorial number of dimensions to locate the data event. For 120 weeks, reviewers used this exploratory approach for triaging.

As expected, reviewers missed important events, especially those in smaller regions and outside of the indicators that are displayed first on the interface. Then, between a) randomly choosing data filters, b) using multiple clicks to drill down to the raw data level, c) finding the appropriate regional tier responsible for the event by trial and error, and d) scrolling to compare indicator behavior across signals, whether relevant or not, reviewers became fatigued. This emphasized the importance of designing an engaging data review system.

Baseline 2: *Temporal Segementation Ablation*. Based on reviewer's emphasis on the temporal dimension as the most important dimension to segment on, we focus on interfaces that emphasize the time series aspect of geospatial data, like [33, 82]. In this approach, we displayed interactive line plots in a static HTML file for the top-k data streams (Fig. 4.2B). Here, we ensured that each data row had some relevant context for reviewers to complete their tasks. This is the starting point for this study.

Triaging System Design

Our design needed to address the questions data reviewers face, namely around decisions on how to segment data, acquire situational awareness, and understand data revisions. Specifically for the UI, we also needed to reconcile the challenge that data reviewers are under pressure and have little time for onboarding or learning about changing systems [23, 58] with the need to summarize and present large volumes of modern data. Our sequential modification strategy described here attempts to do both and also provide reviewers time to adjust to incremental changes and provide feedback.

Basic Interface

The basic interface (Fig. 4.5) segments the data so the temporal dimension is emphasized in analysis. It features a straightforward list of data points ranked only by their event score and

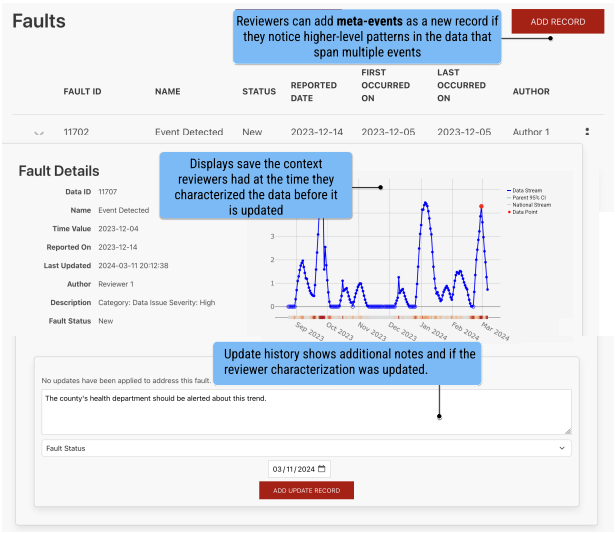


Figure 4.3: The record-keeping interface provides Delphi with enough data to understand why the reviewer triaged a specific event and allows reviewers to add updates and notes. They can also add a record for *meta-events* they notice across events.

presented in a time-series format. Reviewers can easily expand each data row, which includes a custom map to orient the reviewer and other stream properties. Each row also contains an interactive line plot with data streams. Our two key design elements here are: 1) any 0 values are represented as clearly visible open circles because they may represent missing or censored data, and 2) contextualization across tiers is controlled by legend items the reviewer can toggle to see data streams in the same tier that share the same regional parent (i.e., a sibling stream), as well as parent streams and child streams with a 95 % CI.

After reviewers analyze events, they triage the data by creating a record corresponding to the type of event (e.g., a data quality issue), its severity (low, medium, or high), and if the point identified was the source of the event (yes/no). These events themselves may lead to hypotheses across events that reviewers want to investigate, so structures to report, parse, and push *meta-events* to stakeholders are needed. These meta-events are very informative as they provide a broader context and help in understanding the overall data trends.

Once the reviewer submits their event characterization, the results are automatically saved in Delphi's event recording system (Fig. 4.3) for downstream processing. This system allows stakeholders with access to correct or update identified events because it saves *the context* in

which a data point was reviewed[2]. This is especially important as this data is revised, and over time, the values, and thus triage values, may change (see Fig. 4.7). Reviewers also use this interface to record *meta-events* that combine multiple events from individual data points into informative, higher-level phenomena.

Then to support this segmentation strategy, situational awareness, and analysis around data revisions, we incorporate the following modifications sequentially, as shown in Fig. 4.4.

M1: Filtering Data Points After Event Detection

Given the choice to focus on the temporal dimension of the data as the primary interaction segment, we need to design how users will access other data segments. However, preliminary analysis indicated that popular complex, multi-dimensional filtering strategies brought up worries that more complex filtering hypotheses using data fusion that intend to highlight very specific events would mute the more important, widespread events that appear with only a few filters. Another challenge to more popular combination filters was the number of possible filtering combinations in this setting, especially considering the desire for multiple filters (including exclusions) per category of data provider, indicator, and geographic region. We use this step to validate if the performance simple filtering strategy is appropriate for data review.

M2: Displays to Inform Data Point Investigations

Reviewers brought up that the simple news-feed format, which only provides temporal context, fails to provide them the situational context they need to triage potential events. This context is especially important because reviews are usually completed as the reviewer's first assignment each day, so they are missing information about potential indicator data failures or regions with outbreaks and may need to revise their annotations if they lack context.

Our approach supports situational awareness via two displays of OutsHiNes scores segmented and aggregated across geography and indicator. This is only possible because OutsHiNes scores can be compared across these dimensions wheras raw values cannot due to spatial hetero-

[2]Delphi hosts both the Record Keeper and the reviewing interface hosts these Svelte interfaces, containerized with Docker, and served using Apache.

geneity of public health data. Scores **s**(x) across different spatial tiers $e \in \mathcal{E}$, are aggregated across indicators $i \in I$, and time (T:T-7). For the map, the choropleth color value c for each county (on a scale of 0 to 1) is calculated using:

$$\forall r \in \mathcal{R}, \quad c(\mathrm{r}) = \frac{\sum_{e \in \mathcal{E}} \frac{\sum_{i \in I} \log\left(\bar{s}(x_{i,e(\mathrm{r}),T-7:T})+1\right)(/log(\mathrm{w})}{|I|}}{|\mathcal{E}|}$$

where e(r) is the region that subregion r belongs to at tier e. The log scores make the more extreme events appear more clearly on the map. The indicator display scores are calculated using

$$\frac{\sum r \in \mathcal{R}(s(x_{i,r})(T-7:T))}{|\mathcal{R}|}$$

.

In ways, this can be thought of as a fusion strategy once the previously incomparable raw data values have been standardized through a process contextualized for public health data.

M3: Event Evolution

Finally, as data is revised over time, the historical OutsHiNes scores also change with the additional data availability and presence of new events. Capturing this evolution of OutsHiNes scores across historical revisions communicates the uncertainty [23] of the OutsHiNes scores over time to reviewers. Our approach for capturing the evolution OutsHiNes scores is inspired by the design of industrial stagger charts [45]. It involves calculating the rolling mean and standard deviation over time, across data revisions, via Welford's online algorithms [116]:

$$\bar{x}_T = \frac{\bar{x}_{T-1} \times (T-1) + x_T}{T}$$

$$_T = \frac{_{T-1} - (\bar{x}_{T-1} - x_T) \times (\bar{x}_T - x_T)}{T}$$

We include a 1D heat map under the interactive time series plot to give reviewers the context of the event history and provide the average variance score across time. These help the reviewer understand the volatility in OutsHiNes scores over time, as shown in Figure 4.5.

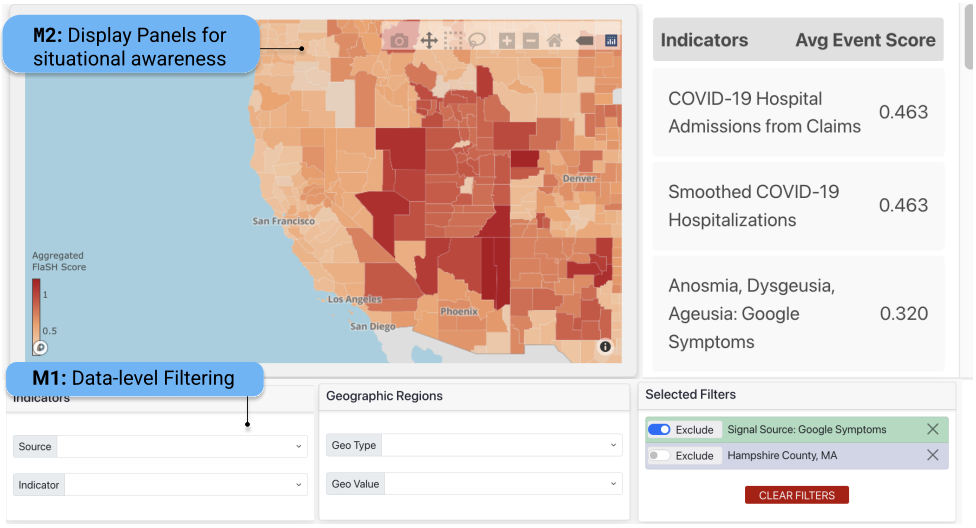


Figure 4.4: The initial displays support situational awareness and help reviewers get a sense of where data events may be found (**M2**). This can help them guide their review via the data level filters (**M1**). **M3** captures the impact of data evolution on OutsHiNes scores for each data row, and is shown on 4.5. Each of these modifications is used in conjunction with the basic interface's visualizations and after the offline event ranking to reduce data misinterpretation.

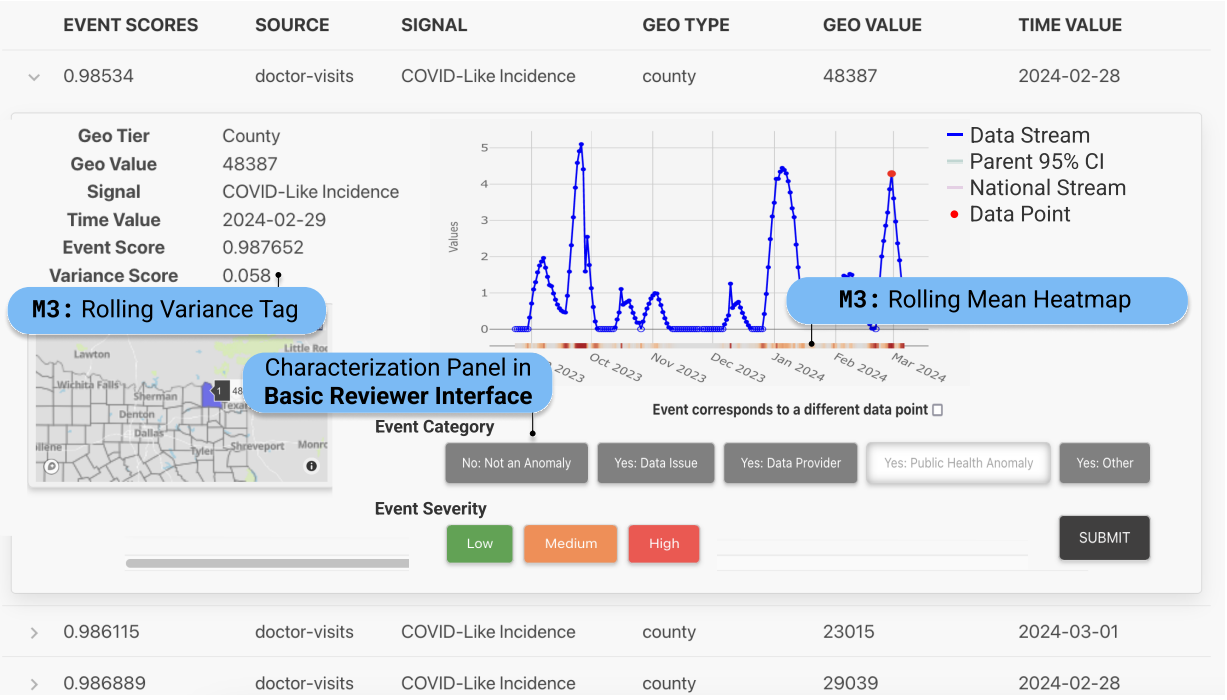


Figure 4.5: The **basic** review interface focuses on segmenting public health data by prioritizing the temporal dimensions. For mechanism **M3**, we added a tag with the rolling variance and a 1D heatmap of the rolling mean OutsHiNes scores so that reviewers have an intuition for how event severity changes over data revisions and time.

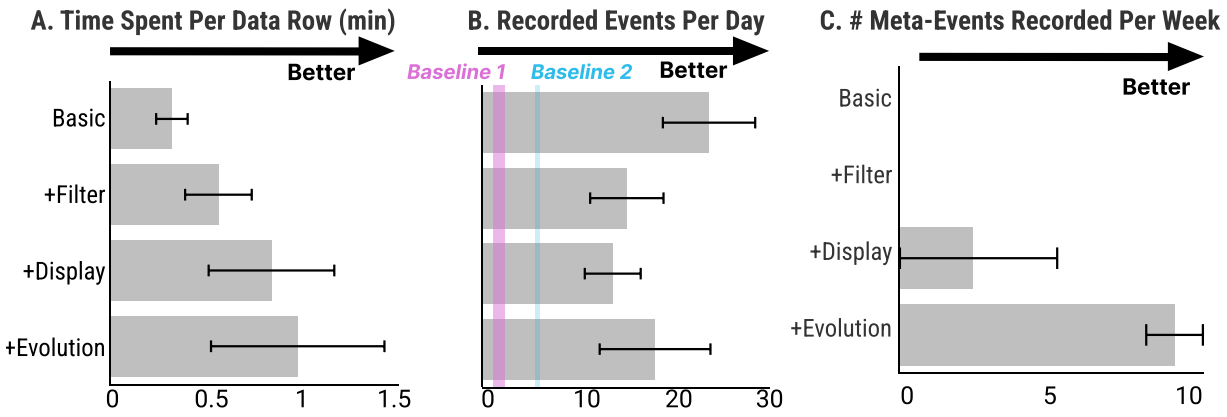


Figure 4.6: **A.** Reviewer engagement, displayed here with 95% CI bars, increased with each added modification. Baselines had no comparable metrics. **B.** Reviewers also recorded significantly more events on average, with some natural variation, while using the triaging system than the prior baselines. **C.** Finally, more meta-events were identified after displays were added.

Evaluating Actionable Data Monitoring

Evaluating public health monitoring systems remains a major challenge in the biosurveillance literature [103], and while there have been pushes towards standardized evaluation [56], longitudinal studies and sequential UI modifications remain under-discussed, reflecting the limited interdisciplinary approaches to data reviewing historically. Longitudinal studies are especially important for this data streaming evaluation because the daily reviewing load and the number of daily events vary. Still, this type of system evaluation, in general, is rare [4]. Given this gap, we design a longitudinal, sequential evaluation with metrics corresponding to key performance indicators for data reviewers:

Data Reviewer KPIs:

- *Efficiency metrics:* (how quickly) time per row, number of events recorded per session
- *Efficacy metrics:* (how well) number of events that were later revised, number of *meta-events* recorded.
- *Output metrics:* filter use, % of times that the algorithm identified data point was not the source of the event, and the distribution of the reviewer characterizations.

