The Shape of View: An Alert System for Video Viewership Anomalies

Antonios Manousis^{*}, Harshil Shah[†], Henry Milner[†], Yan Li[†], Hui Zhang[†], Vyas Sekar^{*}

^{*}Carnegie Mellon University, [†]Conviva Inc

ABSTRACT

Internet video providers rely on alerting workflows to identify and remedy incidents that can impact users (e.g., outages or buggy players). There is growing evidence for the need for *viewership-based analytics*—detecting and diagnosing incidents that manifest through changes in viewership patterns but not in other (e.g., QoE) metrics. However, both detection and diagnosis of viewership anomalies is challenging due to the contextual nature of anomalies, non-stationarity of viewership, and complex dependencies between the structure of events and how they impact different subpopulations of viewers. We present PROTEAS, an alerting framework for video viewership anomalies that tackles these challenges. PROTEAS builds on key spatiotemporal *structural* insights. First, across different subpopulations of viewers and days of the week, we find that the *shape* of the viewership curve remains invariant over multiple weeks, thus enabling anomaly detection. Second, we use the hierarchy of viewership groups to produce compact alerts. Finally, we find that common anomalies manifest with spatiotemporal signatures, which enables us to classify anomalies to produce actionable alerts. We evaluate PROTEAS using 3 months of real viewership data (including the onset of the COVID-19 pandemic) and show that PROTEAS is accurate with over 80% True Positive Rate, average precision of over 86% (i.e., few false positives) and doesn't miss any major events. In addition, we find that approximately half of PROTEAS's alerts refer to events not caught by other alerting workflows, thus adding value to operators' existing toolkit.

CCS CONCEPTS

• Computing methodologies → Modeling methodologies;

ACM Reference Format:

Antonios Manousis, Harshil Shah, Henry Milner, Yan Li, Hui Zhang, Vyas Sekar. 2021. The Shape of View: An Alert System for Video Viewership Anomalies. In *ACM Internet Measurement Conference (IMC '21)*, November 2–4, 2021, Virtual Event, USA. ACM, New York, NY, USA, 16 pages. <https://doi.org/10.1145/3487552.3487819>

This work is licensed under a Creative Commons Attribution International 4.0 License.
IMC '21, November 2–4, 2021, Virtual Event,
USA © 2021 Copyright held by the owner/
author(s). ACM ISBN 978-1-4503-9129-0/21/11.
<https://doi.org/10.1145/3487552.3487819>

Figure 1: High-level view of an alerting framework for viewership anomalies. (Content provider logos are chosen randomly)

KEYWORDS

Performance; Measurement; Video; Anomaly Detection

1 INTRODUCTION

To sustain viewer engagement and maintain ad- and subscription-driven revenue streams [42, 58, 62], Internet video providers monitor the health of their infrastructure. This allows operators to detect and diagnose incidents (e.g., ISP outages, buggy players) [1, 19], and to inform mitigation efforts [52, 53].

Our conversations with analysts working in this domain highlight the need for *viewership-based analytics*,¹ to complement existing monitoring workflows and proactively identify anomalies that would be missed otherwise (e.g., based on monitoring QoE metrics like buffering time or startup latency [52, 53]). To see why, consider the following real-world incidents: AV encoding errors can cause pixelation or missing audio, resulting in viewers dropping out. Similarly, a platform failure, such as a provider's authentication system crash may result in viewers getting disconnected and viewership dropping. However, in both cases delivery will not be impacted, available QoE metrics will continue to look normal and the incident will go undetected. In addition, only considering QoE-based monitoring may not always provide actionable insights to operators. For example, increases in buffering time during live events can either be due to technical issues (e.g., a buggy player device) or due to a flash crowd that stresses the available CDN resources. However, it is hard to disambiguate these two scenarios just by looking at QoE metrics. In this case, flagging an unexpected viewership surge could alert operators for upcoming QoE issues.

To this end, we envision a real-time alerting framework (Figure 1) that *detects* and *diagnoses* such viewership anomalies. However, realizing such a framework is challenging on two main fronts which prevent us from directly using prior work from other domains (e.g., [22, 28, 31, 34, 55, 56, 69]):

¹Viewership is the timeseries of the count of active viewers.

1. **Contextual anomalies and non-stationarity:** A sudden drop/rise in the timeseries of viewership is not always anomalous; *e.g.*, when a popular show ends, viewership is expected to drop sharply and should *not* be marked as anomalous. Therefore, anomaly detection needs to consider the *context* in which a candidate anomaly appears [49]. While this contextual nature of anomalies is not unique to viewership, it suggests we use model-based techniques for detection [38]. Unfortunately, viewership is also non-stationary *i.e.*, the timeseries' statistical properties change over time, making it difficult to accurately capture them with a model.
2. **Need for compact and actionable alerts:** In practice, we want to monitor many *viewership groups i.e.*, subpopulations of viewers; *e.g.*, did viewers using RokuTV have an outage? Note, however, that a single logical event (*e.g.*, RokuTV outage) can simultaneously manifest as an anomaly across multiple viewership groups (*e.g.*, for NYC-RokuTV, NYC-Comcast, SF-RokuTV). Naively raising alerts for all these groups will overwhelm the analyst and complicate the diagnosis, thus raising the need for anomaly summarization. Unfortunately, techniques used in prior summarization and diagnosis works (*e.g.*, [22, 28, 31, 34, 53, 55, 56, 69]), while insightful, cannot account for the features of viewership that determine how a single incident manifests across multiple groups.

