Computationally Assisted Quality Control for Public Health Data Streams

Ananya Joshi , **Kathryn Mazaitis** , **Roni Rosenfeld** , **Bryan Wilder**
Carnegie Mellon University
{aajoshi, kmazaitis, rrosenfeld, bwilder}@andrew.cmu.edu

Abstract

Irregularities in public health data streams (like COVID-19 Cases) hamper data-driven decision-making for public health stakeholders. A real-time, computer-generated list of the most important, outlying data points from thousands of daily-updated public health data streams could assist an expert reviewer in identifying these irregularities. However, existing outlier detection frameworks perform poorly on this task because they do not account for the data volume or for the statistical properties of public health streams. Accordingly, we developed FlaSH (**Fla**gging **S**treams in public **H**ealth), a practical outlier detection framework for public health data users that uses simple, scalable models to capture these statistical properties explicitly. In an experiment where human experts evaluate FlaSH and existing methods (including deep learning approaches), FlaSH scales to the data volume of this task, matches or exceeds these other methods in mean accuracy, and identifies the outlier points that users empirically rate as more helpful. Based on these results, FlaSH has been deployed on data streams used by public health stakeholders.

1 Motivation and Introduction

During the COVID-19 pandemic, daily-updated real-time public health data was used directly [CDC, 2022] or as input to methods that informed critical healthcare decisions and policies [Yu *et al.*, 2021] in support of Sustainable Development Goals such as good health and well being. However, aspects of public health data have hampered this data-driven decision-making in several ways. These include issues like data delays, corrections, and recording errors [Dong *et al.*, 2022; Sáez *et al.*, 2021] that may have masked important trends in disease progression [Kreps and Kriner, 2020], as shown in Fig. 1. Additionally, COVID variants or policy changes often cause sudden, notable distribution shifts in the data [Zhu *et al.*, 2021]. Finally, public health data streams are known to be biased or incomplete [Leslie *et al.*, 2021]. For example, regions with low healthcare resource availability may not have accurate COVID case counts.

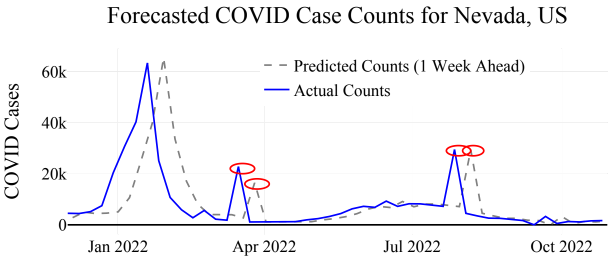

Figure 1: Temporal irregularities in actual case counts, shown by the circled, large spikes in March and July 2022, when cases were trending down, resulted in similar spikes for predicted counts that were then sent to the US Centers for Disease Control and Surveillance.

Addressing these issues is a significant challenge for any organization that curates public health data streams [Kraemer *et al.*, 2021], including the Delphi Group at Carnegie Mellon University (Delphi). Delphi employs a team of full-time developers, statisticians, researchers, and product managers to maintain an accurate and performant public health data source[1]. Delphi's publicly available API [Farrow *et al.*, 2015] and other data products are regularly used by public health authorities in the United States (US), along with researchers, forecasters, journalists, and other users (totaling visits from over 78k unique IP addresses in January 2022). These stakeholders recommended that Delphi continuously monitor their data streams for irregularities so that Delphi's data users have more information about data quality issues, the state of the pandemic, and changes in regional disease behavior, to directly support data-driven decision-making.

To act on this recommendation, expert human reviewers in Delphi would need to regularly monitor at least ten thousand data streams for stakeholders (e.g. cases, deaths, and hospitalizations, at several geographical resolutions, including county, state, territory, and national level resolutions). If done manually, this type of monitoring is prohibitively expensive [Kraemer *et al.*, 2021]. Even if it were feasible, trained reviewers frequently miss critical irregularities due to the sheer reviewing load. While some outliers are so extreme that they require no human review, many outliers that signify irregularities are more nuanced and require close human attention. Computationally assisted quality control, where a reviewer only inspects the top entries from a computer-

[1] Delphi's open source repositories can be found at https://github.com/cmu-delphi

generated ranked list of outlier streams that a human should review, is promising because it could prioritize the reviewer's time for irregularity detection while retaining the trust and expertise a reviewer brings.

Creating this computationally ranked list of outliers in public health streams is a difficult task. In addition to the practical constraints of operating over the large data volume necessitated by this task, outlier detection methods must be robust to the statistical *noise*, *nonstationarity*, *day of week effects*, and *limited historical data* that are prevalent in public health streams in order to provide helpful recommendations [McDonald *et al.*, 2021; Reinhart *et al.*, 2021; Wang *et al.*, 2021]. Further, the outlier detection methods must be simple and intuitive for reviewers to understand and trust them on this task.