These were preregistered on OSF before the evaluation began.

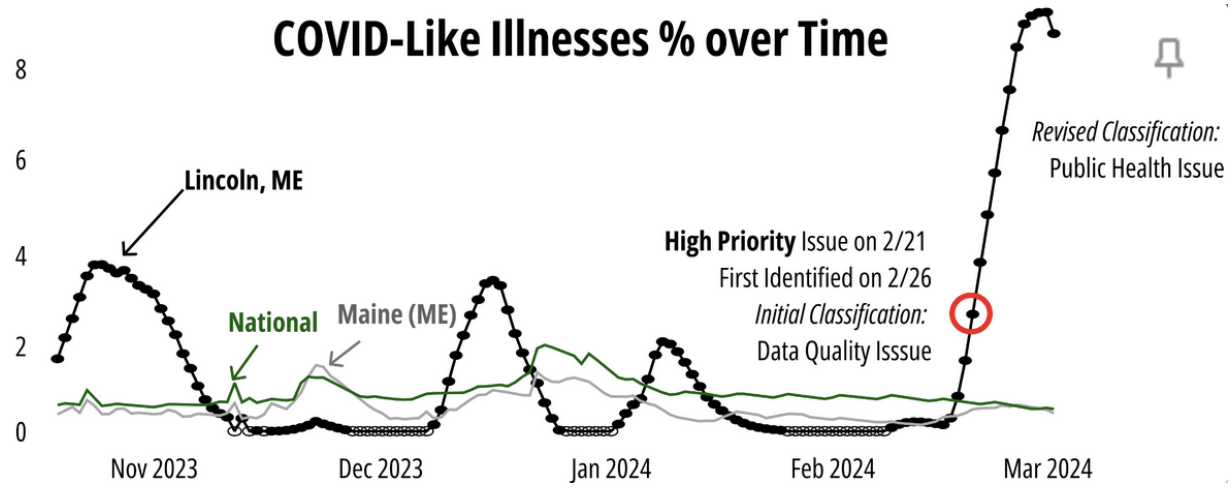


Figure 4.7: For a meta-event in Lincoln County Maine, there were 40 recorded events across 1 month and multiple signals, with a 10-day high severity warning before the peak.

To contextualize these numbers, we also include excerpts from a reflection that the data reviewers published in a blog. Each evaluation phase took the standard public health timeline of 3-4 weeks [13], and, for consistency we compared these metrics to the standard baselines in public health data monitoring as implemented at Delphi, as previously discussed:

(B1) *Exploratory Interface*: deployed for *120* weeks

(B2) *Temporal Segmentation Ablation*: deployed for *30* weeks.

Efficiency/Efficacy Metrics and Analysis

Event Identification Efficiency. Our efficiency metrics quantified a) how long reviewers interacted with each data row and b) the number of recorded events per day. As Fig. 4.6A. shows, reviewers generally spent more time per row after each modification, particularly our choice of filters and adding display panels for situational awareness. This suggests that these modifications allowed reviewers to analyze the data deeply:

> "[the system] allow[s] me to devote more of my time and efforts to assessing [events] of interest."

Reviewers also recorded far more events on average than with the prior baselines (Fig. 4.6B.); reviewers were **54x** faster on average than while using the exploratory system in Baseline 1, and **6x** faster than the Baseline 2 when recording events/minute. Finally, incorporating the charac-

terization panel directly into the basic reviewing interface dropped the average time reviewers spent on the system from 19.21 ± 0.41 minutes to 9.12 ± 2.7 minutes, suggesting that over half the time data reviewers previously spent on prior systems was recording their event characterizations. These metrics are contextualized by the reviewer:

> "[With the prior approaches], I was spending a good amount of time scrolling, manually sorting, documenting, and searching for specific [event] reports I wanted to examine rather than focusing solely on identifying, marking, and analyzing [events]."

Event Identification Efficacy. Reviewers identify high quality events using the triaging system. If reviewers make a mistake and wish to correct a recorded event, they can easily update the record. In the past, this was frequently used as there are multiple informative external sources of outbreaks that reviewers contextualize against. While historically, this led to edits (Baseline 2 's responses had at least 3 edits across a similar experimentation timeline), there were no edits during the duration of this experiment. More importantly, reviewers identified meta-events when they could investigate patterns in the events that suggested higher-level phenomena. For example, a reviewer identified the following meta-event:

> "Several counties in Puerto Rico are repeatedly experiencing sudden upward trending, [respiratory illness] spikes, this month."

Given the data, reviewers likely identified these meta-events because they had high-level information about regions and signals with events from the situational awareness panels, which could give them more direction to investigate using filters, as shown in Fig. 4.6. No such meta-events were recorded for Baseline 1 and only 2 were recorded for Baseline 2. Reviewers also seem to have a positive experience with this UI and visualization, sharing:

> "the updated [triaging system] now enables me to [make meta-events] for exciting [events], trends and other issues of importance, and maintain these notes in an organized, searchable fashion within the platform."

In a quality assurance check, these meta events were re-analyzed and corresponded with notable events, as shown in Fig. 4.7.

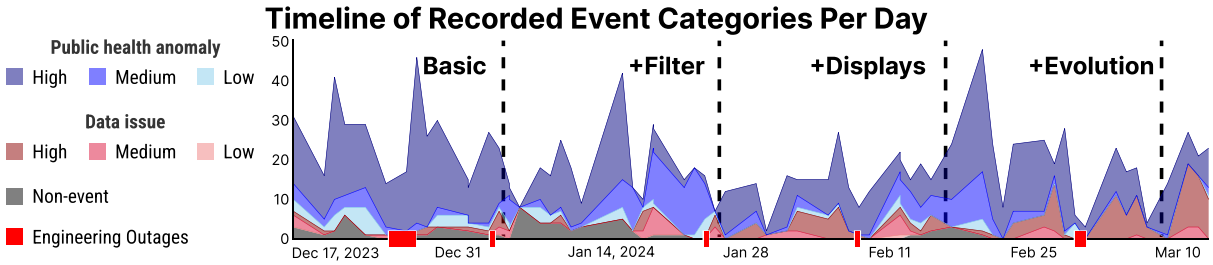


Figure 4.8: Recorded events per session (up to 49) are far greater than the 1-2 events reviewers would detect per session using Baseline 1. After filters were added, fewer rows were marked as a Non-event, suggesting that reviewers knew how to exclude data that was not interesting without complex synthesis strategies.

Output Metrics and Analysis

The resulting triaged events, as shown in Fig. 4.8, were a mix of data quality and public health issues. Before filters were added to support the segmentation decision, there were many more points marked as a non-event. However, after reviewers became more familiar with filters, they could exclude data that would generate high OutsHiNes scores but were not important or meaningful, like indicators that providers have stopped maintaining. Reviewers used filters on average 2.75 ± 0.43 times per day. Each filter can have up to 4 predicates (across signal, source, geo value, geo region), but reviewers only use an average of 1 predicate and only 1 value per predicate, once again supporting the desire for simple segmentation strategies. The most common filters only include specific geographic tiers and exclude particular providers.

Additionally, based on the variance of the number of events detected in Fig. 4.8, we note that online interfaces like the basic review interface are helpful because the number of rows reviewers will process (k) is *unknown and unknowable*. In Baseline 2, reviewers could only review the top 25 streams. This restriction prevented them from dynamically adapting to different reviewing needs when there were more or fewer data events - with up to 49 events were processed in a day in this experiment, and with filters over different single-predicate slices of data.

The insights from this evaluation target data reviewer KPIs across different dimensions: efficiency, efficacy, and output metrics. They demonstrate that this design, aimed to address core challenges with data review at scale, provide promising results.

Metrics Interpretations and Ground Truth:

Contextualizing the efficacy (or power) of triaged events was important to public health experts and statisticians [19]. Specifically, they requested guidelines on how to understand positives and negatives from triaged events *in practice*.

False Positives: From the reviewer's perspective, false positives occur when a data reviewer misclassifies an event. This may happen if a reviewer is biased towards the event ranking algorithm and doesn't thoroughly analyze or triage the data. However, such occurrences are unlikely since reviewers need to record events and, after a quality assurance check, there were no instances of a single recorded event being updated. From an algorithmic perspective, false positives happen when the event detection algorithm erroneously ranks non-event data highly or when the identified data point doesn't correspond to the event. Before our filtering, some false positives in rows were triaged as non-events by reviewers, but there have been only a few of these since then, as seen in Fig. 4.8. Additionally, about 14% of events evaluated by reviewers were due to data points near the one identified, but not exactly matching the identified data point. Thus, both the event detection algorithm and reviewer-in-the-loop approach were needed to identify the event correctly. This insight motivates the importance of incorporating the evolving relationship between reviewer expectations and event detection algorithm output as part of the approach.

False Negatives: From the reviewer's perspective, false negatives may occur when reviewers incorrectly classify data as Non-events, which is uncommon, or when unreviewed data contains events, which is likely common given the numerous events that occur in large-scale data. Still, reviewer capacity is limited, so not all data corresponding to events will be reviewed, and the accuracy of the presented ranking depends on the underlying event detection algorithm. Further, humans are known to anchor on ranking [26, 111], so a reviewer may stop the investigation on a particular day if there are several uninteresting rows. However, the thought-intensive analysis and triaging process [111] may anecdotally reduce this anchoring effect.

Thus, false positives, or events that were incorrectly triaged, are far less common than false negatives, which should be interpreted as events that do not get triaged. Stakeholders receiving the triaged events should note that while presented events are likely real events, there may be missing events that were not triaged. While public health experts emphasize reducing false negatives, "for outbreak and event detection, practitioners prioritize timeliness and sensitivity over positive predictive value [50]", doing so in a way that accounts for the human limitations of data reviewers is a core motivator of this work.

4.2 Anomalous Sequence Detection (Enlighten)

The objective for the final method, Enlighten, was to detect anomalous subsequences in streams. This approach is informed by two recent insights from the time-series anomaly detection/classification community. First, design-centered methodology supports practical generalizability [121]. While traditional anomaly detection methods are often evaluated using standard benchmarks and metrics, optimizing for benchmark metrics often doesn't lead to effective practical outcomes [52]. Instead, researchers advocate for methods grounded in realistic data assumptions. These methods are better suited for deployment and evaluation in real-world settings as they are robust to evolving data settings. Second, my approach incorporates human interaction into the monitoring process as the monitoring task is inherently human-centered, which aligns with the growing trend in the literature that integrates human behavior modeling into method development for informed decision-making [32, 114, 122].

Using the setup and intuition behind OutsHiNes, this method generalizes the approach from outliers to anomalous subsequences as follows. Instead of $\phi(d_{i,r}(t))$, where each test statistic corresponds to a single data point, we calculate test statistics across different *windows*, or the number of dats in a data subsequence.

Step 1: Forecasting and Test Statistics Over Windows For Enlighten, there need to be methods that forecast and create resulting test statistics across *windows*. One simple example is taking the prediction as the linear interpolation between the endpoints of a window and a simple

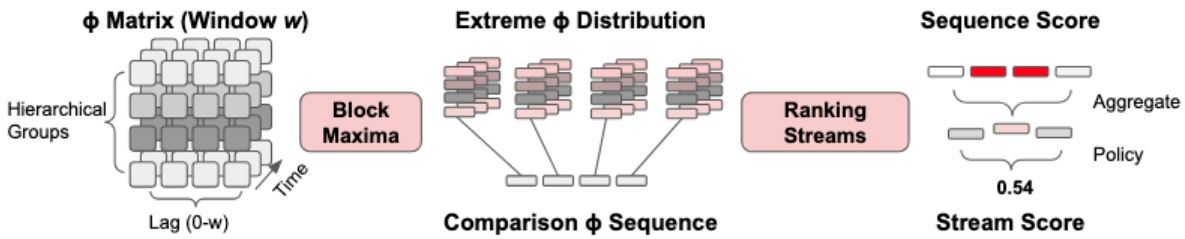


Figure 4.9: Generalization Approach Behind Enlighten For Ranking Streams with Anomalous Sequences

difference between the predicted and observed values. Thus, the test statistics per stream per window forms a 2D matrix, as seen in the first panel of Fig 4.9.

Step 2: Calculating $\mathcal{P}_{i,w,t}$ to contextualize $\phi_{i,t,w}(d_t)$ The input for stream $d_{0:T}$ are in the form:

$$\{\{\phi_{t,w}(d_t) \,\forall\, t \in range(0,T) \,\forall\, w \in range(0,win)\} : \forall\, window \in windows\}$$

For ϕ with different w in $\phi_{t,w}(d_t)$, they come with different associated variance. For example, the w values of 0 and win are close to the known point over which the linear interpolation is calculated, but values in the middle of the window are more likely to deviate from the observed values given that they are multiple forecasted points away from the endpoints. Thus, each element of $\phi_{t,w}(d_t)$ is compared to the same w, using each $\mathcal{P}_{i,w,t}$, calculated per window index using the procedure detailed in OutsHiNes.

Step 3: Aggregating Resulting Scores Over Windows

After using $\mathcal{P}_{i,w,t}$ to evaluate $\phi_{t,w}(d_t)$, per sequence $d_{t:t+window}$ will have scores $y_{t:t+window}$, as contextualized by $\mathcal{P}_{i,w,t} w \in window$. At this point, these scores must be aggregated into a single score. We use the mean as the straightforward interpretation is that the anomalousness of the sequence is the average of (the anomalousness of each data point, conditioned on its position in the sequence), represented by y.

Step 4: Creating a Stream Aggregation Policy

Ranking the subsequences of each data stream across multiple windows results in a combinatorial explosion of ranked sequences. Loading these sequences into the UI described in 4.1 using an API led to long delays, and reviewers would still spend time going over similar sequences in the same data stream across different rows.

Instead, creating a policy that selects 1) specific streams and 2) their top-ranking anomalous subsequences reduces pressure on the application mechanics and can improve the efficiency and analysis capabilities of reviewers who are now seeing all their anomalous subsequences in one place.

Results and Evaluation

We perform a number of evaluations to determine the utility of these steps and their processes. As this is the final iteration of the data monitoring system, we provide complete runtime specifications.