In this work, we address these challenges and present PROTEAS, an alerting framework for video viewership anomalies. Our design builds on the following *structural insights*:

1. **Shape persistence for anomaly detection:** Despite the non-stationarity of viewership, we find that its *underlying shape*² remains invariant over longer periods of time, and can thus be leveraged as a basis for detection. By modeling this key structural invariant using custom Gaussian Processes [73], we enable accurate, timely and robust anomaly detection of viewership anomalies across multiple viewership groups.
2. **Hierarchical group dependencies and spatiotemporal anomaly signatures for diagnosis:** A single logical event *propagates* predictably across viewership groups. Using practical heuristics, we extract the set of candidate groups that best explain an anomalous incident. Moreover, while viewership anomalies might be the result of many different root causes, we observe that anomalies resulting from similar incidents share common spatiotemporal features, allowing us to compile a library of *anomaly signatures*. This enables associating each viewership incident to a small set of possible root causes for further investigation.

We evaluate PROTEAS using real viewership data from 3 major content providers operating in Europe and USA. Our dataset spans a 13-week period (January-March 2020) and also includes the onset of the COVID-19 outbreak in these continents. Obtaining ground truth at this scale is a fundamental challenge and to this end, we corroborate our analysis through (1) manual verification of anomalies by experts of a large video analytics firm, (2) sentiment analysis of viewer behavior from Twitter, (3) cross-evaluation of detected

²By shape (§3), we refer to the structural form of the viewership timeseries after eliminating magnitude variations.

viewership anomalies with alerts on video-QoE metrics, (4) public holiday calendars and (5) public databases providing information on content providers' uptime.

The precision/recall analysis of our pilot study (§7) shows that under normal operating conditions PROTEAS issues a practical (for the analyst) number of alerts with low false positives (precision >86%) and practically no false negatives. PROTEAS outperforms the closest state-of-the-art alternatives by Twitter [49] and Netflix [18] both quantitatively in terms of precision and qualitatively in terms of trust shown by expert analysts to PROTEAS's alerts over those issued by prior work (>95%). Our synthetic analysis shows that the summarization component prunes out more than 99% of redundant anomalies and accurately identifies the root-cause viewership group by classifying anomalies in 4 broad but representative classes of events.

PROTEAS identified 3 large-scale technical outages that affected up to 50% of the content provider's viewers and were not caught by the existing alerting workflows of a major video analytics provider. PROTEAS tracked flash crowds during popular live events (*e.g.*, sports games) before the corresponding QoE alerts were raised, thus exposing the need for resource re-provisioning. Last, PROTEAS uncovered qualitative insights into the pervasive changes in viewership caused by COVID-19.

At the time of writing, PROTEAS's ideas are in the process of being deployed in production at a major video analytics provider.

2 MOTIVATION AND CHALLENGES

In this section, we highlight the need for viewership-based analytics through real-world incidents. Then, we present challenges in realizing this capability.

2.1 A case for viewership analytics

Monitoring a large-scale streaming video infrastructure requires deploying diverse analytics workflows that provide multiple complementary views of the infrastructure's operational status. In that context, our conversations with analysts at a large video analytics provider brought to light several real-world examples advocating for viewership-based analytics. The analysts observed that many incidents currently go undetected through existing QoE-based workflows. However, they also noted that because these incidents often impact user behavior, they may indirectly manifest as anomalies in viewership patterns. Thus, they observe an opportunity for viewership-based alerting to complement existing QoE-based workflows (*e.g.*, [2, 4, 9, 11, 17]).

We discuss a few such illustrative case studies:

1. **Platform failures:** An outage in a content provider's authentication service disconnected viewers and prevented them from logging back in for one hour. During this time, QoE metrics remained unchanged for the remaining viewers and as a result no QoE-based anomalies were raised. However, the viewership of the content provider indicated a sharp drop. For this incident, the provider was only informed about it through its Twitter support page!

Figure 2: Timeseries of viewership for the same day across 4 consecutive weeks. The anomaly in Week 4 is the result of a platform malfunction that went undetected. Real viewership values not shown.

2. **Encoding errors:** During the live broadcast of a popular annual US event, video encoding errors resulted in severe image pixelation. Despite the corrupted content, QoE metrics such as buffering or video start failures remained unaffected during the transmission and the incident went unnoticed until the next day when the provider looked at its unexpectedly low total viewership numbers.
3. **Insufficient streaming resources:** In high-stakes live transmissions, such as sports events, video QoE alerts (e.g., high buffering) are particularly common. These alerts, while indicating the presence of a problem, cannot often explain its underlying root-cause; e.g., is this an outage or a sudden flash crowd that puts unexpected load on streaming resources? Viewership-based analytics can enable operators to disambiguate and, thus take action minimizing the incident’s impact.

More concretely, Figure 2 illustrates an example of a large scale incident that manifested itself as a multi-hour long viewership drop. The incident affected viewers of a major US-based content provider on RokuTV and was attributed to platform malfunction. During that period, no QoE alerts were raised and the issue went undetected. Given that sustaining viewer engagement is one of operators’ overarching goals, this example clearly demonstrates that there is ample opportunity to *augment* existing QoE workflows by tracking video viewership for anomalies. Note that while providers do track long-term aggregate viewership trends, they do so mostly for content creation and provisioning but not for real-time anomaly detection [21], as envisaged here.

Definition of viewership anomalies: We define as a viewership anomaly any unexpected change (rise or drop) in the timeseries of viewership that might warrant technical intervention and appropriate mitigation efforts. In that context, sharp drops could be seen as indicators of a technical outage, whereas unexpected surges would be seen as i) a signal for resource re-provisioning or ii) an indicator of an upcoming technical disruption.