To address these challenges, we present FlaSH (**Fla**gging **S**treams in public **H**ealth), available open source at https://github.com/cmu-delphi/covidcast-indicators/tree/main/_delphi_utils_python/delphi_utils/flash_eval. FlaSH is a new outlier detection framework that produces a *ranked list of recent values from data streams that most warrant human inspection*. FlaSH uses simple, scalable, and intuitive models to explicitly capture the statistical properties of public health data. To address challenges in evaluating unsupervised outlier detection methods in time series data like FlaSH, we also developed and conducted a classification and ranking evaluation of FlaSH's performance using input from several expert human reviewers. This is especially important given that many recent works in anomaly detection use semi-synthetic or simulation evaluations that may not truly reflect an expert user's assessment of the method utility. In this evaluation, FlaSH matches or outperforms previous outlier detection methods, including recent deep learning baselines.

2 Practical Irregularity Detection Goals

Our goal is to develop a framework that assists reviewers in detecting important irregularities in Delphi's data streams on behalf of public health data users. The data streams available to the Delphi Group vary by source (local governments, hospitals, private companies, and surveys), and each source has its own dynamics and measurement definitions. For example, the Johns Hopkins Centers for System Science and Engineering (JHU CSSE) COVID-19 source only curates data from publicly available reports [Dong *et al.*, 2022]. They also only report real-time cumulative estimates. Thus, subsequent corrections to the cumulative figures can appear as large spikes or even negative values in derived daily case counts.

Detecting such irregularities across many sources is uniquely challenging for typical outlier detection methods, leading to a range of failure modes observed in our experiments. First, modern deep learning methods for outlier detection struggle with the large number of time series, each with a short history and rapid distribution shifts [Paleyes *et al.*, 2022]. To perform well, these highly parameterized models require long training histories often unavailable 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 are not attuned to the specific structure of public health data and struggle to accurately identify irregularities [Wong, 2004]. Third, neither class can leverage features of public health data streams that could assist with diagnosing irregularities. Because of these limitations, Delphi currently relies on volunteers and group members to *manually* report issues on all data streams as they encounter them, but this process is unsystematic and expensive.

To start, the proposed outlier detection method should detect specific types of outliers present in public health streams that are relevant to Delphi's stakeholders so that the method is both context and user-dependent [Sejr and Schneider-Kamp, 2021]. To identify these outlier categories, we conducted an exploratory analysis on data streams[2] of COVID Case Counts and Ratios, COVID Deaths, Hospital Visits, Google Symptoms Trends, and Doctors Visits at the national, state, territory, and county level resolutions from the first available date of the streams until December 2021. Using these streams, which each have a different possible range of values based on the region's population and the measurement quantity, we defined the following categories of outliers based on their ability to assist reviewers with identifying irregularities:

Out of Range Values and Global Outliers. These outliers are typically due to retrospective updates made by a data source in the value of a cumulative quantity. Out of range examples include "negative" new cases (if the cumulative total was revised down) or more cases reported on a day than the population of the geographic area due to multi-day batched reports. Similarly, global outliers usually appear as large positive or negative spikes, but the 'global' outlier thresholds may change over time as rapidly shifting disease dynamics undermine static thresholds. Still, both of these outliers are relatively easy to identify and very rarely require human review.

Day of Week Outliers. Many public health streams have systematic *day of week effects* [Reinhart *et al.*, 2021]. For example, fewer COVID cases are reported on weekends partly because fewer people test on weekends. Day of week outliers occur when reported data points are anomalous relative to the expectation for their day of the week (even if they are within distribution for the stream as a whole). Unlike out of range or global outliers, day of week outliers are more difficult for humans to notice but may still indicate an irregularity in the stream.

Trendline Outliers. Data that deviate strongly from the recent trend (e.g. case counts were rising last week, but today's count is low) or from the recent trends of close geographic regions warrant attention. These phenomena are the most difficult for humans to detect and can indicate critical irregularities in the context of recent data.

To address failure modes from existing methods and detect these outlier categories, the proposed method must be intuitive, scale to the data volume, and provide outputs (outlier

[2] 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.

scores) that are correct and complement human judgment in this task. Practically, this requires the method to be a single-pass, point detection algorithm that integrates explainable AI and human computing interaction insights. Further, the ranking for the FlaSH list shown to the expert reviewers will be based on the trendline outlier scores because they can indicate critical irregularities in the context of recent data. Reviewers will also benefit from inspecting the global and day of week outlier scores reported alongside. Finally, each of these desired criteria must be evaluated to justifiably compare different outlier detection methods for this task.

3 FlaSH Outlier Detection Method

FlaSH formalizes the outlier detection problem discussed in the previous section as a model-based hypothesis test [Blázquez-García *et al.*, 2021]. 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[3], and T is the current time. When it is necessary to discuss 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. 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 a sequence of 3 steps:

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 out of range, global, and day of week 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 prior to training. The key challenge in this step is to accommodate the statistical properties of public health data.