Statistical Parameters and Performance

During the testing and deployment of Enlighten, we used the following parameters:

1. Forecasting Methods:
   (a) Simple Forecaster
   $$(x_{t+1} = avg(x_t))$$
   (b) Average of past 7 and future 7 days
   $$x_{t+1} = average(x_{t-7,t}, x_{t,t+7})$$
   (c) Barycentric Interpolation of past 7 and future 7 days
   $$x_{t+1} = interp(x_{t-7,t})$$

2. Test Statistics

   (a)

   $$|Expected - Observed|$$

   (b)

   $$|Expected - Observed| * stream.std()$$

3. Stream Save Policy:

   (a) Policy 1: Maximum value across a stream's subsequences

   (b) Policy 2: Average value across a stream's subsequences,

In practice, to select streams, using either policy, we consider all stream with a policy score $>$ 0.95, and then take the top 1000 streams (if there are that many). For situational awareness, we also add in all states. Then, any subsequences with a score $>$ 0.95 are displayed.

Auxiliary Testing on Baseline Forecasing Method

First, we tested if there was a statistical difference between the choices for forecasting method and test-statistics. Each of the forecasting methods was evaluated for accuracy on data subset from as a judge of their relative forecasting power, as shown in Fig 4.10.

While the first two forecasting methods are fairly similar, we expect method 2 to have lower errors in general as it benefits from the future data point (not just the past). While expect method 3 to perform the best as it tries to fit a line of best fit through points in the window, given the noisiness of the data, this method may not scale in public health data. We observed this problem when testing the data. While RMSE may not be the best way to evaluate forecasting methods for computational epidemiology [95], these results in Fig. 4.10 show that relative performance clearly varies, especially as window sizes increase.

Then, the ranking outputs of each combination of forecasting method and test statistic (ϕ generation process) were compared for similarities. Using the Kendall Tau metric for ranking, we compared list similarities across the 6 combinations of forecasting methods and test statistics. In the implemented Kendall Tau metric, all rows are weighted equally in importance, so rows that

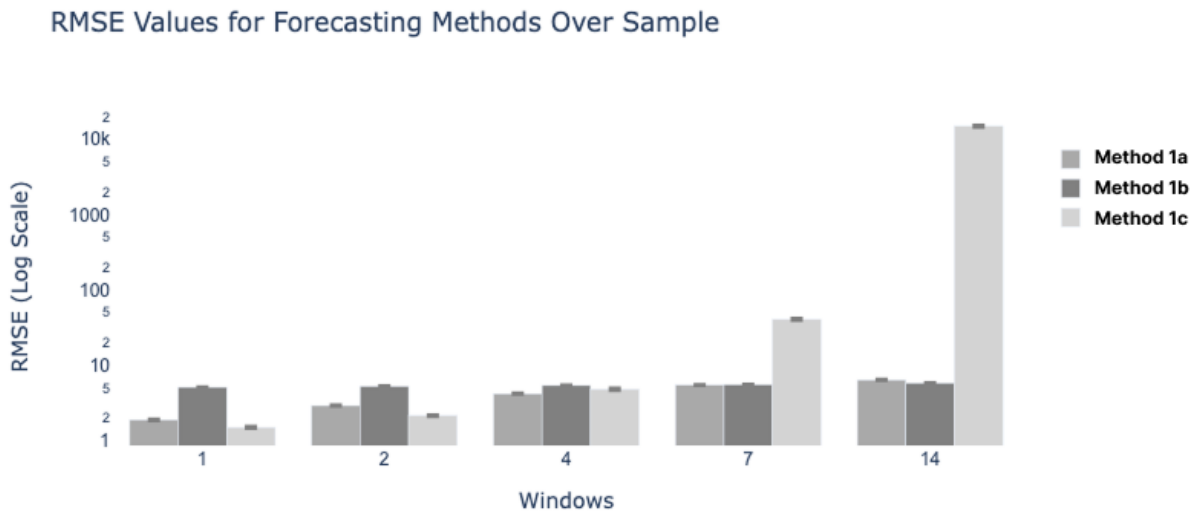


Figure 4.10: The RMSE of different forecasting methods (detailed earlier) varies widely across different windows.

are not matching regardless of their position are considered differently. We evaluate the scores in Fig 4.11 across the whole list, outputs using policy 1, outputs using policy 2 (top row), and different windows that reflect the results from Fig 4.10 (bottom row).

The results support the observations that the resulting ranked lists are different if the baseline forescasting and difference metric that generate ϕ are different. In particular, there is generally high similarity across prior forecast (1a) and simple interpolation methods (1b) , and a negative Kendall-Tau score for complex interpolation (1c). Given this observation, a data scientist or methodologist is well suited for identifying which combination of forecasting method and test statistics is best suited for this application. They may, for example, consider other factors, like the runtime [3] , shown in Table 4.1 over the span of 3 weeks of data.

Next, there was a check on if the top ranked anomalous sequences were different from their constituent outlier detection scores. If not, we could just display all the point outliers instead, or use approximations like multiplying the resulting test statistic scores as a heuristic (given that the scores for consecutive data points are not independent) as follows:

[3]Runtime scales linearly with the number of windows considered. Historically, we evaluate windows [1, 2, 4, 7, 14], because these are typical timeliness during which there are changes to public health data.

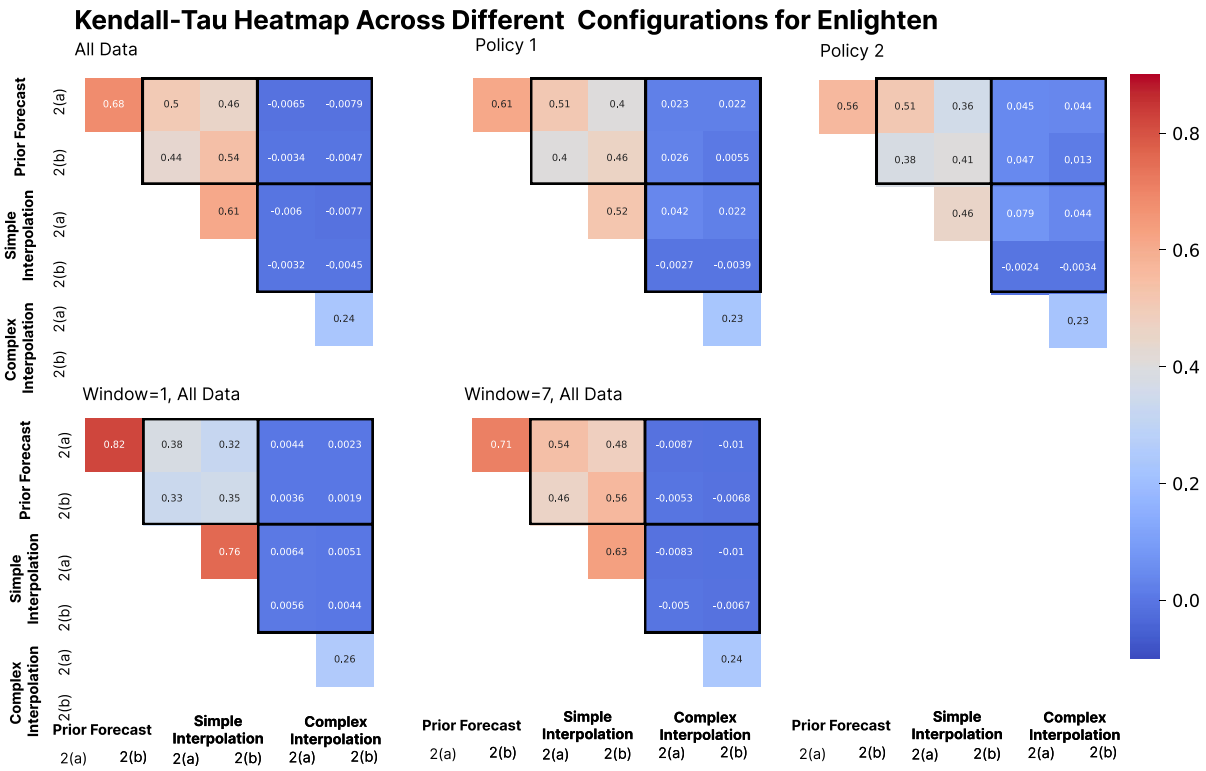


Figure 4.11: Mean Kendall-Tau comparisons across 3 weeks and different configurations measure how different the ranked lists are across forecasting methods, difference methods, policies, and windows.

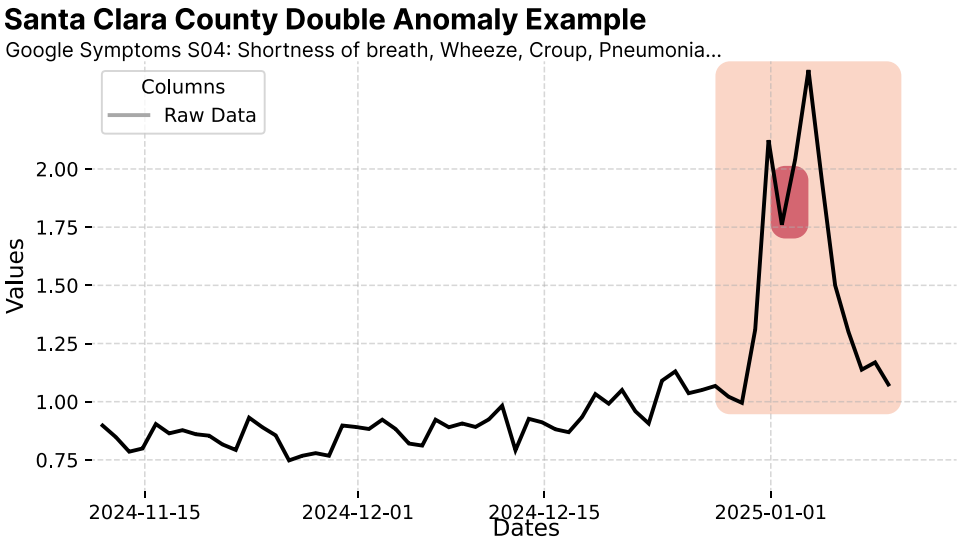


Figure 4.12: Multiple anomalous subsequences across different windows are present in the data. These are frequent enough that they should be detected and handled separately and should not be handled by segmentation-based anomalous subsequence detection methods.

Timing Univariate ϕ Methods (s)

Forecasting Method	Difference Method	
	2(a)	2(b)
1(a)	140.48 ± 1.85	143.2 ± 1.68
1(b)	146.46 ± 1.8	147.35 ± 1.7
1(c)	178.6 ± 2.29	181.81 ± 2.96

Table 4.1: Timing Across Different Combinations of Test Statistic Calculations using the 'Quidel' source that was updating at the time of evaluation.

Null Hypothesis (H_0): The mean of the outlier scores within a window is equal to the anomalousness score of that window.

Alternative Hypothesis (H_1): The mean of the outlier scores within a window is not equal to the anomalousness score of that window.

However, in examples like in Fig 4.12, there are crucial instances where there are multiple types of anomalous subsequences squashed together. Across all 6 configurations and 3 weeks of data, using a paired t-test across the the mean of outlier scores within a window and the anomalousness score of the window is 0 with a std dev of 0. This is unsurprising considering some of the extreme example differences in Fig 4.13 below.

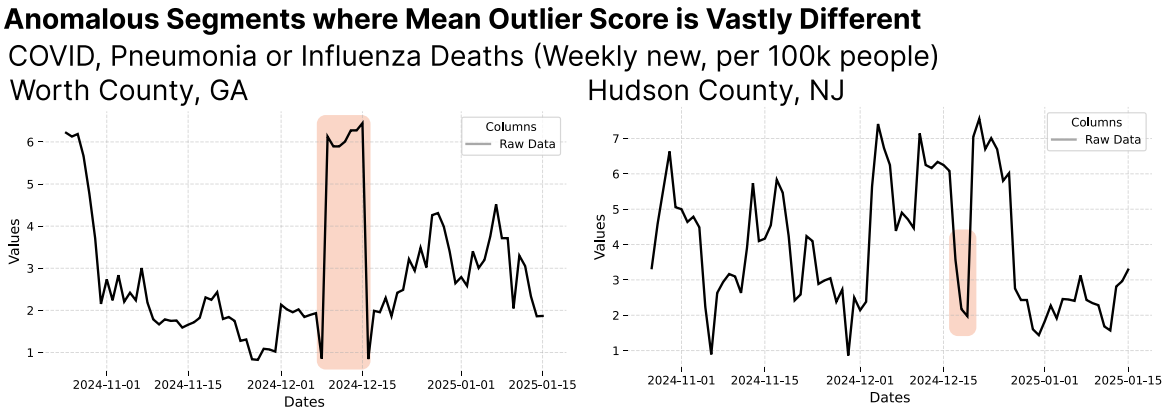


Figure 4.13: Examples where anomalous sequence scores are vastly different than mean outlier scores of the sequence. These are most evident when, if the sequence was removed, there would be a clear straight line or expected pattern.

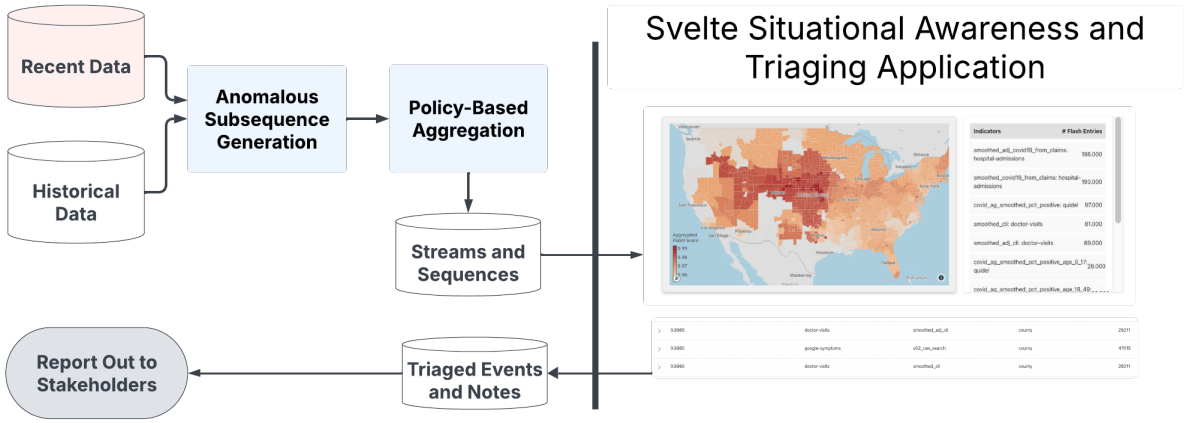


Figure 4.14: Simplified achitecture schema for monitoring system combines the methods, engineering, and reviewer perspectives.