2.2 Challenges in Viewership Analytics

Operators conceptually use a 2-step monitoring workflow that consists of a *detection* and a *diagnosis* phase (Figure 4). During detection, the framework ingests data and uses them to detect anomalies either across the entire dataset or across different subpopulations of it (i.e., viewership groups). Intuitively, the metrics that are monitored scope the types of incidents the framework can detect. During diagnosis,

Attributes	Description
ISP	ISP viewers receive their traffic from.
Site	Content provider of requested video content.
City	City where the viewer is located.
CDN	CDN viewers receive their content from.
Player	App/Web portal to access content.
Device	e.g., iOS, AppleTV, RokuTV, HTML5 etc.
Protocol	Bitrate streaming protocol (e.g., DASH).
Live	Binary indicator of Live vs. VoD content.
Channel	Channel in the content provider’s network.

Table 1: Video session attributes.

the framework translates the detected anomalies into insightful alerts for the operator. Seen in a general context, detection and diagnosis are not new problems and have been examined in several prior efforts in networked systems (e.g., [22, 28, 31, 34, 55, 56, 69]) and in data mining (e.g., [23, 38, 59, 60, 71, 74, 75, 80]). However, as we will see below, video viewership raises a number of challenges that make it difficult to directly apply or extend these efforts.

Dataset: To highlight these challenges, we use a real-world dataset consisting of 13 weeks of viewership data from 3 major content providers $CP_{\{1,2,3\}}$, two being US-based and one national TV provider in northern Europe. For each content provider, our dataset contains ~350 million individual video sessions which are grouped into viewership groups based on the session attributes shown in Table 1. Overall, the dataset spans 6 large CDNs, 180 ISPs, 60 different cities, and 170 TV channels. §7 provides more details on our dataset and Appendix A discusses ethical considerations.

Detection Challenges: Viewership exhibits two key properties that raise important detection challenges:

Contextual anomalies: Viewership anomalies need to be seen in *context*. That is, in addition to detecting a statistically unexpected rise/fall in the signal, the analyst also needs to consider when that change appears in time i.e., its temporal context. To illustrate this, Figure 3 shows two 30-min episodes of the same popular weekly TV show. At the start and end of each episode, viewership exhibits sharp change points. As such, these changes are expected events and should not be flagged as anomalous. On the other hand, any other sharp drop/rise should likely be flagged as an anomaly. This observation renders seemingly intuitive detection strategies like change-point detection (e.g., [27]) insufficient. More generally, the contextual nature of viewership anomalies renders *model-free* anomaly detection techniques that only rely on short history windows for detection (e.g., control charts [80], EWMA [89], CUSUM [64] etc.) ineffective.

Lack of stationarity: The contextual nature of anomalies hints toward *model-based* anomaly detection techniques (e.g., exponential smoothing [39], autoregressive models [88], STL decomposition [32, 40, 77]) that model what expected viewership looks like and use that knowledge as a detection baseline. However, many model-based approaches rely on a *stationarity* assumption [65]. That is, they assume that the statistical properties of the signal do not change over time and, thus, historical data can be used as predictors of the future.

Figure 3: The first rise and drop indicate the beginning and end of show. Third drop (lower figure) is anomalous. Real values not shown.

Indeed, our experiments using stationarity tests *e.g.*, the Augmented Dickey Fuller test [45] showed that viewership is non-stationary due to its large temporal magnitude variability. For the same week-day, viewership values fluctuate greatly from week to week as the result of exogenous factors, such as content popularity, weather, competing shows *etc.* with differences in its magnitude being as high as 30%. Model-based techniques that assume stationarity in the underlying signal would naturally try to capture this variability thus resulting in inaccurate viewership baselines and false positives.

Diagnosis Challenges: There are two major challenges associated with diagnosis:

Anomaly Summarization: The same anomalous incident can redundantly manifest itself across overlapping viewership groups. To see why, consider the following real-world example where a major content provider rolled out a buggy firmware update for its RokuTV player which resulted in viewers on RokuTV (irrespective of geography) not being able to access any content. While the event should only raise an alert for the (RokuTV) viewership group, anomalies were detected at multiple other viewership groups including (i) finer partitions of the root-cause group (*e.g.*, <RokuTV, NYC>, <RokuTV, Comcast> *etc.*) and (ii) broader viewership groups such as <ALL_viewers> (given that RokuTV is a major player among streaming devices). Naively alerting on all such detected anomalies for all viewership groups is clearly impractical as it unnecessarily increases analysts’ efforts to diagnose the incident. Therefore, how do we efficiently *summarize* these alerts and identify the viewership group that best describes the incident? In §5.1, we discuss why seemingly intuitive approaches proposed in prior work for other domains (*e.g.*, ad-systems [34]) are not a natural fit for viewership.

Root-cause Attribution: Ideally, an insightful alert should contain pointers to the incident’s potential root causes to help analysts investigate the event. However, associating a viewership alert to candidate underlying events is not straightforward, *e.g.*, how can we disambiguate between a viewership drop that was the result of an outage or an encoding error? The complexity of the streaming ecosystem, the possible external factors that can trigger an anomaly and the lack of direct feedback from viewers as to why they stopped streaming complicates the task of root-cause attribution.