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, like many deep learning models, are unsuitable because of the limited data history available to tune the model, the expensive ground truth labels, and the rapid distribution shifts in the types of irregularities per stream. Further, stakeholders prioritize interpretability, so the model class must be intuitive.

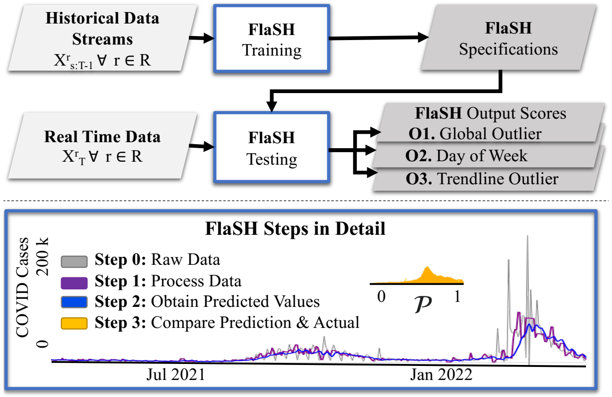

Figure 2: In the FlaSH outlier detection method, data stream inputs are processed through FlaSH to generate informational outlier scores. FlaSH itself has three steps. The raw data (gray) is processed [S1] (purple), and model m is used to predict future values [S2] (blue). Then, the historical performance of model m is captured with the test statistic distribution (gold), and this distribution is used to compare predicted and actual values [S3].

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.

We now discuss each step, as displayed in Fig. 2.

3.1 Process Data

Trendline outliers cannot be reliably identified if the model is trained on data that also includes out of range, global, and day of week outliers. However the thresholds for determining these outliers change with distribution shifts in the stream. To address this challenge, first, different COVID regimes, or waves, are identified via changepoints. Then, within each regime, existing outliers are detected and imputed.

Identifying Changepoints in Nonstationary Streams

Values that would be outliers when there is no COVID wave may not be outliers during a COVID wave. This phenomenon of distinct, underlying waves, or regimes, in public health streams is why they are known to be statistically *nonstationary* [Chimmula *et al.*, 2020][4]. To identify these regimes in historical data, FlaSH uses the Pelt Changepoint Algorithm [Killick *et al.*, 2012; Truong *et al.*, 2020], parametrized with a Gaussian model and a minimum of four weeks between change points[5]. However, individual streams may be

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

[4] Operationally, we consider regimes present in streams that are updated daily with at least 60 historical data points. On streams with fewer than 60 data points, we provide interquartile range-defined outliers.

[5] Four weeks is the maximum horizon for many short-term forecasts [Cramer *et al.*, 2022], likely because health dynamics change drastically after that horizon.

very noisy, and Pelt sometimes overfits to this noise to return regimes inconsistent with expert knowledge of disease dynamics. Therefore, we take advantage of geographical dependencies[6] by searching for changepoints that are jointly applicable across a set of nearby regions. Specifically, we run the Pelt algorithm on the streams for all counties within a given state, jointly optimizing Pelt's objective across these regions to find changepoint days that describe the regimes well across these streams.

While Pelt can identify changepoints in historical data, it does not identify if real-time data represents a changepoint. Instead of retraining FlaSH daily to find new changepoints, which would be computationally expensive, FlaSH assumes there is no changepoint until there is sufficient evidence of nonstationarity to trigger retraining as follows. Under the null hypothesis, there is no changepoint, and p-values are uniformly distributed by definition. If the distribution of the test statistic for the hypothesis test H_0 significantly shifts, then the Kolmogrov-Smirnov test can identify whether the empirical p-value distribution since the last retraining deviates significantly from the uniform distribution. The user can select the test significance level α according to their desired trade-off between the computational expense of retraining and increased accuracy. Even if a new changepoint is not detected, FlaSH is retrained every 3 months, which roughly corresponds to a change in season, and the expert reviewer can retrain at any time.

Identifying Outliers Within Regimes

Within each changepoint-defined regime, FlaSH identifies out of range, day of week, and global outliers, and it imputes non-outlying values that are later used for modeling. First, out of range outliers, like negative COVID Cases, are identified and imputed to be in range. Second, data is separated by day of week, and points where $|z_{score}| \geq 3$ with respect to the points for that day of the week in the regime are identified as day of week outliers. The median value of the day of the week in that regime is then imputed for downstream analysis. Day of week sensitivity is important here because of systematic patterns across the week, like that the median value for Sundays is usually lower than the median for Tuesdays.

Third, we process the time series to remove systematic day of week effects (unlike the previous step, which handled points far outside the typical pattern for their weekday). FlaSH uses a Poisson regression method w (part of Delphi's public API[7]) which outputs a weekday-corrected value $w(X_t)$. This model removes systematic differences in mean values across days (e.g. by scaling values on Saturday up and scaling Mondays down) to obtain a time series without day of week effects. Removing such systematic periodicity enables downstream predictive models to fit the data-generating process using fewer parameters.