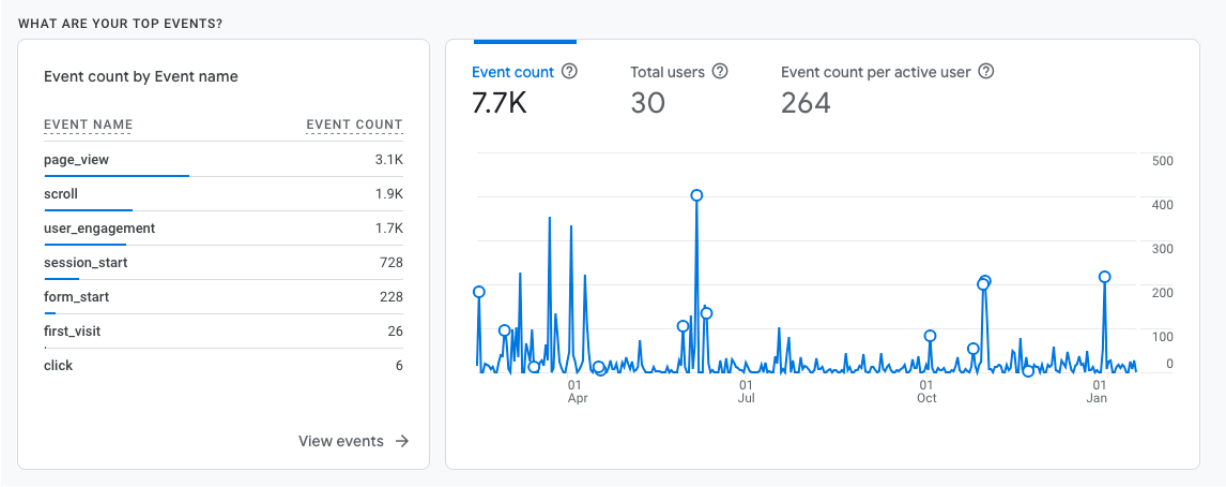


Figure 4.15: Web events recorded by Google Analytics since Feb 1 2024.

Engineering & Performance:

The architecture for the monitoring process in Fig 4.14 now runs on Delphi's production systems. The backend processes are maintained by Delphi's backend systems using Cronicle. The performance of the front-end is displayed in Fig 4.15.

1. **Google Chrome's Lighthouse Analytics:** The overall performance of this metric (an aggregate of desirable website properties) is 87 %. Notably, most of the website loads within 2.2 seconds, which is far less than it was historically because of the size of the API call enabled by the policy-based structure. Often on a new reload, the upper 'situational aware-

ness panels' require a page reload so that the data is cached. It typically takes between 5-6 seconds to load the entire page. Each row takes about 3 seconds to open and transfers about 0.1 MB of data.

2. **Google Page Analytics:** since being deployed over the past year, we've been able to track changes over time. See Fig. 4.15 for the dashboard values.

On the backend, the overall algorithm runtime is a function of the data processed and the windows considered. Of note, while processing a longer window might take a univariate algorithm longer, there are also fewer sequences to process. The deployed timing across different sources, considering 60 days of history using Delphi's infrastructure generally ranges from 3-10 minutes/source.

Survey Performance:

We administered a survey to evaluate the performance of the anomalous subsequence detection method vs. other anomalous subsequence detection methods and across Enlighten output quantiles. The comparison anomalous segmentation algorithms are implemented by the TSB-UAD package and were based on an Autoencoder method, Matrix Profile [125], PCA, and LSTM. This combination of deep learning, dimension reduction, and recent anomaly detection methods provides coverage across notable classes of anomaly detection algorithms.

In the survey, there are 4 HTML files , each containing 5 data streams. Streams were selected based on the quantile of the algorithmic output for the first HTML file and the top 3 ranked data streams for the other files. One stream was repeated to test internal consistency per reviewer. Streams are then ranked in a Google Form. For each category, the first question on the Google form asks users to name the file they are using. Results were excluded if the answer to this question does not match the question for that section in the Google form (e.g., Respondent writes they were looking at form A in the form B section) or if there is any evidence of technical difficulties (e.g., required questions are unanswered).

There were a total of n=21 survey respondents, with *n=19* respondents internally consistent across the survey. To analyze the consistency between results, we consider the concordance between user values (0.57) and if users ranked the same survey questions the same value (0.85

Algorithm	**Autoencoder**	**Matrix Profile (DR)**	**PCA (DR)**	**LSTM**	**Enlighten**
# Ties	136	**2**	129	**1**	**1**
Timing (s)	6602.39	**31.3**	**9.5**	6885.19	**339.6**

Table 4.2: Metrics comparing methods

	Accuracy	F1	ROCAUC	Distance	Correlation
Outshines Quantile Survey	0.81 ± 0.11	0.82 ± 0.11	0.78 ± 0.12	0.73 ± 0.08	0.56 ± 0.12

Table 4.3: Survey results from Outshines quantile rankings.

$\pm$ 0.15). The lower concordance score suggests there was some variance in how different people evaluated different segments – with some users purely focusing on data quality issues, others on public health issues, and most on both.

Like in the OutsHiNes evaluation, the following metrics were considered:

1. Binary: To identify positive class points, we select the top-k streams to calculate the Accuracy, F1 Score, and ROCAUC.
2. Ranking: We calculate correlation and distance.

As shown in Fig 4.16, the results across the top-ranked sequences from different algorithms favor the Enlighten ranking approach across various metrics. This evaluation is reported as it was preregistered. However, the source of the high variance is that multiple users disagreed on the anomalousness of *missing* data sequences. It is likely based on the annotations and descriptions from survey respondents that those who found missing data anomalous were less familar with the recent data from Delphi at that data, which generally had notable data outages across all streams. Nevertheless, on the ranking metrics like distance and correlation, the performance boost of the Enlighten method is clearer. Adjusting the threshold over which the binary metrics were calculated could better reflect the uncertainty around missing data and would improve the performance of the Enlighten method over the presented alternatives.

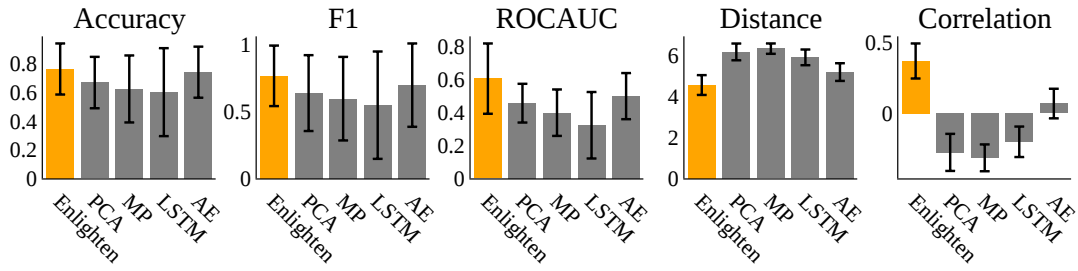


Figure 4.16: Results reflect that while the Enlighten method performs the best on average, the variance built in from people with different opinions about missing anomalous sequences is more than for single outlier points like in the OutsHiNes evaluation. Still, the correlation metric is telling that the other approaches are not providing relevant rankings.

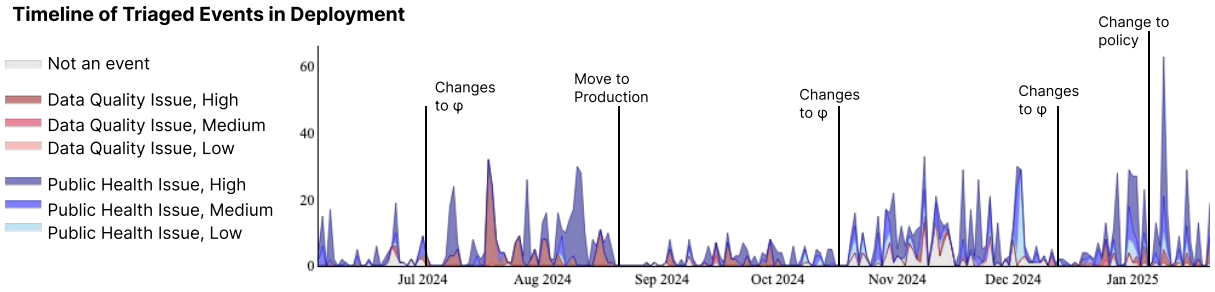


Figure 4.17: System-performance timeline for deployed results and anomalous segments triaged.

Deployed Performance

Versions of Enlighten have been deployed since June 2025, where numerous changes to the display policies have been implemented. In Fig. 4.17, we show the performance over these changes.

The story that this data tells us highlights different aspects of the overall monitoring system. In early June 2025, the Enlighten method was deployed on Delphi's production data. Over the following month, it became clear that the underlying test statistic ϕ wasn't as informative because it was calibrated for detecting changes in diseases dynamics, which are less common during the summer, instead of data quality issues, which are more common. As the grey 'not an event' triaged data in Fig. 4.17 increased and the number of meaningful triaged data decreased, we updated ϕ to match the summer data dynamics. The success of that change is in the second region

where more data quality issues were detected. Around September, the system was handed off to the Delphi engineering team, which started it on a smaller volume of data. The number of triaged data points adapted accordingly so that fewer points were reviewed. Following an engineering change, the algorithm stopped updating daily, leading to an interesting natural experiment where reviewers would go down the list of events, giving us some notion of recall in a deployed setting. As expected, despite processing many data streams, the frequency of triaged data points dipped until the system was fully functional again in early December. Finally, given the start of the respiratory illness season, ϕ was modified to reflect highlighting events that likely corresponded to notable outbreaks. The final quantitative performance metrics are summarized in Table. 4.4.

Metric	Enlighten	Closest Baseline (Outshines)
Time Spent/Day	12.9 ± 8.2	16.4 ± 5.3
Rows/Min	1.69 ± 0.74	1.509 ± 0.480
Rows/Day	9.08 ± 0.01	19.4 ± 6.3
Points per Triaged Anomaly	7.25 ± 1.65	1
# Streams Shown	357 ± 161	$3500000 \pm 280k$
Meta Events/Week	17.1 ± 7.79	10 ± 3.35

Table 4.4: Survey results from Outshines quantile rankings during the experimental study.

These results show that Enlighten enabled 1.7x more meta events detected than Outshines alone, 2.17x increased efficiency via events/day, and an overall 288x efficiency in data points triaged than manual baseline. To add qualitative context to these quantitative metrics, Tina Townes, a system user, wrote this reflection:

”Compared to using the early 2024 version of the FlaSH platform, the latest iteration of this platform that I have been using in the past few months contains several enhancements that make it more efficient for me to evaluate various anomalous subsequences.

First, this new version has an updated, more fully and consistently populated general overview map at the top of the page. This improved map immediately highlights for me the locations with anomalous subsequences in an easily viewable bright dark red color. With this map, I can more quickly scan this updated map and see the severity of the anomalous subsequences, with the

most severe locations highlighted in bright red as well as less severe locations with decreased red color gradients. The overview map in the previous FlaSH version was often missing information and was not consistently populated. For example, recent overview maps have consistently been indicating a noticeable concentration of deep red coloration in the midwest and, slightly less so, in the southwestern areas of the United States. Seeing this upon opening the daily FlaSH report immediately helps me remember these highlighted locations and helps me recall, on a larger scale, anomaly trends in locations on a weekly, and even monthly, basis.

To the right of the overview map is a chart that is now also more fully and consistently populated with a list of signals in order from highest to lowest number of FlaSH entries. In the prior iteration of FlaSH, this chart was frequently poorly properly populated, or not showing information at all, making it difficult for me to set my expectations for what sorts of signals and the frequency of occurrence of these signals I should see in the current day's FlaSH report. In setting my expectations, I can spend less time in the day's FlaSH report in starting from scratch to individually identify a signal's geographic location and keeping track of the frequency of its occurrence. With this improved chart, I can now quickly identify the signals with the most anomalous subsequences and devote more effort on evaluating the actual severity of the reported anomalous subsequences and noticing larger trends in anomalous subsequences being detected.

Another improvement in this latest iteration of the FlaSH platform is the new inclusion of anomaly indicators in the form of red vertical bar highlights on the graphs of each individual FlaSH anomaly. These red highlights draw the eye quickly to points on the days, past through present, when severe anomalous subsequences occurred or are occurring. The red bars highlight both dramatic peaks in anomalous subsequences, where within a matter of a few days anomalous subsequences have risen by leaps and bounds, as well as sudden dips and valleys where high points fall quickly within a few days. These indicator bars are new, and occasionally a noticeably high or low data point is not highlighted as expected. Conveniently each daily report graph has an "Anomaly Accuracy" feature that allows me to select dates on the graph during which I think anomalous subsequences occurred but the red bar indicators did not highlight. This new feature helps me zero in on actual anomalous subsequences and saves me from having to take time to visually scan, digest and examine an entire graph to locate anomalous subsequences from scratch.

Finally, having a note-taking feature easily accessible next to each signal's graph makes my anomaly review process more efficient and enjoyable. Rather than having to move my eyes onto a separate tab or window to fill in a separate notes form, as I had to in the prior version of FlaSH, I can now quickly take notes in the box provided while still looking at the signal's graph and my anomaly severity selection."

4.3 Auxiliary Evaluations

In evaluating the thesis methods, we typically focused on performance-oriented aspects of the scoring process. For example, in the FlaSH experiments, random streams were selected for evaluation, where the sample size was augmented by the length of these streams. Then for OutsHiNes and Enlighted, we focused on the top-k sequences. To get a sense of statistcal false negatives and recall in addition to precision, we conducted the following auxilliary experiments:

1. Evaluation at Top-k: In the OutsHiNes and Enlighten experiments (survey and deployed), the focus of the experiment was on measuring the precision of the methods at top-k events or streams, where k changes based on the reviewer's capacity.

2. Evaluation at Top-nk for n=2,3,4...: Natural experiments like in Fig. 4.17 demonstrate, in practice, that the number of events triaged goes down through the ranked list from a single point in time.

To study recall intentionally, we conducted a small-scale evaluation by sampling random events conditioned on their Enlighten scores and asking reviewers to classify the points as potential events (0/1). This sample was pulled from the 'doctors-visits' source and using 300 score-based quintiles from a total of 76,338 subsequences in the last 7 days with windows 1, 2, 4. Reviewers inspected the resulting 247 rows (some quantiles had no rows), and the results from our most experienced reviewers showed that the only sequence selected as the event was also the top-ranked sequence, which is *perfect precision and recall*.

In another evaluation, a sample of 247 rows were evaluated as subsequences that would somewhat warrant suspicion based on a larger group of reviewers (n=3), where if at least one reviewer thought the point merited inspection, it was included as a positive event label. The following ranks were with the positively labeled events from any of n=3 reviewers (1 experienced, 2 inexperienced) : 1-10, 15, 21, 33, 37, 51, 73, and 191. This supports what was observed experimentally that the number of events detected decreases down the list of scored points.