3 SYSTEM OVERVIEW

To fill this missing piece in the video analytics toolkit, we present PROTEAS, an alerting framework that detects and diagnoses viewership anomalies at scale. Figure 4 depicts a high-level overview of PROTEAS and its two key components. The detection component

Figure 4: A conceptual view of PROTEAS and its components.

leverages structural insights about viewership to build models for real-time detection at a viewership group granularity. The diagnosis component collects group-specific anomalies, summarizes them, and produces candidate diagnoses to inform further investigation.

Inputs: PROTEAS every minute receives batches of raw *video sessions*³ as input. It groups video sessions by all combinations of session attributes (Table 1) and for every resulting group it computes viewership as the count of the group’s sessions. For each group, PROTEAS maintains a short timeseries of recent viewership history (~1 hour) that the new observation gets appended to.

Detection component (§4): The detection component tracks viewership across groups and consists of two submodules. The offline submodule leverages a key structural invariant regarding the shape of the viewership curve to learn a shape-based detection model per viewership group. Each model is trained using summaries of the group’s historical viewership data and is fed into the real-time module that determines in near real-time (<10 minutes) whether a new observation is anomalous. To ensure that expected changes in viewership patterns are accounted for (*e.g.*, due to content changes) and that viewership baselines are always up-to-date, the offline component re-trains its models weekly. This module outputs a list of potentially anomalous viewership groups per minute.

Diagnosis component (§5): The diagnosis component consists of two sub-modules, summarization and root-cause attribution. Their goal is to provide the human analyst with actionable anomaly alerts consisting of the most informative summary of detection’s output. Given the list of anomalous groups from the detection step, PROTEAS’s summarization uses a set of heuristics that model the propagation paths of a single anomalous incident across groups to prune out redundant anomalies. The group that best explains the event is read by root-cause attribution whose goal is to identify likely root-causes. To that end, root-cause attribution relies on a library of anomaly signatures, (that is periodically updated as more signatures are identified) and matches the observed new event to one or more of these signatures.

4 SHAPE- BASED ANOMALY DETECTION

We now introduce the domain-specific observations about viewership that drive PROTEAS’s detection mechanism. We summarize the notation used throughout this section in Table 2.

³A video session represents a user consuming one piece of content. Each session describes the content across a number of attributes (Table 1) and also contains real-time measurements of video QoE.

Notation	Definition
(g, w, d)	(viewership group, weekday, date)
T	Set of all timestamps in one day
D	Set of all dates
$v_{g,w}^d$	Timeseries of observed viewership for (g, w, d)
$s_{g,w}^d$	Timeseries of the shape of viewership for $v_{g,w}^d$
$\mu_{g,w}^d$	Mean of timeseries of $v_{g,w}^d$
$\sigma_{g,w}^d$	Standard deviation of $v_{g,w}^d$
$\mathcal{M}_{g,w}$	Prediction model trained on $s_{g,w}^d, \forall d \in D$
$\hat{s}_{g,w}(t)$	Estimate of shape of viewership for $t \in T$
$\hat{u}_{g,w}(t)$	Confidence bound on $\hat{s}_{g,w}(t)$
$S_{g,w}(t)$	Random variable for shape of viewership at time $t \in T$
X	Set of timestamps where viewership is observed
Y	Set of timestamps where viewership is unknown
μ	Mean of the joint distribution $S_{g,w}(T)$
Σ	Covariance matrix of $S_{g,w}(T)$

Table 2: PROTEAS Notation. First, we introduce general notation for viewership. Then we introduce terms related to the general prediction model and, last, notation specific to the formulation of Gaussian Processes.

4.1 High-level Insight: Shape Invariance

Our insight to tackle the contextual and non-stationary nature of viewership modeling is that the *shape* of the viewership curve (formally defined below) is consistent over multiple weeks. Figures 3 and 5 visually suggest that despite the variability in magnitude for two viewership groups, the *shape* of each group is consistent across weeks.

Figure 5: Intuition on the invariability of shape of viewership.

While this visual result is appealing, a natural question is whether this observation holds more generally across groups, weekdays, and content providers. To this end, we compute the pairwise structural similarity of the viewership timeseries for all (group, weekday) tuples through their *cosine distance*.⁴ Figure 6 shows these results for two content providers and a testing period of 3 months. Indeed, we see that (i) when the two curves are from the same group and weekday, the cosine distance is both low and minimal and (ii) this observation does not necessarily hold for any random choice of groups.

Problem Formulation: The above observation leads to a shape-based detection workflow described below (Figure 7).

⁴The cosine distance of two vectors quantifies structural similarity through their angular difference. A value of 0 indicates a perfect match [5].

Figure 6: Viewership curves of the same group and weekday exhibit high structural similarity, unlike unrelated pairs of viewership curves.

Figure 7: Shape, Training Observations and $\mathcal{M}_{g,w}$.

Let g indicate a viewership group and $v_{g,w}^d$ be its timeseries of viewership on weekday w (e.g., Monday) and date d (e.g., March 23). Let T be the set of all timestamps in one day and D the set of all dates. $s_{g,w}^d$ is a timeseries of the *shape* of viewership derived from $v_{g,w}^d$ after eliminating magnitude variability for that day:

$$s_{g,w}^d(t) = \mathcal{Z}^d(v_{g,w}^d(t)) = \frac{v_{g,w}^d(t) - \mu_{g,w}^d}{\sigma_{g,w}^d} \quad (1)$$