Finally, after the day of week correction, FlaSH addresses the *noisiness* of the stream by identifying global outliers in the day of week corrected data as those with $|z_{score}| \geq 3$, calculated from all weekdays in the day of week corrected data. These points are imputed using the mean value of the current regime. Having removed out of range, global, and day of week outliers, FlaSH treats the processed data across all regimes as the null distribution and can now identify trendline outliers as specified by the following two steps of FlaSH.

3.2 Obtain Predicted Values

To identify trendline outliers, FlaSH uses a small sample of the processed historical data to train a predictive model m for X_T from model class $\mathcal{M}$ and then uses the remaining processed historical data to characterize the performance of the model. Specifically, the training set for FlaSH's null hypothesis model is the maximum of 10% of the historical data or 30 points. FlaSH then uses $\mathcal{M}$: Linear Autoregressive (AR) models (lag=7) , where m is characterized by the linear weights, $\hat{\beta}$, fit during training. This class of models is preferred in public health applications for its simplicity and performance with *limited historical data* [McDonald *et al.*, 2021]. The remaining processed historical data (not used to fit the model) is used to generate predictions $\hat{X}_t$.

3.3 Compare Predicted and Observed Values

Models from any model class $\mathcal{M}$ fit with the null historical data will not perform uniformly across all streams. Accordingly, out-of-sample data is essential to quantify the typical discrepancy between model predictions and observed values per stream. Outliers can then be identified when the discrepancy between predictions and observations is more than typical, as determined by a distribution of historical performance. For example, if a model consistently predicts higher values than what is observed, then the outlier score should reflect the fact that $\hat{X}_T > X_T$ is not surprising.

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 k_t:

$$k_t = (P(w(X_t^r) < D))$$

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

This test models the counts in a region as a binomial distribution D. The probability of infection per person is the number of predicted counts divided by the region's population size. Intuitively, we test the hypothesis that the actual observed counts are drawn from a distribution parameterized by our predictions. Extreme values of the test statistic indicate that the observations were much bigger or smaller than expected given the predictions. 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, with a minimum of 30 points characterizing each

[6] Data reporting and health policies are generally consistent at the state level [Simon, 2021].

[7] https://github.com/cmu-delphi/covidcast-indicators/blob/main/_delphi_utils_python/delphi_utils/weekday.py

distribution if there are only 60 days of historical data. Accordingly, we define the pooled test statistic distribution $\mathcal{P}$, specified by $\bigcup_{r \in R} k^r_{30:T-1}$, 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. 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).

3.4 FlaSH Output

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

4 FlaSH Labels, Evaluation, & Feedback

As noted in the literature, accurately evaluating algorithms for unsupervised time series outlier detection is challenging [Wu and Keogh, 2021]. In most previous work, human-generated labels have not been provided by experts (instead coming from readily available subjects such as students or Mechanical Turk workers). Non-expert labels are noisy since identifying outliers often requires domain-specific knowledge. However, outlier detection method performance on simulated data or data with synthetically injected outliers [Lai *et al.*, 2021] rarely translates to practical performance on real-world data in epidemiology generally [Wong, 2004].

One of our key contributions is to address this limitation in the outlier detection literature via a rigorous, real-world evaluation of outlier detection methods for public health data. We obtained high-quality labeled data from human subject matter experts- Delphi members who are directly involved in building statistical or software systems using public health data and who regularly encounter the impact of data irregularities. In contrast to the binary labels standard in previous work, which may not be sufficient as different experts have different thresholds for outlier determination, asking experts to rank outliers that warrant human inspection provides a more informative comparison for FlaSH's output.

For additional evaluation rigor, we preregistered the FlaSH version, survey design, and analysis before data collection began [Joshi *et al.*, 2023]. This ensures that our algorithm was finalized before any data collection occurred, giving an unbiased (prospective) evaluation of FlaSH's performance. Real-time COVID Case data streams (3341 streams at county, state, territory, and national levels available from May 2020-May 2022) were initially sourced daily from JHU CSSE. Of these, five streams, including the national stream, were randomly chosen from the following sets to ensure stream variety for the evaluation: the top 10% of populous states (Pennsylvania), bottom 90% of populous states (Arkansas), top 10% populous FIPS regions (36081 Queens County, NY), and bottom 90% populous FIPS regions (72043 Coamo Municipality, PR), as per the US Census.

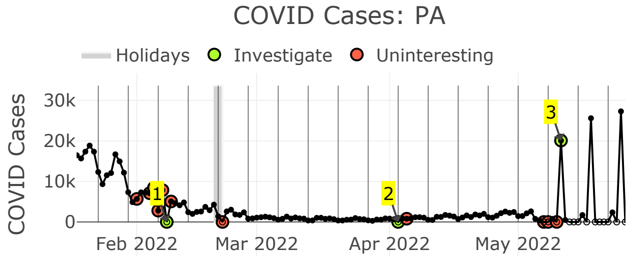

Figure 3: Example of a Survey Task. Respondents click on the time series plot to mark points as unevaluated, uninteresting, or warrants investigation. They also rank points that warrant investigation, and these rankings appear on the plot in yellow. Respondents could zoom, pan, and see a 7 day average per graph.