3. Evaluation across all data: Although the prior evaluations focus on the meaningfulness

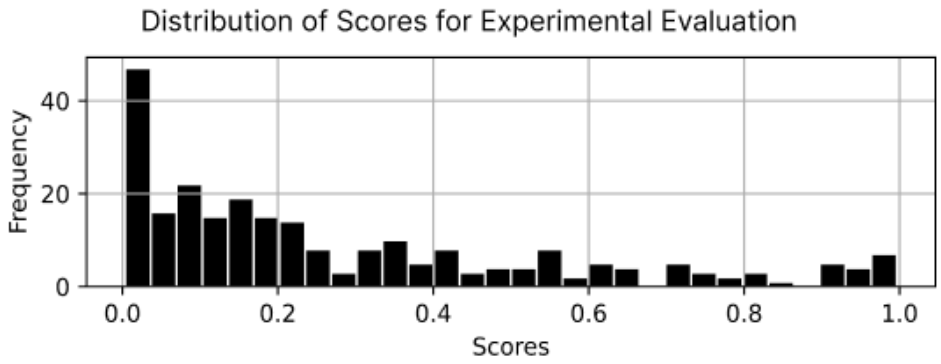


Figure 4.18: Distribution of scores in the random sampling over max events per stream to obtain information on recall across the stream rankings.

of ranked data based on score, the design of these scores means that a considerable number of sequences have a score of 0 (which means that they were not more extreme than any of the historically observed differences). In a final experiment, 250 random samples of streams output by Enlighten using a maximum value policy from 'doctors-visits' streams in the last 14 days with windows 1, 2, 4, 7, and 14, as they would appear, in order, were reviewed. Unlike the top-nk setup, this random sample meant that the underlying scores are skewed, with many of them being near or tied at 0 (see Fig. 4.18). This final experiment gives us information on the performance of Enlighten across all data and not just the non-zero scored data.

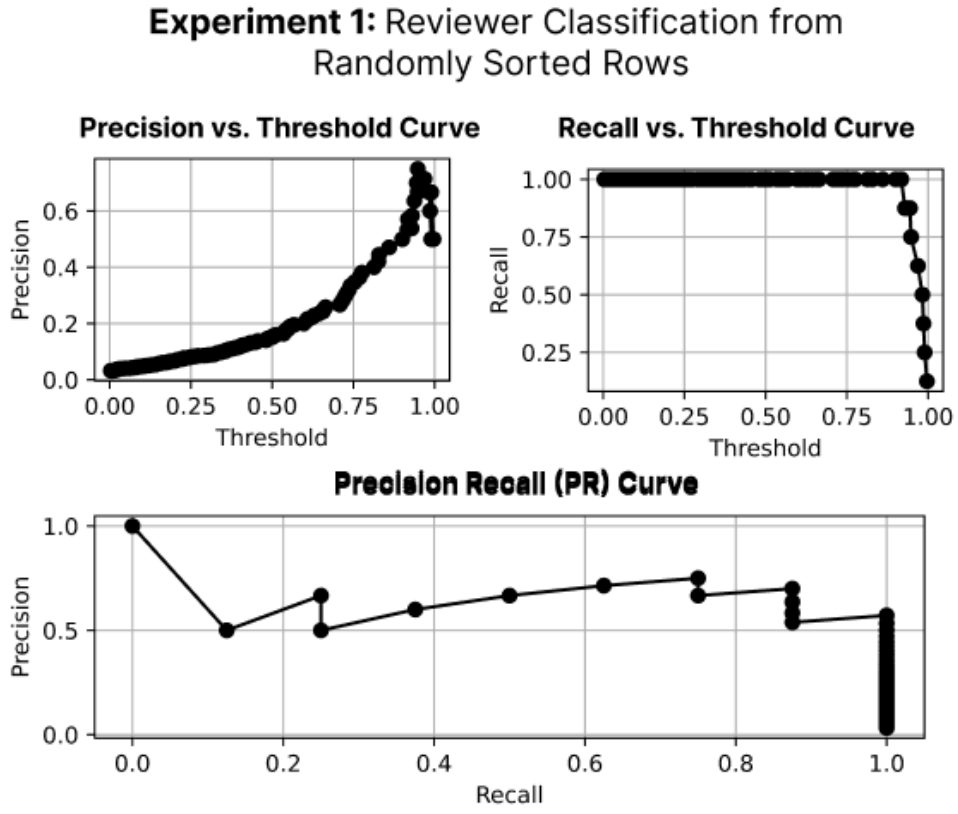


Figure 4.19: Precision and recall graph based on ground truth labels over randomly sampled streams (based on the maximum event score policy used in practice).

Like in the evaluation at Top-nk, events were randomized and reviewers needed to identify which events were interesting, with the precision and recall values shown in Fig 4.19. While some highly ranked rows were not classified as interesting, lower ranked rows were not selected.

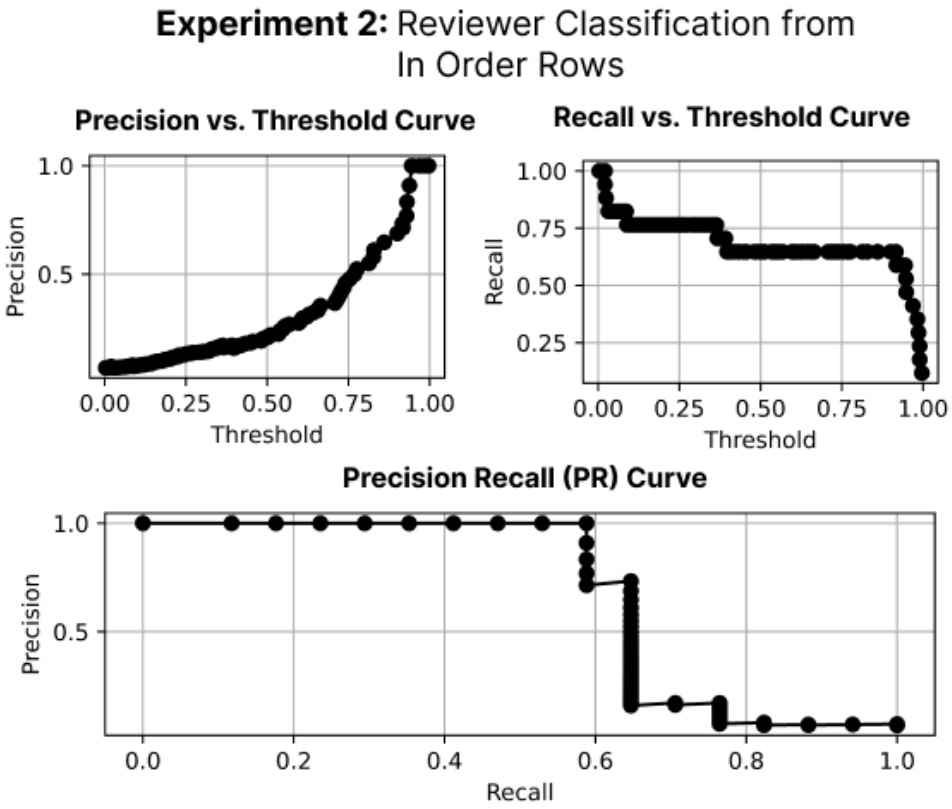


Figure 4.20: Precision and recall graph based on ground truth labels over in order reviewed streams with any possible event serverity (High/Medium/Low) for completeness.

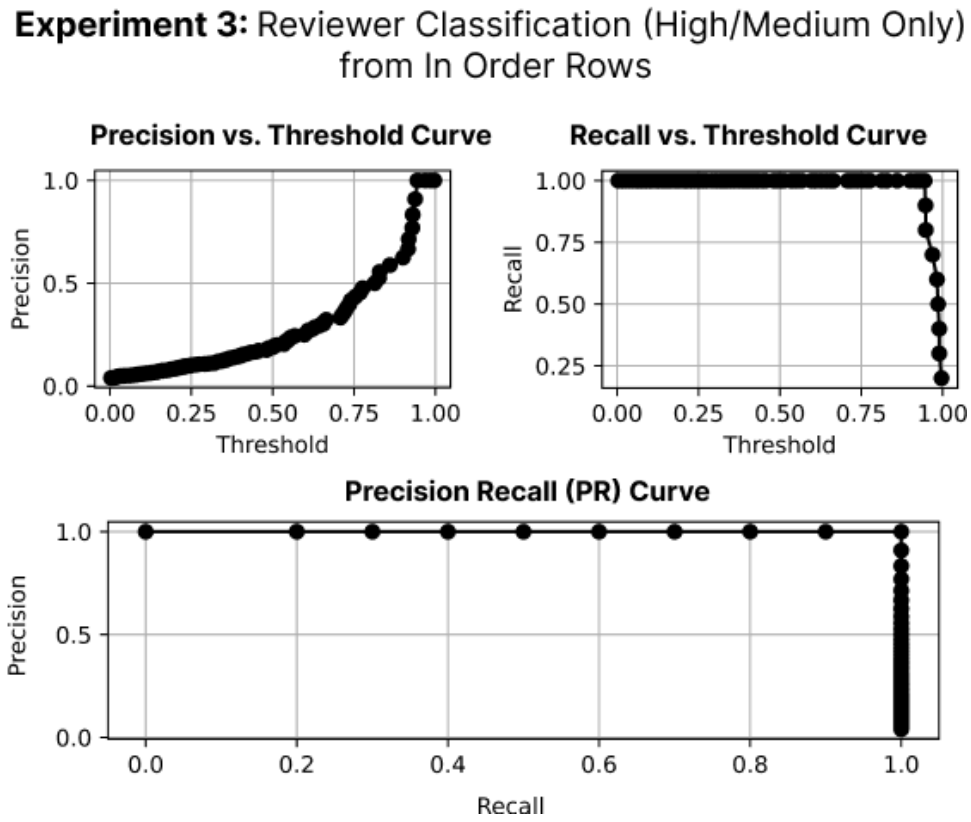


Figure 4.21: Precision and recall graph based on ground truth labels over in order reviewed streams with only the events the reviewer would likely only consider in practice (Medium/High).

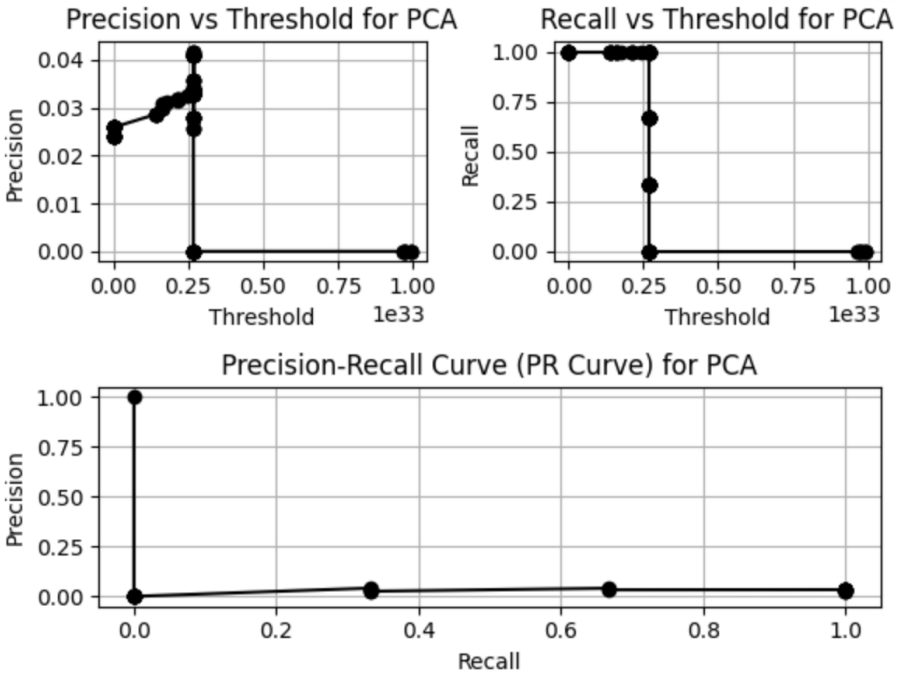


Figure 4.22: Precision and recall graphs for PCA-based **baseline** demonstrate the improvement the *Enlighten* approach provides.

Still, based on phenomena observed during FlaSH, to focus reviewer attention on lower ranked points, the reviewer was told to review the events in order and explicitly compare lower ranked streams to higher ranked streams. While their top classifications for 'high' and 'medium' concern triaged data aligned with top ranking data, there was some interesting variance in 'low' ranking data, where some points that were ranked in the 200's were more interesting than # 16 (relevant metrics shown in Fig 4.20). Nevertheless, reviewers shared they would not have kept reviewing beyond 'medium' concern triaged data if presented to them in a ranked list daily. Keeping only the highly triaged events results in the metrics shown in Fig 4.21. These curves show far better performance than baselines like PCA (Fig 4.22).

The results from this final evaluation once again supported that that there were few positive events beyond the top-ranked data, supporting that the method not only generates a useful prioritization for practical use, but also likely does not deprioritize important events.

4.4 Thesis Conclusion

This thesis presented novel monitoring algorithms and demonstrated their effectiveness as part of a human-in-the-loop monitoring approach for large volumes of heterogeneous modern public health streams. Constraints of the monitoring setting, including the noisy and challenging statistical properties of public health data, human attention, and engineering responsiveness, impacted the design of the FlaSH, OutsHiNes, and Enlighten methods. The resulting system meets our initial design goals of supportive informative detection, reducing the overwhelming alerts phenomena, and identifying anomalous subsequences in service of situational awareness. The resulting evaluations were comprehensive, encompassing engineering metrics like runtime and compute, statistical measurements, like ties, offline surveys on binary and ranking metrics, and, most importantly, sustained deployment in practice. This approach has met the initial thesis goals across multiple types of correctness and feasibility evaluations and has now been handed off to the data reviewers and engineers in Delphi to maintain going forward.

Overview of Miscellaneous Projects

Outside of monitoring, I worked on several tools for public health organizations [95, 96, 96, 97]. These projects required collaborations with domain experts to deliver actionable intelligence.

1. **Cases2Beds**: Developed a model to predict the anticipated number of hospital beds required based on COVID-19 case rates for the Allegheny County Public Health Department (ACHD).
2. **Identifying Gaps in Claims Data**: Investigated and highlighted intricacies and deficiencies in public health data streams derived from claims data.
3. **Leading Indicators using Changepoint Detection**: am advising Tara Lakadwala in quantifying relationships between public health-related indicators.

These experiences contributed to my perspective and qualifications in developing the core thesis contribution in data monitoring for critical settings.

5.1 Cases2Beds:

Adapted from the CSD Blog Open Source Code

In early November 2020, as the case rates in Allegheny County continued to increase, the Allegheny County Health Department was worried that the county's hospitals would run out of hospital beds for COVID-19 patients. They needed at least a week to open emergency COVID facilities but did not want to deploy already stretched-out resources if they wouldn't be used. To provide them with county-level intelligence on hospital bed usage 1-2 weeks in advance, we developed the Cases2Beds model.

The model used publicly available and ACHD's line-level data to estimate:

1. the probability that a person who tested positive for COVID-19 would require hospitalization
2. offset: the gap between testing and hospitalization
3. duration: the length of hospital stay
4. the current number of COVID infections

.