Let $\mathcal{M}_{g,w}$ denote our prediction model trained using *summaries* of historical observations of $s_{g,w}^d, \forall d \in D$ (Figure 7). $\mathcal{M}_{g,w}$ takes as input a timestamp t and outputs the corresponding estimate $\hat{s}_{g,w}(t)$ along with a confidence bound $\hat{u}_{g,w}(t)$ as shown in (2):

$$\mathcal{M}_{g,w}(t) = \langle \hat{s}_{g,w}(t), \hat{u}_{g,w}(t) \rangle, \quad \forall t \in T. \quad (2)$$

Having $\mathcal{M}_{g,w}$, we can determine whether viewership $v_{g,w}^d(t')$ is anomalous. First, we compute $s_{g,w}^d(t')$ using (1). In an offline setup where $v_{g,w}^d$ is apriori known, computing $s_{g,w}^d(t')$ is straightforward because $\mu_{g,w}^d$ and $\sigma_{g,w}^d$ are known. However, in the online scenario where $\mu_{g,w}^d$ and $\sigma_{g,w}^d$ are unknown, they need to be approximated (§4.4). Ultimately, $s_{g,w}^d(t')$ is marked as anomalous if it doesn't lie within the interval defined by $\hat{s}_{g,w}(t') \pm c \cdot \hat{u}_{g,w}(t')$, where c is a parameter determining the width of the confidence bound [3]. Generally, we set $c = 3$ for a confidence bound of $\sim 99\%$. In §7, we provide a brief discussion on detection sensitivity as a function of c .

Given this formulation, ideally $\mathcal{M}_{g,w}$ should be: (1) *General* to model a large (and a priori unknown) number of shapes; (2) *Automated* and require minimal manual tuning to handle 100k+ viewership groups; (3) *Robust* to outliers in historical data and (4) *Compact* in terms of space/time efficiency of storing and using models.

4.2 A case for using Gaussian Processes

	General	Automated	Robust	Compact
Decomposition	N	N	Y	Y
Pointwise Estimation	Y	Y	N	N
Vanilla GP	?	Y	?	Y
PROTEAS GP	Y	Y	Y	Y

Table 3: Strawman solutions vs. PROTEAS’s detection approach.

Unfortunately, meeting all of these requirements is challenging (Table 3). To see why, consider two candidate solutions. One class of work uses basis functions (e.g., polynomials [35], wavelets [26], redundant dictionaries [25]) or PCA (as done by Netflix’s Surus [18]) to decompose and compactly model traffic timeseries using a few coefficients. However, it is intractable to manually identify and tune the basis set for a large number of diverse groups. On the other hand, we can use historical traces to compute an empirical pointwise mean-variance for each (d, w, t) combination. Unfortunately, this is not resilient to outliers as a single surge or missing data observation can skew our model. Furthermore, maintaining these point estimates for a large number of groups is not compact. For instance, ARIMA models are not robust to outliers and are not compact as they require long history windows [88]. The same compactness argument applies to LSTM neural networks and other deep learning-based approaches [46, 83].

Why Gaussian Processes: Rather than look for a universal basis function, we revisit the idea of *stochastic Gaussian Processes* [47, 73]. Conceptually, a Gaussian Process (GP) is a probability distribution over all possible shape functions. Due to this non-parametric nature, GPs can model *any shape with minimal apriori structural assumptions*. GPs enable simple, robust, and automated learning procedures that eliminate the need for manual tuning [73].

Primer on GP: Figure 7 visualizes a GP-based detection model to learn $M_{g,w}$ from history. The GP models both the underlying shape and the uncertainty such that, for any unknown timestamp t' , we can produce (through interpolation) an $\hat{s}_{g,w}(t)$ and appropriate error bounds. We refer readers to [47, 73] for more details beyond this high-level primer.

The key assumption behind GPs is that each observation $s_{g,w}^d(t)$ is a *sample* drawn from a normally distributed random variable $S_{g,w}(t)^5$. The entire timeseries $s_{g,w}^d$ then can be seen as a sample drawn from the joint (Gaussian) distribution of the set of random variables $S_{g,w}(T) := \{S_{g,w}(t)\}_t, \forall t \in T$.

Let $X \subseteq T$ be the set of all timestamps where viewership is observed and Y the set of timestamps where viewership is unknown. We let $S_{g,w}(X)$ and $S_{g,w}(Y)$ be the vectors of all random variables that correspond to timestamps in X and Y respectively. The joint distribution $S_{g,w}(T)$ then becomes:

$$S_{g,w}(T) := \begin{bmatrix} S_{g,w}(X) \\ S_{g,w}(Y) \end{bmatrix} \sim \mathcal{N}(\mu, \Sigma), \quad \text{where} \quad (3)$$

⁵Making a normality assumption is not unreasonable. Recent work suggests using a student distribution but the performance benefits are marginal [84].

μ is the mean vector of the distribution and Σ its covariance matrix. For simplicity, μ is commonly set to 0. Given (3), making point estimates about $\hat{s}_{g,w}(t')$, $t' \in Y$ reduces to computing the conditional expectation $\mathbb{E}[S_{g,w}(Y) | S_{g,w}(X), X, Y]$ at $t' \in Y$. Then, $\hat{u}_{g,w}(t')$ becomes the variance of the conditional distribution at t' .

4.3 Using Gaussian Processes in PROTEAS

Given this background on GP, two questions remain:

1. **Modeling $S_{g,w}$ with time-dependent noise:** Traditional GP assumes that their output is either noise-free or the noise is time-independent. In viewership, however, noise levels depend on the time of day, which needs to be considered when modeling $S_{g,w}$, without breaking the theoretical guarantees of GPs.
2. **Modeling covariance Σ :** The covariance matrix Σ is at the heart of a GP as it encodes the impact that neighboring observations have on viewership estimates at time t' [73]. To learn Σ , GPs commonly use appropriate *kernels* $\kappa(t, t', \theta)$ i.e., covariance functions that take as input a set of hyperparameters θ , timestamps t, t' and return the corresponding covariance between $S_{g,w}(t), S_{g,w}(t')$. Thus, modeling $M_{g,w}$ boils down to a suitable choice of kernel(s) and optimizing their hyperparameters θ .