4.1 Survey and Analysis

To gather ground truth in order to understand FlaSH's performance on empirical data, we designed a web survey that has 10 interactive questions (Fig. 3) [8]. First, respondents classify candidate data points from a public health stream as 'warrants human investigation' or 'uninteresting'. Then, they are asked to rank (with possible ties) the subset of these candidates they think would warrant additional human inspection.

To form this candidate set of evaluation points, we needed to select stream values that are at least somewhat anomalous. That way, survey respondents could meaningfully *distinguish* between points that are potentially anomalous. In practice, we expect $< 1\%$ of all points in a stream to represent irregularities. Accordingly, this candidate set is formed by taking the union of the top outlying points output by both FlaSH and 8 previously proposed outlier detection methods given all historical data (see Sec. 5). By filtering to data points at the top of at least one algorithm, 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.

In Questions 1-5 (Q1-5), the candidate set was formed from the top 5% of points from at least one algorithm for each of the 5 possible data streams. In Q6-10, respondents were asked to reconsider the top 2% of points from at least one algorithm, a subset of these candidates from Q1-5, in more detail to test for respondent internal consistency. 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 allows us to measure the value added by the algorithm over what would have been obvious to a human.

We evaluate 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). To compare the survey results to the outlier detection method outputs, we use a range of metrics to capture the complexity of real-world outlier detection. Both traditional binary classification and ranking metrics provide information on how well the system finds points that the majority of respondents thinks warrants human inspection. We

[8] https://github.com/Ananya-Joshi/IJCAI23_Supplemental

also seek to understand which points from outlier detection algorithms provided the most benefit to users based on self-reports. The points which were rated as both highly anomalous and unlikely to have been flagged without an algorithm are the most valuable potential contributions of computationally assisted quality control.

Survey Quality. The total number of survey participants (n=13) is a significant increase over previous work (e.g. n=2 in [Wong, 2004]). We tested for internal consistency in respondents that answered both sets of survey questions (Q1-5 and Q6-10) by measuring response centrality between the Copeland aggregate per paired question (e.g. Q1 & Q6, Q2 & Q7) and the raw ranks per person. High centrality values (0.83 ± 0.13) suggest that respondents generally were consistent in their pairwise preferences between the two sets. Still, the average number of points that warranted inspection per person varied from < 2 to > 6, supporting that the *threshold* for identifying points of interest varies greatly by individual and reinforcing the importance of sampling a wider range of experts than historical standards suggest.

5 Results and Analysis

We compare the *trendline outlier scores* from FlaSH to outlier scores from the following off-the-shelf outlier detection algorithm baselines implemented in TODS[9] that span recent deep learning methods, classical machine learning, and statistical approaches: DeepLog [Du *et al.*, 2017], Telemanom (Telem.) [Hundman *et al.*, 2018], Variational Autoencoder (VAE) [An and Cho, 2015], Local Outlier Factor (LOF) [Breunig *et al.*, 2000], Lightweight Online Detector of Anomalies (LODA) [Pevnỳ, 2016], Isolation Forest (IF) [Liu *et al.*, 2008], k-Nearest Neighbors (KNN) [Angiulli and Pizzuti, 2002], and Linear AR Model [Agarwal *et al.*, 2022]. These methods have in-built data processing [S1] and prediction comparison [S3] steps, just like FlaSH. We use default hyperparameters for the TODS implementations because there is not enough recent data to select hyperparameters. In fact, many of these models are too costly to even train once on the full set of streams, much less to do hyperparameter selection with many repeated runs. One strength of FlaSH is that it has no hyperparameters because it is designed for this task.

Additionally, for an ablation study, we compare 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. We find that FlaSH easily scales to a large number of data streams, while many deep learning methods become infeasible. Performance statistics (Table 5) were reported from experiments using a 2.6 GHz 6-Core Intel Core i7 machine. Each algorithm was trained on the full 3341 JHU CSSE COVID-19 case streams with 60 days of history. This setup mimics the setting that we expect algorithms to scale to in deployment. A few algorithms (mainly deep learning algorithms) did not finish training within one day (DNF). Training time can only increase for these deep learning implementations as historical data increases. 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. Although many of the existing outlier detection methods have infeasibly long training times for daily deployment, we compare the performance of all algorithms using the labeled data from the survey. Table 5 shows the 95% CI of various traditional binary and ranking outlier detection metrics across all participants for Q1-5 per algorithm.