Across the United States, according to data at county and state public health deparments, these values vary across age groups and, to a lesser extent, sex and race. We wanted to use this data to perform Monte Carlo simulations, but because the model used probabilities derived from Protected Health Information (PHI), ACHD needed to run it privately and offline using Microsoft Excel, which is ill-suited for these large simulations.

Instead, we developed an analytical model lightweight enough to be used as part of an Excel macro, where some fraction of individuals who test positive today will be hospitalized after a varying offset and variable duration based on their age, sex, and race. These parameters are used to generate *Offset Fractions*, which is the probability that a patient with given traits will occupy a bed for a duration of k days after their COVID test. These Offset Fractions and the daily positive case breakdown give us the expected mean and variance up to 1 month in the future of the number of patients in the hospital per day based on the cases already seen. This information can be used to generate plots like (Fig. 5.1), which shows that based on the cases we know, only a few people will be hospitalized for more than a month.

- $O_{r,l}$: The offset value for a given subset of the population $r \in R$ where $R :=$ {race} x {gender} x {age group}for a given day l where $-10 \leq l \leq 30$. This probability distribution function (pdf) is derived from a piecewise function using segments of exponential distributions characterized by the offset parameters.
- $D_{r,k}$: The duration value for a given subset of the population $r \in R$ for a given day k where $0 \leq k \leq 40$. This pdf is derived from a piecewise function using segments of exponential distributions characterized by the duration parameters.

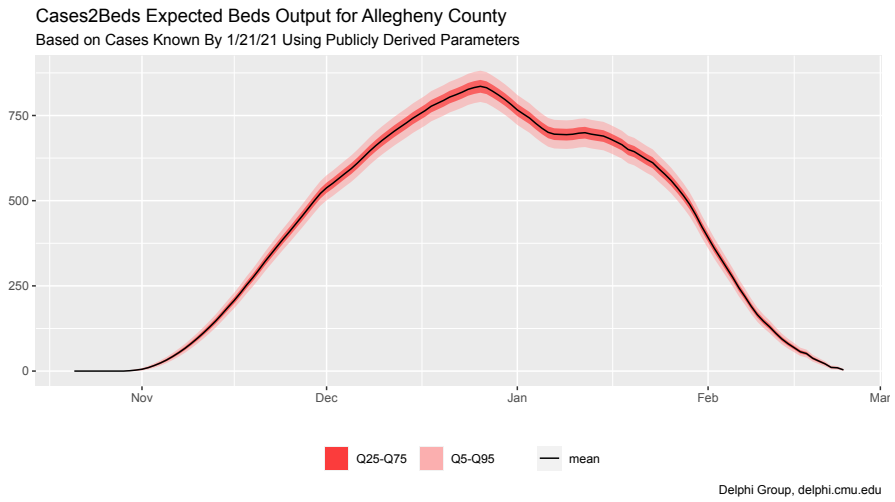


Figure 5.1: Output of Cases2Beds using historical data until January 21st 2021 for Allegheny County using public parameters

- h_r: The hospitalization rate for a given subset of the population $r \in R$ where $0 \leq h_r \leq 1$.
- $c_{r,d}$: The number of cases for a given subset of the population $r \in R$ on a particular COVID test date d (ex: 5 cases with a COVID test on January 1st, 2021).

Then,

$$OF_{r,j} = \sum_{l=-10}^{30} \sum_{k=0}^{40} \mathbb{I}(l \leq j \leq l+k) O_{r,l} * D_{r,k} * h_r$$

is the offset fraction for a given subset of the population $r \in R$ for a given delta j where $-10 \leq j \leq 30$, which is the probability a patient with given traits will occupy a bed on j days after the specimen testing date.

$$\mathbb{E}[\beta_i] = \sum_{d \in D} \sum_{r \in R} \sum_{j=-10}^{30} \mathbb{I}(d+j=i) OF_{r,j} * c_{r,d}$$

Then, $\mathbb{E}[\beta_i]$ is the expected number of beds on date i, where i can start 10 days before the first case date and can end 30 days after the last case date ($c_{r,d}$). If we assume independence between patients, the mean and variance calculations are exact. However, our quantile estimates are based on approximating the sum of independent binary variables, so the accuracy of the more extreme quantiles (95%+) depends on the number of cases present. *I used this insight when designing the test-statistic for FlaSH.*

Deployment By the end of November 2020, ACHD was using the Cases2Beds spreadsheet. Over the following months, we also introduced the spreadsheet to other health departments and hospitals by generating tailored, public parameters instead of relying on ACHD line-level data. Many of these organizations needed projections more than 2 weeks out, so we used Cases2Beds as an input to a hospital utilization forecasting model, which, in preliminary evaluations, had decent predictive power.

Lessons for Data Quality: Cases2Beds highlights the vulnerability of downstream models to shifts in data quality. Notably, changes in data completeness, as witnessed in the early stages of the pandemic, can markedly influence the forecasted hospitalization counts. Still, the Cases2Beds model is an example of a predictive model for hospitalization indicators that reviewers could input into the monitoring algorithms developed like Enlighten.

5.2 Identifying Gaps in Claims Data

My second project was to identify leading indicators in a claims data set from a data provider that covers around 50 % of the medical claims in the United States. Due to the limitations of the data provider, I will only discuss generic lessons about data quality issues from working with a claims dataset.

Lesson 1: Data Quality Issues Are Common, Counter-intuitive, and Numerous

Despite the richness of a large claims data source, this data was still incomplete (e.g., groups that receive medical treatment less frequently will be underrepresented), inaccurate (e.g., many manual reporting errors), and untimely (e.g., with delays up to 60 days). For some important indicators, the data quality issues overpowered the underlying signal, which led us to exclude these indicators from the Delphi repository. One example was a potential *vaccination indicator*, where data showed that the number of vaccination claims were 40x fewer than expected, given the market share of the claims. We also should have been able to use the gap between provider eligibility pings, which occur when a doctor's office checks for a patient's insurance (first when the appointment is made, then on the appointment day), to identify acute illnesses. But even

though there was an average of 2 pings per outpatient visit as expected, most pings were isolated, with a few outlier patients having hundreds of pings. While both of these indicators had rapidly changing (improving) data quality as the provider added more claims, those changes in quality were more prominent than any public health phenomena in the aggregated indicator streams. To help prevent this issue in investigating future indicators and domain experts understand the claims data, I created a patient stories module, which allowed engineers and data scientists to follow patient trajectories through and run automated exploratory check from the available claims history across multiple relevant tables.

Lesson 2: Claims Data is High Dimensional and Heterogeneous

Our subset of claims did not perfectly represent the United States; market share varied by state, patient's ages (e.g., 65+ usually processed by Medicare), and affiliation (e.g., veterans were excluded). Some patients also had data on their Social Determinants of Health, including race, education, and income, but these proportions were different from U.S. Census data.

Isolating meaningful indicators from this high health dimensional claims data is difficult due to the joint interactions between the data features. Consider the hundreds of streams constructed from all Social Determinants of Health combinations for a single ICD-10 code. The available streams from these weakly dependent inter-sectional streams across different geographies uncover public health insights that are invisible from the marginal stream. Because this data also has changing resolutions and availability, many streams at different levels of aggregation should be considered even though the correlations between these streams change drastically across different indicators over time.

I developed a context-sensitive method to generate multiplicative corrective factors on select features features to recover an estimate of the population values for data streams. But, because the available data is continuously changing, these corrective factors would have needed to be recalculated with every sub-daily update.

5.3 Changepoint Detection to Identify Leading Indicators

With Tara Lakadwala

Formalizing relationships between existing indicators using raw data values prove challenging due to the statistical properties of public health data streams. Our approach investigates how changepoints derived from different changepoint detection algorithms in revised public health data align with known shifts in the dominant COVID-19 variants from the CDC. We noticed that the efficacy of these methods had high geospatial variance and that the relationships between changepoints between public health-related indicators, even in traditional public health indicators like COVID-19 Cases and Deaths, changed dramatically given the phase of the variant wave. Overall, Changepoint detection is a powerful tool to identify early indicators. Of Delphi's sixty indicators, we identified several on time and early indicators of emerging variants from the data available. We also found out that for many of the indicators, the number of days they led or lagged disease phenomena changed over time. Still, if these public health indicators continue to receive high quality data, tracking these indicators closely can help us identify changing health dynamics.

5.4 SAE Steering and Healthcare Results

At IBM Research, Nairobi
For medical applications, adapted from [62] and [70]

Recent work [109] shows that Sparse Autoencoders (SAE) applied to large language model (LLM) layers have neurons corresponding to interpretable concepts. These SAE neurons can be modified to align generated outputs, but only towards **pre-identified** topics and with some parameter tuning. Our approach leverages the observational and modification properties of SAEs to enable alignment for **any** topic. This method 1) scores each SAE neuron by its semantic

similarity to an alignment text and uses them to 2) modify SAE-layer-level outputs by emphasizing topic-aligned neurons. We assess the alignment capabilities of this approach on diverse public topic datasets including Amazon reviews, Medicine, and Sycophancy, across the currently available open-source LLMs and SAE pairs (GPT2 and Gemma 2.0) with multiple SAEs configurations. Experiments aligning to medical prompts reveal several benefits over fine-tuning, including increased average language acceptability (0.25 vs. 0.5), reduced training time across multiple alignment topics ($333.6s$ vs. $62s$), and acceptable inference time for many applications (+$0.00092s/token$).

Generative technologies were used for light document grammar editing and clarity.

Appendix

A.1 Appendix A: Additional Details on Acute Approach

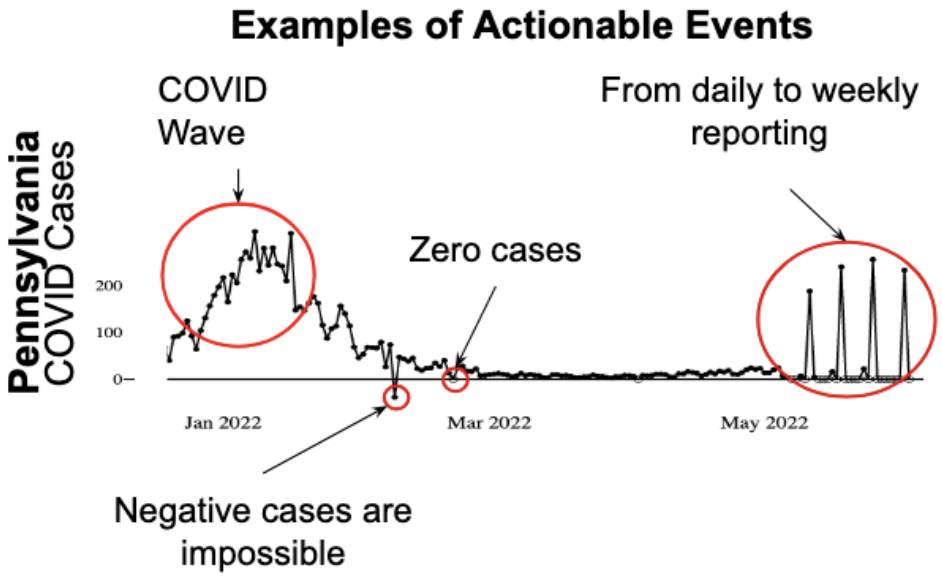


Figure A.1: Additional examples of events in a public health data stream at Delphi.

.

In the first two years of the pandemic, I re-implemented and adapted many of the algorithms detailed in [5] and variations of the validation pipeline from the Epidata package as part of Delphi's tooling, via the validation package in covidcast-indicators. Upon review, the resulting events numbered in the tens of thousands and required continuous parameter tuning. These problems persisted even when considering only higher-tier geographies, such as state- and nation-level data, which tend to be more stable, in part, because the frequency of these events is not necessarily uniform over time (see Fig A.1)

Next, I explicitly modeled the data-generating process and residual distributions using bi-

nomial and zero-inflated binomial distributions, as well as other intuitive modeling approaches from the epidemiological domain. This process of parameter tuning remained inconsistent and unreliable across geographies and time, necessitating an empirical approach that formed the basis of the described thesis methods. As many of these experiments were run on private data, please email me for any additional details.

A.2 Appendix B: Initial FlaSH Evaluation

From the alerting paradigm, the approach that was closest to being deployed was an alerting version of the FlaSH algorithm. This initial architecture was notably different than the ranked list paradigm in the thesis. For example, the outputs were sent to Delphi users via Slack alerts, as shown in Fig. A.2.

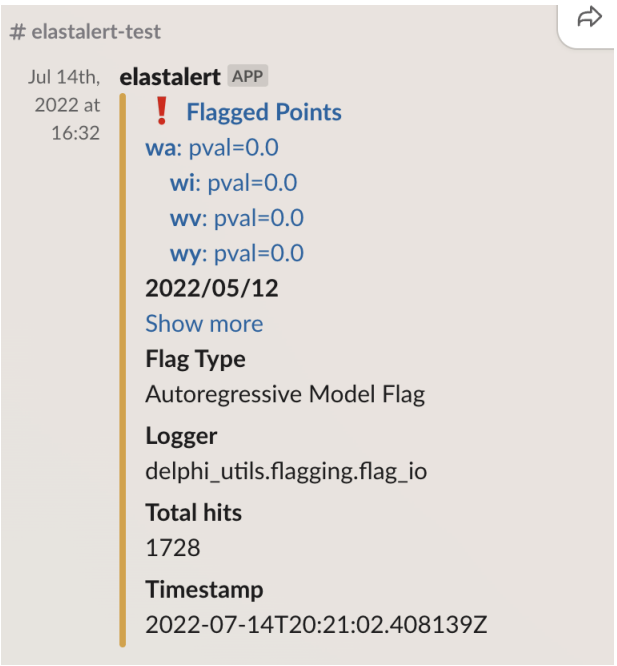


Figure A.2: Initial alert-based approach had some important elements of the final design (e.g. external visualization interface for context), but still relied on alerts and did not involve saving human triaged data

.

These alerts were tested extensively but were never deployed in practice despite strong preliminary practical evaluations over various configurations. However, they informed the first eval-

uation I conducted (published as part of my speaking skills slides) using a threshold ($\alpha = 0.01$) to classify points as alerts.

The test samples were collected using a random, stratified sample across % Doctors Visits with COVID-Like Illnesses and COVID-19 Case Counts streams (90 Points/Location) for Los Angeles, NY, USA, TX, and Loving County, TX. In that evaluation, 5 volunteers were tasked with (1) identifying interesting data points in select data streams and then (2) classifying specific points that were output at the top of different algorithms (outlier detection based on xStream [80], Prophet [108]) on a sliding scale of if they disagree or are confident that the specific point is an outlier. The results from this process provided a ground truth measure that included the users' preferences and their annotations of specific points identified by the candidate algorithms. Out of the starting set of 450 points across the selected streams, a majority of reviewers, followed by a consensus among the group of volunteers, confidently labeled **28** points as ground truth, which was then compared to FlaSH and out-of-the-box evaluations for Prophet and xStream outlier detection implementations.