We discuss how we address these two key issues in PROTEAS.

Extending GP to support time-dependent variance: We incorporate noise in our model using the intuitive definition of the shape of viewership. We define $S_{g,w}$ as follows:

$$S_{g,w}(t) := \bar{S}_{g,w}(t) + \bar{\epsilon}_{g,w}(t) \quad \forall t \in T, \quad \text{where} \quad (4)$$

$$\bar{S}_{g,w}(t) \sim \mathcal{N}\left(\frac{1}{|D|} \sum_i s_{g,w}^{d_i}(t), 0\right) \quad \text{and} \quad \bar{\epsilon}_{g,w}(t) \sim \mathcal{N}(0, \sigma_t^2)$$

Specifically, each latent random variable $S_{g,w}(t)$, $t \in X$ has a normal distribution whose mean is the mean of historical shape observations at time t and a noise component σ_t^2 equal to the standard deviation of $s_{g,w}^d(t)$, $\forall d \in D$.

Choice of Covariance Kernel: Kernels intuitively represent families of functions (e.g., periodic, differentiable etc.) and allow us to incorporate prior knowledge on the shape of viewership.⁶ We identify the following structural characteristics that $\kappa(t, t', \theta)$ should account for; (1) *Decaying periodicity*: Viewership often exhibits a periodic structure with values dropping in early morning in order to rise again during prime time hours; (2) *Lack of smoothness*: Viewership exhibits sharp change points (e.g., at the beginning or end of a show) that need to be included; and (3) *Time-dependent noise*. To address (1), we combine a periodic and a squared exponential kernel [12, 15]. To capture (2), we add non-smoothness with a Matérn kernel [10]. Finally, for (3) we use a white-noise kernel, tweaked to account for time-dependent noise [20].

Kernels can be combined together through addition or multiplication [73]. The final version of our kernel is as shown below and a detailed mathematical analysis can be found in Appendix B. After

⁶We note that the choice of kernel is a much less restrictive process in terms of prior assumptions compared to strawman solutions [43].

Figure 8: Lattice of session groups

a choice of kernel has been made by the analyst, a fully automated fitting process is carried out to optimize its hyperparameters θ [73].

$$\kappa M_{g,w} = \kappa_{periodic} * \kappa_{sqExp} + \kappa_{Matern} + \kappa_w Noise \quad (5)$$

4.4 Online detection

Having learnt $M_{g,w}$, the next step is to use (1) to detect in real-time whether a newly observed viewership value is anomalous or not. However, recall that the standardization operator Z^d is date-dependent, *i.e.*, it requires knowledge of the (unknown) mean $\mu_{g,w}^d$ and the standard deviation $\sigma_{g,w}^d$ for the current date d , thus raising our next challenge.

To address this issue, we resort to a simple iterative algorithm for approximating $Z_{g,w}^d$ for the current date. Our approach relies on the intuition that we can begin with an approximation of $\mu_{g,w}^d$ and $\sigma_{g,w}^d$ estimated with historical data of the most recent past observation for that weekday (*e.g.*, data from the past week) and refine these estimates as new observations of the current date arrive. More specifically, we maintain a very sparse representation of historical observations of raw viewership (*e.g.*, sampled every 30 minutes for a total of 48 samples). Every 30 minutes we replace one stale observation with the corresponding most recent viewership observation of the current date. This heuristic enables us to compute an estimate of Z^d that quickly converges to the optimal value.

5 GENERATING ACTIONABLE ALERTS

Next, we present the PROTEAS’s diagnosis module. This module takes as inputs the set of group-specific viewership anomalies and provides: (1) a summarized alert consisting of the top- k viewership groups that best explain the incident and (2) a small set of likely root causes.

5.1 Summarizing viewership anomalies

As discussed in §2, an anomalous incident often manifests itself redundantly across multiple viewership groups. Naively issuing an alert for every such anomaly increases the analysts’ workload for diagnosis and degrades the overall usability of the framework. For instance, in the case of the buggy firmware update for RokuTV discussed in §2, the resulting outage (that was only caught through viewership) manifested at more than 100 different viewership groups. Naively raising 100 redundant alerts would compromise PROTEAS’s usability.

Thus, we need a mechanism that identifies the *critical* viewership group(s) that can best summarize the detection anomalies. To see

why this is challenging, consider two strategies from prior work [34, 52, 53].

1. **Summarize by group succinctness** [53]: Jiang *et al.* in their work on video QoE suggest that an anomaly is best explained by the most *succinct*⁷ session group whose anomalous sessions, when removed, result in the remaining groups becoming “healthy” (*i.e.*, fraction of anomalous sessions drops below a threshold). Quantifying a group’s “health” through its fraction of anomalous sessions means we can freely remove subsets of sessions and measure the resulting changes in that group’s health. In PROTEAS, however, such an approach would alter the baseline shape of a group and would introduce additional complexity.
2. **Summarize by “surprise”** [34]: Prior work by Bhagwan *et al.* in summarizing anomalies in ad-systems argues that succinctness alone is insufficient and reports “surprising” data partitions that exhibit the greatest relative change with respect to its expected value. This is built upon a domain-specific observation about ad systems, that anomalies typically appear in coarse one-dimensional groups (*e.g.*, at ISP or CDN level). However, this does not hold in our scenario as anomalies can originate both in coarser (*e.g.*, viewership in SF drops) and in finer groups (*e.g.*, NYC-based viewers on Comcast using RokuTV have an outage).