In the binary analysis, points identified by the majority of respondents as to-investigate were marked as outliers (ground truth)[10]. To calculate binary labels from each algorithm to compare to this ground truth, we used the following process. Let k denote the number of human-identified outliers for a stream. For each algorithm, we took the top k points, 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. We report the 95% CI metrics per person and per question for accuracy, balanced accuracy score, F1 score, and the ROC-AUC score. On average, FlaSH meets or exceeds the performance of all baselines in the binary analysis. FlaSH performs slightly better 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 other kinds of outliers (e.g. trendline or day of week outliers). For the ranking analysis, each algorithm's ranking of the subset points available in Q1-5 was compared to each respondent's rankings using Hamming distance (lower is better), Ranked-Biased Overlap (RBO) [Webber *et al.*, 2010], 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. While the TODS AR method is uncompetitive with other approaches, by using data processed using FlaSH's first step (Mixed), 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, as reflected in the results. Compared to the TODS implementation with the same model class $\mathcal{M}$, FlaSH's processing [S1] and comparison [S3] steps together provide clear performance benefits.

FlaSH can complement human judgment. We find that FlaSH identifies useful points that were unlikely to have been inspected without computational assistance (via an algorithm identifying the point), as shown in the Assistive Rank section of Table 5. Specifically, we examine 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

[9] Each algorithm had a setting 7 day windows where applicable to account for *day of week effects*.

[10] The base rates were: US (2/14), Pennsylvania (9/14), Arkansas (3/16), FIPS 36081 (6/24) and FIPS 72043 (5/21).

Model Class		**AR**			**DeepLog**	**Telem.**	**VAE**	**LOF**	**LODA**	**IF**	**KNN**
Implementation		TODS	Mixed†	**FlaSH**	TODS						
Training (s)		$10.1_{\pm 0.3}$		$169_{\pm 0.8}$	DNF	DNF	DNF	$8_{\pm 0.2}$	$71_{\pm 0.1}$	DNF	$7_{\pm 0.08}$ ✓
Binary	Accuracy	$0.78_{\pm 0.02}$	$0.71_{\pm 0.04}$	$0.8_{\pm 0.03}$ ✓	$0.8_{\pm 0.04}$✓	$0.6_{\pm 0.04}$	$0.76_{\pm 0.04}$	$0.69_{\pm 0.01}$	$0.68_{\pm 0.04}$	$0.79_{\pm 0.04}$	$0.74_{\pm 0.03}$
	Bal.Acc.	$0.68_{\pm 0.02}$	$0.59_{\pm 0.06}$	$0.73_{\pm 0.05}$✓	$0.72_{\pm 0.05}$	$0.42_{\pm 0.03}$	$0.67_{\pm 0.07}$	$0.55_{\pm 0.03}$	$0.54_{\pm 0.05}$	$0.7_{\pm 0.07}$	$0.62_{\pm 0.05}$
	F1	$0.54_{\pm 0.05}$	$0.43_{\pm 0.09}$	$0.64_{\pm 0.08}$✓	$0.63_{\pm 0.07}$	$0.19_{\pm 0.07}$	$0.53_{\pm 0.12}$	$0.33_{\pm 0.08}$	$0.34_{\pm 0.09}$	$0.56_{\pm 0.11}$	$0.42_{\pm 0.09}$
	ROCAUC	$0.79_{\pm 0.02}$	$0.73_{\pm 0.06}$	$0.75_{\pm 0.06}$	$0.82_{\pm 0.05}$✓	$0.42_{\pm 0.07}$	$0.68_{\pm 0.06}$	$0.62_{\pm 0.04}$	$0.44_{\pm 0.07}$	$0.66_{\pm 0.08}$	$0.65_{\pm 0.07}$
Ranking	Distance	$0.66_{\pm 0.39}$	$1_{\pm 0}$	$0.62_{\pm 0.39}$✓	$0.63_{\pm 0.36}$	$0.83_{\pm 0.24}$	$0.66_{\pm 0.37}$	$0.66_{\pm 0.39}$	$0.71_{\pm 0.39}$	$0.67_{\pm 0.39}$	$0.66_{\pm 0.39}$
	RBO	$0.84_{\pm 0.1}$	$0.89_{\pm 0.08}$	$0.84_{\pm 0.1}$	$0.84_{\pm 0.1}$	$0.84_{\pm 0.1}$	$0.89_{\pm 0.07}$	$0.88_{\pm 0.08}$	$0.93_{\pm 0.06}$✓	$0.91_{\pm 0.11}$	$0.88_{\pm 0.08}$
	Corr.	$0.2_{\pm 0.63}$	$0.42_{\pm 0.45}$	$0.37_{\pm 0.57}$	$0.43_{\pm 0.54}$✓	$-0.13_{\pm 0.71}$	$0.18_{\pm 0.64}$	$0.21_{\pm 0.67}$	$0.24_{\pm 0.69}$	$0.17_{\pm 0.68}$	$0.22_{\pm 0.66}$
Assistive Rank*		$8.00_{\pm 6}$	$3.66_{\pm 1}$	$1.33_{\pm 0.7}$ ✓	$2.33_{\pm 0.7}$	$41.33_{\pm 38}$	$32.00_{\pm 57}$	$24.00_{\pm 40}$	$70.67_{\pm 51}$	$47.33_{\pm 39}$	$5.33_{\pm 5}$

* 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 5: Summary of Algorithm Comparison with 60 Days Historical Data. ✓marks the best algorithm in each row.

identified the point without algorithmic assistance. We report the mean rank assigned to such points, where a smaller rank indicates that the algorithm would prioritize those points more for human inspection. We find that 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.