Metric	**FlaSH**	**Prophet**	**xStream**
Accuracy	0.98	0.93	0.84
F1 Score	0.90	0.50	0.12
Precision	0.82	1.00	0.12
Recall	1.00	0.33	0.11
Balanced Acc	0.99	0.67	0.51

Table A.1: Performance metrics of FlaSH vs. state-of-the-art outlier detection algorithms in initial binary experiment based on alerts.

As shown in Table A.1, FlaSH demonstrated competitive accuracy metrics. It also had performance advantages (5m one-time cost and 0.17m daily evaluation cost per signal vs. 1.16m for Prophet and 0.33m for xStream). Despite strong results, the need for α tuning and the scaling challenges of the alerting approach—despite modifications for aggregation, tiering, and filtering anomalous data—made this alerting design less promising for data monitoring than the redesigned ranking approach.

Bibliography

[1] 2020. URL https://www.cdc.gov/nssp/biosense/docs/BioSense_Data_Quality_Dashboard.2020.pdf. 1.2, (a), (b)

[2] 2020. URL https://cdn.who.int/media/docs/default-source/data-quality-pages/2021_-dqa_module-2_desk-review-of-data-quality.pdf?sfvrsn=7a0999e_9. 1.2, 1.2

[3] 2023. URL https://docs.dhis2.org/en/develop/using-the-api/dhis-core-version-240/data-validation.html. 1.2

[4] Asmaa Abduldaem and Andy Gravell. Principles for the design and development of dashboards: literature review. *Proceedings of INTCESS*, pages 1307–1316, 2019. 4.1

[5] Charu C Aggarwal and Charu C Aggarwal. *An introduction to outlier analysis*. Springer, 2017. A.1

[6] Marcel Altendeitering, Stephan Dübler, and Tobias Moritz Guggenberger. Data quality in data ecosystems: Towards a design theory. 2022. 1.2

[7] Jinwon An and Sungzoon Cho. Variational autoencoder based anomaly detection using reconstruction probability. *Special lecture on IE*, 2(1):1–18, 2015. 3

[8] Wilson Andrews and Lisa Waananen Jones. Data modification, Mar 2023. URL https://www.nytimes.com/2023/03/22/us/covid-data-cdc.html. 1.2

[9] Fabrizio Angiulli and Clara Pizzuti. Fast outlier detection in high dimensional spaces. In *Principles of Data Mining and Knowledge Discovery: 6th European Conference*, pages 15–27. Springer, 2002. 3

[10] Hiba Asri, Hajar Mousannif, Hassan Al Moatassime, and Thomas Noel. Big data in healthcare: Challenges and opportunities. In *2015 International conference on cloud technologies and applications (CloudTech)*, pages 1–7. IEEE, 2015. 1.2

[11] American Hospital Association. coronoavirus hhs makes significant changes to covid-19

daily data reporting process, Jul 2020. URL https://www.aha.org. 1.2

[12] Gökce Babür, Anastasiya Bosova, Qian Chen, Guo Xianda, Guo Zifeng, Ananya A Joshi, Niki Kanaki, Li Jin, Paola A Mendoza Salinas, Mahshid Motie, et al. Creative data mining: Documentation of the teaching results from the spring semester 2018. Technical report, ETH Zurich, 2018. (document)

[13] Matthew Biggerstaff, Michael Johansson, David Alper, Logan C Brooks, Prithwish Chakraborty, David C Farrow, Sangwon Hyun, Sasikiran Kandula, Craig McGowan, Naren Ramakrishnan, et al. Results from the second year of a collaborative effort to forecast influenza seasons in the united states. *Epidemics*, 24:26–33, 2018. 4.1

[14] David L Blazes and Sheri H Lewis. *Disease surveillance: technological contributions to global health security*. CRC Press, 2016. 1.3

[15] Ane Blázquez-García, Angel Conde, Usue Mori, and Jose A Lozano. A review on outlier/anomaly detection in time series data. *ACM Computing Surveys (CSUR)*, 54(3):1–33, 2021. 1.2, 2.1, 3.1

[16] Markus M Breunig, Hans-Peter Kriegel, Raymond T Ng, and Jörg Sander. Lof: identifying density-based local outliers. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, pages 93–104, 2000. 3

[17] David L Buckeridge, Howard Burkom, Murray Campbell, William R Hogan, Andrew W Moore, et al. Algorithms for rapid outbreak detection: a research synthesis. *Journal of biomedical informatics*, 38(2):99–113, 2005. 1.2

[18] Howard Burkom, Wayne Loschen, Richard Wojcik, Rekha Holtry, Monika Punjabi, Martina Siwek, Sheri Lewis, et al. Electronic surveillance system for the early notification of community-based epidemics (essence): overview, components, and public health applications. *JMIR public health and surveillance*, 7(6):e26303, 2021. 1.2

[19] Howard S Burkom. Evolution of public health surveillance: status and recommendations, 2017. 1.2, 1.2, 4.1

[20] Howard S Burkom, S Murphy, J Coberly, and K Hurt-Mullen. Public health monitoring tools for multiple data streams. *Morbidity and Mortality Weekly Report*, 54(Supplement

on Syndromic Surveillance):55–62, 2005. 1.2

[21] Li Cai and Yangyong Zhu. The challenges of data quality and data quality assessment in the big data era. *Data science journal*, 14:2–2, 2015. 1.2, 1.2

[22] Nan Cao, Chaoguang Lin, Qiuhan Zhu, Yu-Ru Lin, Xian Teng, and Xidao Wen. Voila: Visual anomaly detection and monitoring with streaming spatiotemporal data. *IEEE transactions on visualization and computer graphics*, 24(1):23–33, 2017. 4.1

[23] Lauren N Carroll, Alan P Au, Landon Todd Detwiler, Tsung-chieh Fu, Ian S Painter, and Neil F Abernethy. Visualization and analytics tools for infectious disease epidemiology: a systematic review. *Journal of biomedical informatics*, 51:287–298, 2014. 4.1, 4.1

[24] Hsinchun Chen, Daniel Zeng, and Ping Yan. *Infectious disease informatics: syndromic surveillance for public health and biodefense*, volume 21. Springer, 2010. 3

[25] Hsinchun Chen, Daniel Zeng, Ping Yan, Hsinchun Chen, Daniel Zeng, and Ping Yan. Biosense. *Infectious Disease Informatics: Syndromic Surveillance for Public Health and BioDefense*, pages 109–119, 2010. 1.2

[26] Isaac Cho, Ryan Wesslen, Alireza Karduni, Sashank Santhanam, Samira Shaikh, and Wenwen Dou. The anchoring effect in decision-making with visual analytics. In *2017 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pages 116–126. IEEE, 2017. 4.1

[27] Michael A Coletta and Hong Zhou. What can you really do with 35,000 statistical alerts a week anyways? *Online Journal of Public Health Informatics*, 11(1):e62444, 2019. 1.2

[28] Andrew Cook, Göksel Mısırlı, and Zhong Fan. Anomaly detection for iot time-series data: A survey. *IEEE Internet of Things Journal*, 7(7):6481–6494, 2019. 3.1

[29] Johns Hopkins CSSE. The times switches to c.d.c. covid data, ending daily collection, Mar 2023. URL `https://github.com/CSSEGISandData/COVID-19/tree/master/csse_covid_19_data`. 1.2

[30] Understanding Patient Data. Using patient data is vital to improve health and care for everyone. `https://understandingpatientdata.org.uk/why`, 2018. 1.2

[31] Definitive-Healthcare. Hospital referral regions. https://www.definitivehc.com, 2023. Accessed: 2023-06-05. 3.2

[32] Ziquan Deng, Xiwei Xuan, Kwan-Liu Ma, and Zhaodan Kong. A reliable framework for human-in-the-loop anomaly detection in time series. *arXiv preprint arXiv:2405.03234*, 2024. 4.2

[33] Xueying Ding, Nikita Seleznev, Senthil Kumar, C Bayan Bruss, and Leman Akoglu. From explanation to action: An end-to-end human-in-the-loop framework for anomaly reasoning and management. *arXiv preprint arXiv:2304.03368*, 2023. 4.1, 4.1

[34] Ensheng Dong, Jeremy Ratcliff, Tamara D Goyea, Aaron Katz, Ryan Lau, Timothy K Ng, Beatrice Garcia, Evan Bolt, Sarah Prata, David Zhang, et al. The johns hopkins university center for systems science and engineering covid-19 dashboard: data collection process, challenges faced, and lessons learned. *The lancet infectious diseases*, 22(12):e370–e376, 2022. 2.1

[35] Min Du, Feifei Li, Guineng Zheng, and Vivek Srikumar. Deeplog: Anomaly detection and diagnosis from system logs through deep learning. In *Proceedings of the 2017 ACM SIGSAC conference on computer and communications security*, pages 1285–1298, 2017. 1.2, 3

[36] Lisa Ehrlinger and Wolfram Wöß. A survey of data quality measurement and monitoring tools. *Frontiers in big data*, 5:850611, 2022. 1.2

[37] Céline Faverjon and John Berezowski. Choosing the best algorithm for event detection based on the intended application: A conceptual framework for syndromic surveillance. *Journal of biomedical informatics*, 85:126–135, 2018. 1.2

[38] Centers for Disease Control and Prevention. National syndromic surveillance program (nssp). October 2023. 4

[39] Centers for Disease Control and Prevention. National syndromic surveillance program (nssp) new users. `https://www.cdc.gov/nssp/new-users.html`, 2023. 4

[40] US Centers for Disease Control and Prevention. Data modernization initiative. `https://www.cdc.gov/surveillance/data-modernization/`

index.html, April 2023. 1.2, 1.2, 1.2

[41] Ronald D Fricker Jr and Howard S Burkom. Data aggregation in disease surveillance. *Journal of Quality Technology*, 53(1):38–43, 2021. 1.2

[42] M Ivette Gomes and Armelle Guillou. Extreme value theory and statistics of univariate extremes: a review. *International statistical review*, 83(2):263–292, 2015. 3.2

[43] Florian Gottwalt, Elizabeth Chang, and Tharam Dillon. Corrcorr: A feature selection method for multivariate correlation network anomaly detection techniques. *Computers & Security*, 83:234–245, 2019. 1.2

[44] National Electronic Disease Surveillance System Working Group. National electronic disease surveillance system (nedss): a standards-based approach to connect public health and clinical medicine. *Journal of Public Health Management and Practice*, pages 43–50, 2001. 3

[45] Andrew S Grove. *High output management*. Vintage, 2015. 4.1

[46] Yi Guo, Shunan Guo, Zhuochen Jin, Smiti Kaul, David Gotz, and Nan Cao. Survey on visual analysis of event sequence data. *IEEE Transactions on Visualization and Computer Graphics*, 28(12):5091–5112, 2021. 4.1, 1

[47] Manish Gupta, Jing Gao, Charu C Aggarwal, and Jiawei Han. Outlier detection for temporal data: A survey. *IEEE Transactions on Knowledge and data Engineering*, 26(9): 2250–2267, 2013. 3

[48] Esther Hamblion, Neil J Saad, Blanche Greene-Cramer, Adedoyin Awofisayo-Okuyelu, Dubravka Selenic Minet, Anastasia Smirnova, Etsub Engedashet Tahelew, Kaja Kaasik-Aaslav, Lidia Alexandrova Ezerska, Harsh Lata, et al. Global public health intelligence: World health organization operational practices. *PLOS Global Public Health*, 3(9):e0002359, 2023. 3

[49] HHS. Hhs regional offices. https://www.hhs.gov/about/agencies/iea/regional-offices/index.html, 2023. Accessed: 2023-06-01. 3.2

[50] Richard S Hopkins, Catherine C Tong, Howard S Burkom, Judy E Akkina, John Bere-

zowski, Mika Shigematsu, Patrick D Finley, Ian Painter, Roland Gamache, Victor J Del Rio Vilas, et al. A practitioner-driven research agenda for syndromic surveillance. *Public Health Reports*, 132(1_suppl):116S–126S, 2017. 1.2, 4.1

[51] Burkom Howard. How should i set my alerting thresholds? what sensitivity and positive predictive value can i expect? *Presentation to NSSP Group*, 2025. 1.2

[52] Bing Hu, Yanping Chen, and Eamonn Keogh. Time series classification under more realistic assumptions. In *Proceedings of the 2013 SIAM international conference on data mining*, pages 578–586. SIAM, 2013. 4.2

[53] Kyle Hundman, Valentino Constantinou, Christopher Laporte, Ian Colwell, and Tom Soderstrom. Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 387–395, 2018. 3.1

[54] Kyle Hundman, Valentino Constantinou, Christopher Laporte, Ian Colwell, and Tom Soderstrom. Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding. *arXiv preprint arXiv:1802.04431*, 2018. 3

[55] Kathy J Hurt-Mullen and J Coberly. Syndromic surveillance on the epidemiologist's desktop: making sense of much data. *MMWR Morb Mortal Wkly Rep*, 54(Suppl):141–6, 2005. 1.2, 1.2, 4.1

[56] Susanne Hyllestad, Ettore Amato, Karin Nygård, Line Vold, and Preben Aavitsland. The effectiveness of syndromic surveillance for the early detection of waterborne outbreaks: a systematic review. *BMC Infectious Diseases*, 21:1–12, 2021. 4.1

[57] Zahra Jadidi, Shantanu Pal, Mukhtar Hussain, and Kien Nguyen Thanh. Correlation-based anomaly detection in industrial control systems. *Sensors*, 23(3):1561, 2023. 1.2

[58] Andrea Janes, Alberto Sillitti, and Giancarlo Succi. Effective dashboard design. *Cutter IT Journal*, 26(1):17–24, 2013. 4.1

[59] Gonçalo Jesus, António Casimiro, and Anabela Oliveira. A survey on data quality for dependable monitoring in wireless sensor networks. *Sensors*, 17(9):2010, 2017. 1.2

[60] Ananya Joshi. The effect of curcuma longa extract on the rate of aggregation and concentration of proteins in albumen. *International Journal of Pharmaceutical Excipients*, 5(4), 2016. (document)

[61] Ananya Joshi. Creating an automated ideologicaltransformer using moral reframing. 2019. (document)

[62] Ananya Joshi. Enabling new applications with today's mechanistic interpretability toolkit. 2024. 5.4

[63] Ananya Joshi and Clayton Miller. Review of machine learning techniques for mosquito control in urban environments. *Ecological Informatics*, 61:101241, 2021. (document)

[64] Ananya Joshi, Nolan Gormley, Richa Gadgil, Catalina Vajiac, Roni Rosenfeld, and Bryan Wilder. Visualizing public health data streams for data classification. 2023. 4