While these strawman solutions do not directly apply in our context, we can learn from them. From Jiang *et al.* we observe that video incidents manifest in a natural *hierarchical structure* across the space of session attributes, which also applies to viewership (Figure 8). From Bhagwan *et al.* we can build on the insight of finding a small/sparse set of root causes that have most *explanatory power*. Yet, it is not clear how to define the explanatory power of a group in our context. To that end, we build on the following two observations: First, it is rare for multiple independent anomalous incidents to occur simultaneously. That is, the set of underlying root-causes is *sparse*. This intuitively suggests that most (if not all) anomalous events can be traced back to a few critical groups. Second, anomaly *propagation* follows two natural structural rules:

1. **Propagation to descending nodes:** For each group g , each of its descendants accounts for a fixed share of g ’s viewership. When an anomalous incident originates at g , each descendant will experience a viewership change proportional to their share of g ’s viewership. By the same token, the change propagates recursively to all descendants in the hierarchy. In the (RokuTV) case, the anomaly also appeared at (RokuTV, Comcast), (RokuTV, Verizon), and all members of the finer (Player, ISP) set.
2. **Propagation to parent nodes:** The viewership of g ’s parent is the sum of g ’s viewership plus that of the parent’s other descendants with whom g shares the same attributes⁸. As a result, when an anomalous incident originates at g , the parent node experiences a change in viewership nominally equal to g ’s. That change, however, will be less pronounced as the

⁷in terms of number of attributes

⁸If we partition the direct descendants of each node by session attributes, the sum of viewership within each resulting partition equals the viewership of the node.

relative change of viewership is smaller in the larger group. By the same token, the change will propagate recursively up until the root.

Using the above rules, our strategy for summarization boils down to quantifying for each anomalous group g its *Explanatory Power* (EP_g), a score that quantifies how well group g explains the manifestation of the incident in the remaining anomalous groups. We use the following heuristics to estimate EP_g ;

1. **Range of impact:** If g explains the anomalous incident, the incident should (i) simultaneously manifest in g 's direct descendants and (ii) the cumulative change in traffic of the anomalous descendants, grouped by combination of attributes AC , is approximately equal to the change in g 's viewership.

Intuitively, given the first propagation rule, the fewer anomalous descendants g has, the lower its explanatory power should be. We quantify this quantity as follows:

$$H_{g,1} = \min\left\{\frac{\#anomDescendants}{\#allDescendants}\right\}_{AC} \quad (6)$$

2. **Surprise:** If g explains the anomalous incident, its change in viewership *i.e.*, the difference between the predicted viewership value and the observed one, relative to other anomalous groups should be large. This is inspired by the notion of “surprise” introduced in prior work [34]. We quantify this measure of change as:

$$H_{g,2} = \frac{|\widehat{V}_g - V_g^{obs}|}{\max\{|\widehat{V}_{g'} - V_{g'}^{obs}|\}} \quad (7)$$

Given these heuristics, we formally define the EP_g as $EP_g = H_{g,1} \times H_{g,2}$, where a higher value of EP_g indicates a better ability to explain the anomaly. Our heuristic computes this for all groups and outputs the top- k groups. Since our goal is to identify a suitable starting point for further investigation, we believe that releasing the top- k groups is a more realistic approach than releasing only the top group. As we will see in §7, in practice, the true answer lies within a small k (< 3).

5.2 Root-cause suggestions

The final step is to annotate the summarized incident with candidate root causes. Since many factors are invisible to PROTEAS (*e.g.*, ISP upgrades), our goal in practice is not exact root-cause attribution but to reduce the analyst's overhead by providing a set of candidates for further investigation.

Our high-level intuition is that each anomalous event can be associated with a set of *anomaly signatures*. Consequently, we can match a new anomaly against a library of signatures learned from history to determine (in real time) a set of candidate root-causes. For instance, in the RokuTV example, the intrinsic properties of the anomaly (*e.g.*, its shape, duration, start time, the critical viewership group *etc.*) potentially indicate that the underlying root-cause is an outage. Based on our data-driven insights, we see that this signature has two facets: (1) a *spatial* dimension capturing the groups that have been impacted by the anomaly and their relationships in the group hierarchy; and (ii) a *temporal* dimension such as its duration or its views-per-time pattern.

Spatial dimension of anomaly signatures: An intuitive way to capture the spatial component is to enumerate the anomalous viewership groups. For instance, for our RokuTV outage, we identify all the related anomalous groups. However, this approach cannot generalize to other occurrences of similar events; *e.g.*, if FireTV manifested the same anomaly, we cannot match the signature. Thus, we extend this basic idea and represent the spatial characteristics of an anomaly signature at an *attribute-level* granularity (*i.e.*, Device and descendants), rather than specific groups (*i.e.*, RokuTV). To this end, we create a hierarchy of attribute groups and mark the nodes that contain anomalous groups. Matching a new anomaly to an anomaly signature is a simple comparison of the fraction of shared nodes.

Temporal dimension of signatures: In practice, we find that the shape of the anomaly remains largely consistent across affected groups. This has two implications with respect to the anomaly's temporal signature. First, to isolate a signature's temporal component, it suffices to characterize one of the affected groups. Second, matching an anomaly to our library, requires a timeseries similarity metric.