Overall, FlaSH's strength lies in leveraging specific features of public health data, a simple model class to meet deployment criteria, and an intuitive test statistic. The combination of these ideas is why FlaSH can scale to the data volumes 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.

6 Deployment and Lessons Learned

Based on FlaSH's empirical performance and design, it has been deployed as part of Delphi's daily workflow since February 2023. It runs on selected streams, and an expert reviewer inspects the ranked, outlying points. To support this interaction, we added a dashboard where expert reviewers can visualize each of FlaSH's calculations before flagging them. As new types of irregularities arise, an analyst in the loop can modify FlaSH to detect those respective outliers.

Lessons Learned. For outlier detection methods that produce actionable outputs, intuitive methods with informative outputs that explicitly navigate contextual nuances (like how FlaSH directly leverages the statistical properties of public health streams) innately enhance trust in method outputs that may also translate to performance gains. Additionally, method evaluations should consider expert-generated ground truth tasks that cover classification, because classification can be more straightforward for humans, and ranking, because thresholds for classification may vary.

7 Related Works

There are numerous outlier and anomaly detection methods [Blázquez-García *et al.*, 2021], but recent advancements in the field focus on deep learning applications [Pang *et al.*, 2021]. In our experiments, we find deep learning methods perform poorly on this task for various reasons. Accordingly, only a handful of real-time outlier streaming algorithms have been adapted for public health streaming data. Specifically, point outlier detection approaches for COVID-19 streams like [Jombart *et al.*, 2021; Karadayi *et al.*, 2020; Wang *et al.*, 2021; Agarwal *et al.*, 2022] consider the *non-stationarity* of the data streams but use simulations for evaluation or only consider a limited set of outlier categories. Hence, they are not fully applicable to our setting. Some source-specific COVID outlier detection methods [Dong *et al.*, 2022] that operate on data streams before Delphi receives them do not have publicly-available methods, but the continued presence of irregularities in those streams that impacts Delphi stakeholders underscores the importance of FlaSH.

8 Conclusion

This paper presents FlaSH, a practical framework for computationally assisted quality control in public health data streams. FlaSH creates a list of the most important outlying recent data points for domain experts to review by using simple models to explicitly account for the nuances of public health streaming data. In our experimental evaluation, which addressed some open design and evaluation challenges in unsupervised time series outlier detection, FlaSH scaled to the task requirements, outperformed other methods (including deep learning approaches) in traditional outlier detection metrics, and successfully prioritized points that would not have been discovered without algorithmic assistance. Our results demonstrate that effective, practical outlier detection systems require careful, user-informed design and sustained effort. These efforts will have considerable benefits for Delphi's stakeholders and, ultimately, for public health data users.

Ethical Statement

This research was conducted in accordance with the principles embodied in the Declaration of Helsinki and in accordance with local statutory requirements. All participants consented to the study and could exit the study at any time. Approval was granted from Carnegie Mellon Univeristy IRB number STUDY2022_00000240.

Acknowledgements

We want to thank the whole Delphi team and members of the Carnegie Mellon community for their input to and support of this project. 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". 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. This material is based upon work supported by the National Science Foundation Graduate Research Fellowship under Grant No. DGE1745016 and DGE2140739. Any opinion, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

[Agarwal *et al.*, 2022] Pulak Agarwal, Pranav Aluru, and B Aditya Prakash. Real-time anomaly detection in epidemic data streams. In *Epidemiology meets Data Mining and Knowledge discovery*, 2022.

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

[Angiulli and Pizzuti, 2002] 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.

[Blázquez-García *et al.*, 2021] 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.

[Breunig *et al.*, 2000] 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.

[CDC, 2022] US CDC. Covid-19 by county. https://www.cdc.gov/coronavirus/2019-ncov/your-health/covid-by-county.html, publisher=Centers for Disease Control and Prevention, Aug 2022.

[Chimmula *et al.*, 2020] Vinay Chimmula, Reddy Kumar, and Lei Zhang. Time series forecasting of covid-19 transmission in canada using lstm networks. *Chaos, Solitons & Fractals*, 135:109864, 2020.