[65] Ananya Joshi, Kathryn Mazaitis, Roni Rosenfeld, and Bryan Wilder. Computationally assisted quality control for public health data streams. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, pages 6004–6012, 2023. 1.3, 2, 3.3, 4.1

[66] Ananya Joshi, Bryan Wilder, and Roni Rosenfeld. Large scale population-level outliers detection in public health data. 2023. 3

[67] Ananya Joshi, Bryan Wilder, Roni Rosenfeld, and Kathryn Mazaitis. Towards detecting points of interest from public health data streams. 2023. 2

[68] Ananya Joshi, Tina Townes, Nolan Gormley, Luke Neureiter, Roni Rosenfeld, and Bryan Wilder. Outlier ranking in large-scale public health streams. *arXiv preprint arXiv:2401.01459*, 2024. 3

[69] Ananya Joshi, Bryan Wilder, and Roni Rosenfeld. Actionable data monitoring in modern data streams. 2024. 1.2

[70] Cintas Celia Joshi, Ananya and Skyler Speakman. Enabling sparse autoencoders for topic alignment in large language models. 2024. 5.4

[71] Suraj P Kesavan, Takanori Fujiwara, Jianping Kelvin Li, Caitlin Ross, Misbah Mubarak,

Christopher D Carothers, Robert B Ross, and Kwan-Liu Ma. A visual analytics framework for reviewing streaming performance data. In *2020 IEEE Pacific Visualization Symposium (PacificVis)*, pages 206–215. IEEE, 2020. 4.1

[72] Moritz UG Kraemer, Samuel V Scarpino, Vukosi Marivate, Bernardo Gutierrez, Bo Xu, Graham Lee, Jared B Hawkins, Caitlin Rivers, David M Pigott, Rebecca Katz, et al. Data curation during a pandemic and lessons learned from covid-19. *Nature Computational Science*, 1(1):9–10, 2021. 1.2

[73] Nancy Krieger. Public health monitoring: An active phrase for vigilance, warning, guidance, and accountability, 2024. 1.2

[74] Martin Kulldorff, Richard Heffernan, Jessica Hartman, Renato Assunçao, and Farzad Mostashari. A space–time permutation scan statistic for disease outbreak detection. *PLoS medicine*, 2(3):e59, 2005. 1.2

[75] Kwei-Herng Lai, Daochen Zha, Guanchu Wang, Junjie Xu, Yue Zhao, Devesh Kumar, Yile Chen, Purav Zumkhawaka, Minyang Wan, Diego Martinez, and Xia Hu. Tods: An automated time series outlier detection system. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(18):16060–16062, May 2021. 3.1, 3.3

[76] Doris Jung-Lin Lee, Himel Dev, Huizi Hu, Hazem Elmeleegy, and Aditya Parameswaran. Avoiding drill-down fallacies with vispilot: Assisted exploration of data subsets. In *Proceedings of the 24th International Conference on Intelligent User Interfaces,{IUI} 2019, Marina del Ray, CA, USA, March 17-20, 2019*, 2019. 4.1

[77] Jona Lilienthal, Leandra Zanger, Axel Bücher, and Roland Fried. A note on statistical tests for homogeneities in multivariate extreme value models for block maxima. *Environmetrics*, 33(7):e2746, 2022. 3.2

[78] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. Isolation forest. In *2008 eighth ieee international conference on data mining*, pages 413–422. IEEE, 2008. 3

[79] Joseph Lombardo, Howard Burkom, Eugene Elbert, Steven Magruder, Sheryl Happel Lewis, Wayne Loschen, James Sari, Carol Sniegoski, Richard Wojcik, and Julie Pavlin. A systems overview of the electronic surveillance system for the early notification of

community-based epidemics (essence ii). *Journal of urban health*, 80:i32–i42, 2003. 1.2

[80] Emaad Manzoor, Hemank Lamba, and Leman Akoglu. xstream: Outlier detection in feature-evolving data streams. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1963–1972, 2018. A.2

[81] Daniel J McDonald, Jacob Bien, Alden Green, Addison J Hu, Nat DeFries, Sangwon Hyun, Natalia L Oliveira, James Sharpnack, Jingjing Tang, Robert Tibshirani, et al. Can auxiliary indicators improve covid-19 forecasting and hotspot prediction? *Proceedings of the National Academy of Sciences*, 118(51):e2111453118, 2021. 2.2

[82] Brian Montambault, Camelia D Brumar, Michael Behrisch, and Remco Chang. Pixal: Anomaly reasoning with visual analytics. *arXiv preprint arXiv:2205.11004*, 2022. 4.1, 4.1

[83] Roger A Morbey, Alex J Elliot, Andre Charlett, Neville Q Verlander, Nick Andrews, and Gillian E Smith. The application of a novel 'rising activity, multi-level mixed effects, indicator emphasis'(rammie) method for syndromic surveillance in england. *Bioinformatics*, 31(22):3660–3665, 2015. 1.2

[84] Sean Patrick Murphy and Howard Burkom. Recombinant temporal aberration detection algorithms for enhanced biosurveillance. *Journal of the American Medical Informatics Association*, 15(1):77–86, 2008. 1.3

[85] Mehrbakhsh Nilashi, O Keng Boon, Garry Tan, Binshan Lin, and Rabab Abumalloh. Critical data challenges in measuring the performance of sustainable development goals: Solutions and the role of big-data analytics. *Harvard Data Science Review*, 5(3), 2023. 1.2

[86] A. Noufaily, R. A. Morbey, F. J. Colón-González, A. J. Elliot, G. E. Smith, I. R. Lake, and N. McCarthy. Comparison of statistical algorithms for daily syndromic surveillance aberration detection. *Bioinformatics*, 35(17):3110–3118, 2019. doi: 10.1093/bioinformatics/btz060. 1.2

[87] MN Department of Health. Data: Quality, analysis, and interpretation, 10 2022. 1.2

[88] United States Government Accountability Office. Covid-19 data quality and considerations for modeling and analysis. https://www.gao.gov/assets/gao-20-635sp.pdf, 07 2020.

1.2

[89] World Health Organization. World health statistics 2023 - monitoring health for the sdgs. https://www.who.int/publications/i/item/9789240074323, 2023. 1.2

[90] Andrei Paleyes, Raoul-Gabriel Urma, and Neil D Lawrence. Challenges in deploying machine learning: a survey of case studies. *ACM Computing Surveys*, 55(6):1–29, 2022. 2.1

[91] William Peter, Amir H Najmi, and Howard S Burkom. Reducing false alarms in syndromic surveillance. *Statistics in Medicine*, 30(14):1665–1677, 2011. 1.2

[92] Tomáš Pevnỳ. Loda: Lightweight on-line detector of anomalies. *Machine Learning*, 102: 275–304, 2016. 3

[93] Marco Piangerelli, Bardh Prenkaj, Ylenia Rotalinti, Ananya Joshi, and Giovanni Stilo. Workshop on discovering drift phenomena in evolving data landscape (delta). In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 6731–6732, 2024. (document)

[94] Stephan Rabanser, Tim Januschowski, Kashif Rasul, Oliver Borchert, Richard Kurle, Jan Gasthaus, Michael Bohlke-Schneider, Nicolas Papernot, and Valentin Flunkert. Intrinsic anomaly detection for multi-variate time series. *arXiv preprint arXiv:2206.14342*, 2022. 3.1

[95] Alex Reinhart, Logan Brooks, Maria Jahja, Aaron Rumack, Jingjing Tang, Sumit Agrawal, Wael Al Saeed, Taylor Arnold, Amartya Basu, Jacob Bien, et al. Beyond cases and deaths: The benefits of auxiliary data streams in tracking the covid-19 pandemic: An open repository of real-time covid-19 indicators. *Proceedings of the National Academy of Sciences of the United States of America*, 118(51), 2021. 1, 4.2, 5

[96] Alex Reinhart, Logan Brooks, Maria Jahja, Aaron Rumack, Jingjing Tang, Sumit Agrawal, Wael Al Saeed, Taylor Arnold, Amartya Basu, Jacob Bien, et al. An open repository of real-time covid-19 indicators. *Proceedings of the National Academy of Sciences*, 118(51):e2111452118, 2021. 1.1, 1.2, 5

[97] Alex Reinhart, Logan Brooks, Maria Jahja, Aaron Rumack, Jingjing Tang, Sumit Agrawal, Wael Al Saeed, Taylor Arnold, Amartya Basu, Jacob Bien, et al. Wichada la motte-kerr, yeon jin lee, kenneth lee, zachary c. *medRxiv*, 2021. 5

[98] Harald E Rieder. Extreme value theory: A primer. *Lamont-Doherty Earth Observatory*, 2014. 3.2

[99] Ori Rottenstreich, Ariel Kulik, Ananya Joshi, Jennifer Rexford, Gábor Rétvári, and Daniel S Menasché. Cooperative rule caching for sdn switches. In *2020 IEEE 9th International Conference on Cloud Networking (CloudNet)*, pages 1–7. IEEE, 2020. (document)

[100] Ori Rottenstreich, Ariel Kulik, Ananya Joshi, Jennifer Rexford, Gábor Rétvári, and Daniel Sadoc Menasché. Data plane cooperative caching with dependencies. *IEEE Transactions on Network and Service Management*, 19(3):2092–2106, 2021. (document)

[101] Jonas Herskind Sejr and Anna Schneider-Kamp. Explainable outlier detection: What, for whom and why? *Machine Learning with Applications*, 6:100172, 2021. 1.2

[102] Shreya Shankar, Labib Fawaz, Karl Gyllstrom, and Aditya Parameswaran. Automatic and precise data validation for machine learning. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pages 2198–2207, 2023. 1.2

[103] Galit Shmueli and Howard Burkom. Statistical challenges facing early outbreak detection in biosurveillance. *Technometrics*, 52(1):39–51, 2010. 2, 1.2, 4.1

[104] Bolelang H Sibolla, Serena Coetzee, and Terence L Van Zyl. A framework for visual analytics of spatio-temporal sensor observations from data streams. *ISPRS International Journal of Geo-Information*, 7(12):475, 2018. 4.1

[105] David Siegrist and J Pavlin. Bio-alirt biosurveillance detection algorithm evaluation. *Morbidity and Mortality Weekly Report*, pages 152–158, 2004. 1.2

[106] Leslie Z Sokolow, N Grady, H Rolka, D Walker, P McMurray, R English-Bullard, and J Loonsk. Deciphering data anomalies in biosense. *MMWR Morb Mortal Wkly Rep*, 54 (Suppl):133–139, 2005. 4

[107] Janani Sriram, Minho Shin, David Kotz, Anand Rajan, Manoj Sastry, and Mark Yarvis. Challenges in data quality assurance in pervasive health monitoring systems. In *Future of Trust in Computing: Proceedings of the First International Conference Future of Trust in Computing 2008*, pages 129–142. Springer, 2009. 1.2

[108] Sean J. Taylor and Benjamin Letham. Forecasting at scale, 2018. URL `https://facebook.github.io/prophet/`. Accessed: 2025-03-24. A.2

[109] Adly Templeton. *Scaling monosemanticity: Extracting interpretable features from claude 3 sonnet*. Anthropic, 2024. 5.4

[110] UN. United nationals global issues: Big data for sustainable development. `https://www.un.org/en/global-issues/big-data-for-sustainable-development`, 2021. 1.2

[111] Helena Vasconcelos, Matthew Jörke, Madeleine Grunde-McLaughlin, Tobias Gerstenberg, Michael S Bernstein, and Ranjay Krishna. Explanations can reduce overreliance on ai systems during decision-making. *Proceedings of the ACM on Human-Computer Interaction*, 7(CSCW1):1–38, 2023. 4.1

[112] MisChele A Vickers. Monitoring and improving syndromic surveillance data quality. *Online Journal of Public Health Informatics*, 11(1), 2019. 1.2

[113] Victor Del Rio Vilas, M Kocaman, Howard Burkom, Richard Hopkins, John Berezowski, Ian Painter, Julia Gunn, G Montibeller, M Convertino, LC Streichert, et al. A value-driven framework for the evaluation of biosurveillance systems. *Online Journal of Public Health Informatics*, 9(1), 2017. 1.2

[114] Yao Wang, Zhaowei Wang, Zejun Xie, Nengwen Zhao, Junjie Chen, Wenchi Zhang, Kaixin Sui, and Dan Pei. Practical and white-box anomaly detection through unsupervised and active learning. In *2020 29th international conference on computer communications and networks (ICCCN)*, pages 1–9. IEEE, 2020. 4.2

[115] William Webber, Alistair Moffat, and Justin Zobel. A similarity measure for indefinite rankings. *ACM Transactions on Information Systems (TOIS)*, 28(4):1–38, 2010. 2.3

[116] BP Welford. Note on a method for calculating corrected sums of squares and products.

Technometrics, 4(3):419–420, 1962. 4.1

[117] WHO. Covid-19 forecast hub data reports (google group). Mar. 2021. Accessed: 2024-09-10. 1.2

[118] WHO. Who hub for pandemic and epidemic intelligence. https://pandemichub.who.int/publications/m/item/the-who-hub-for-pandemic-and-epidemic-intelligence-strategy-paper, note = Accessed: 2023-06-05, Dec. 2022. 1.2

[119] WHO. Who hub for pandemic and epidemic intelligence. `https://www.who.int/initiatives/preparedness-and-resilience-for-emerging-threats`, Mar. 2023. 1.2

[120] Weng-Keen Wong. *Data mining for early disease outbreak detection*. Carnegie Mellon University, 2004. 2.1

[121] Renjie Wu and Eamonn Keogh. Current time series anomaly detection benchmarks are flawed and are creating the illusion of progress. *IEEE Transactions on Knowledge and Data Engineering*, 2021. 1.2, 2, 4.2

[122] Xingjiao Wu, Luwei Xiao, Yixuan Sun, Junhang Zhang, Tianlong Ma, and Liang He. A survey of human-in-the-loop for machine learning. *Future Generation Computer Systems*, 135:364–381, 2022. 4.2

[123] Xiaochen Xian, Andi Wang, and Kaibo Liu. A nonparametric adaptive sampling strategy for online monitoring of big data streams. *Technometrics*, 60(1):14–25, 2018. 1.2

[124] Xiaochen Xian, Alexander Semenov, Yaodan Hu, Andi Wang, and Yier Jin. Adaptive sampling and quick anomaly detection in large networks. *IEEE Transactions on Automation Science and Engineering*, 2022. 1.2

[125] Chin-Chia Michael Yeh, Yan Zhu, Liudmila Ulanova, Nurjahan Begum, Yifei Ding, Hoang Anh Dau, Diego Furtado Silva, Abdullah Mueen, and Eamonn Keogh. Matrix profile i: all pairs similarity joins for time series: a unifying view that includes motifs, discords and shapelets. In *2016 IEEE 16th international conference on data mining (ICDM)*, pages 1317–1322. Ieee, 2016. 4.2