[Cramer *et al.*, 2022] Estee Y Cramer, Yuxin Huang, Yijin Wang, Evan L Ray, Matthew Cornell, Johannes Bracher, Andrea Brennen, Alvaro J Castro Rivadeneira, Aaron Gerding, Katie House, et al. The united states covid-19 forecast hub dataset. *Scientific data*, 9(1):462, 2022.

[Dong *et al.*, 2022] 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*, 2022.

[Du *et al.*, 2017] 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.

[Farrow *et al.*, 2015] David C Farrow, Logan C Brooks, Aaron Rumack, Ryan J Tibshirani, and Roni Rosenfeld. Delphi epidata api. *The Lancet Infectious Diseases. https://github. com/cmu-delphi/delphi-epidata*, 2015.

[Hundman *et al.*, 2018] 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.

[Jombart *et al.*, 2021] Thibaut Jombart, Stéphane Ghozzi, Dirk Schumacher, Timothy J Taylor, Quentin J Leclerc, Mark Jit, Stefan Flasche, Felix Greaves, Tom Ward, Rosalind M Eggo, et al. Real-time monitoring of covid-19 dynamics using automated trend fitting and anomaly detection. *Philosophical Transactions of the Royal Society B*, 376(1829):20200266, 2021.

[Joshi *et al.*, 2023] Ananya Joshi, Kathryn Mazaitis, Roni Rosenfeld, and Bryan Wilder. Osf pre-data collection registration: Towards detecting points of interest from public health data streams. https://osf.io/2v8f5, 2023.

[Karadayi *et al.*, 2020] Yildiz Karadayi, Mehmet N Aydin, and Arif Selçuk Öğrencí. Unsupervised anomaly detection in multivariate spatio-temporal data using deep learning: early detection of covid-19 outbreak in italy. *Ieee Access*, 8:164155–164177, 2020.

[Killick *et al.*, 2012] Rebecca Killick, Paul Fearnhead, and Idris A Eckley. Optimal detection of changepoints with a linear computational cost. *Journal of the American Statistical Association*, 107(500):1590–1598, 2012.

[Kraemer *et al.*, 2021] 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.

[Kreps and Kriner, 2020] Sarah E Kreps and Douglas L Kriner. Model uncertainty, political contestation, and public trust in science: Evidence from the covid-19 pandemic. *Science advances*, 6(43):eabd4563, 2020.

[Lai *et al.*, 2021] Kwei-Herng Lai, Daochen Zha, Junjie Xu, Yue Zhao, Guanchu Wang, and Xia Hu. Revisiting time series outlier detection: Definitions and benchmarks. In *Thirty-fifth conference on neural information processing systems datasets and benchmarks track*, 2021.

[Leslie *et al.*, 2021] David Leslie, Anjali Mazumder, Aidan Peppin, Maria K Wolters, and Alexa Hagerty. Does “ai” stand for augmenting inequality in the era of covid-19 healthcare? *bmj*, 372, 2021.

[Liu *et al.*, 2008] 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.

[McDonald *et al.*, 2021] 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.

[Paleyes *et al.*, 2022] 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.

[Pang *et al.*, 2021] Guansong Pang, Chunhua Shen, Longbing Cao, and Anton Van Den Hengel. Deep learning for anomaly detection: A review. *ACM computing surveys (CSUR)*, 54(2):1–38, 2021.

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

[Reinhart *et al.*, 2021] 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.

[Sáez *et al.*, 2021] Carlos Sáez, Nekane Romero, J Alberto Conejero, and Juan M García-Gómez. Potential limitations in covid-19 machine learning due to data source variability: A case study in the ncov2019 dataset. *Journal of the American Medical Informatics Association*, 28(2):360–364, 2021.

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

[Simon, 2021] Sara Simon. Inconsistent reporting practices hampered our ability to analyze covid-19 data. here are three common problems we identified. *The COVID Tracking Project*, Apr 2021.

[Truong *et al.*, 2020] Charles Truong, Laurent Oudre, and Nicolas Vayatis. Selective review of offline change point detection methods. *Signal Processing*, 167:107299, 2020.

[Wang *et al.*, 2021] Guannan Wang, Zhiling Gu, Xinyi Li, Shan Yu, Myungjin Kim, Yueying Wang, Lei Gao, and Li Wang. Comparing and integrating us covid-19 data from multiple sources with anomaly detection and repairing. *Journal of Applied Statistics*, pages 1–27, 2021.

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

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

[Wu and Keogh, 2021] 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.

[Yu *et al.*, 2021] Shuo Yu, Qing Qing, Chen Zhang, Ahsan Shehzad, Giles Oatley, and Feng Xia. Data-driven decision-making in covid-19 response: A survey. *IEEE Transactions on Computational Social Systems*, 8(4):1016–1029, 2021.

[Zhu *et al.*, 2021] Di Zhu, Xinyue Ye, and Steven Manson. Revealing the spatial shifting pattern of covid-19 pandemic in the united states. *Scientific reports*, 11(1):1–9, 2021.