However, there are two challenges. The detection time of an anomaly does not necessarily coincide with the incident's start time as viewership might have been anomalous for a while before we flag it. Thus, to extract the temporal pattern of the anomaly, we also include a short window of viewership history (empirically set to 15 mins) to ensure the onset of the incident is not missed. Second, two similar events with the same temporal manifestation may still differ in duration, suggesting the use of Dynamic Time Warping (DTW) [33] for comparisons.

Building a library of signatures: Given this approach, we explain the workflow for bootstrapping and maintaining a library for associating anomalous events with their respective signatures. To bootstrap the library, we first generate “raw” signatures as described above corresponding to different anomalies. Then, we systematically group together possibly similar signatures, and ultimately determine whether there is a representative pattern that can be associated with an anomalous event. To this end, we apply DTW across all pairs of “raw” signatures and then apply a clustering algorithm on the resulting distances. Updating the library, however, requires some analyst intervention. For every new event that does not map to a known signature from our library, we defer to the analyst's forensics investigation to diagnose the event and add its signature to the library.

Our signature library consists of 4 broad categories of incidents; (1) *Technical issues* for alerts warranting technical intervention; (2) *Flash crowds*, encompassing short-lasting viewership surges, scoped by the duration of the piece of content they manifest in; (3) *Model Drift* for changes in viewership being the result of change in offered content or viewing habits; (4) *Measurement errors*, which included short lasting drops due to the potential loss of session batches or system updates *etc.*

6 IMPLEMENTATION

We implement PROTEAS on Apache Spark [87]. PROTEAS receives batches of raw video sessions every minute which are then grouped by combinations of session attributes.

Detection: There are two practical issues. First, with detection running every minute for tens of thousands of groups per cluster node, we need compact models to ensure minimal storing and fetching cost. Second, to capture model drift, PROTEAS needs to periodically retrain models.

Given our GP-based approach (§4), there is natural tradeoff between model fidelity and the temporal granularity (*i.e.*, size of the set of observed timestamps X). Intuitively, if X incorporates more prior observations, the model fidelity will be higher. However, larger X implies higher overhead to store necessary parameters. Empirically, we achieve a reasonable tradeoff when timestamps in X are sampled at 5-minute intervals ($\sim 5KB$ per group).

Any learning approach in practice has to tackle *model drift*; *e.g.*, content popularity, or viewership habits change over time. To that end, PROTEAS performs an offline retraining of viewership models daily (of the subset of models corresponding to that weekday). Specifically, for each timestamp in X , we recompute its summary statistics to incorporate the most recent observations of the shape of viewership. To prioritize recent observations over older ones, we estimate summary statistics using exponentially weighted moving averages and standard deviations and observe that re-training is lightweight (<20 sec per group).

Diagnosis: The diagnosis process begins the moment an anomaly is first detected. We iteratively compare the current anomaly with entries in the library. For some events, our confidence increases as the anomaly progresses over time. For other abrupt events (*e.g.*, ISP outage), the onset of the anomaly is often sufficient in order for PROTEAS to reach an accurate diagnosis soon (*e.g.*, within 10 minutes of the onset). When we do not find a reasonable match in the library, PROTEAS informs the operator of the need to start a manual root-cause attribution analysis.

7 EVALUATION

In this section, we show that PROTEAS is: (1) *accurate* with a mean True Positive Rate (TPR) $>86\%$ in the common case; (2) *adds value* to operators, with $\sim 50\%$ of its alerts being distinct from alerts by other QoE-tracking tools; (3) provides *useful alerts*, within minutes after the incident’s onset; and (4) is *usable* as it produces a manageable volume of daily alerts (up to 3 alerts per incident), thus preventing alert fatigue. We compare PROTEAS with the two closest state-of-the-art alerting frameworks; (1) Twitter’s offline statistical anomaly detector [49] and (2) PCA-based Surus by Netflix [18] and show that PROTEAS is more accurate than prior work and that analysts in a blind survey trust PROTEAS’s alerts over those of prior work $\sim 93\%$ of the time.

7.1 Methodology

We begin with a real-world study to assess the system’s overall value (§7.2) and complement it with synthetic, trace-driven sensitivity analysis of PROTEAS and its components (§7.4).

Setup: Table 4 summarizes key features of our real-world dataset. Our dataset comes from three content providers offering different types of service; CP_1 mostly offers VoD content across three channels. CP_2 is a Linear TV provider *i.e.*, it combines scheduled content and scheduled live transmissions under one channel. Finally, CP_3 combines VoD, scheduled content and live transmissions under 164 (regional and US-wide) channels.

Feature	CP_1	CP_2	CP_3
Dataset	January - March 2020		
Duration of Pilot	March 2020 (4 weeks)		
Geography	USA	EU	USA
Session Attributes	7	7	9
Viewership Groups (Average)	3000	650	5500
Channels	3	1	164
Live/VoD ratio (Average)	0	0.2	10

Table 4: PROTEAS’s real-world dataset

Ground Truth: A key challenge in this domain is the lack of ground truth for real-world anomalies. This challenge is not unique to our work and prior works attempt to address this issue through manual data labelling [34, 49, 84]. For PROTEAS, in addition to expert analysis, we leverage complementary sources of ground truth to augment this standard approach and to reduce potential biases. Specifically:

1. Twitter-based Sentiment Analysis: During an outage, viewers often resort to Twitter to express annoyance or to reach out to operators. We run a simple sentiment analysis to expose viewer reactions that allow us to correlate a viewership alert with its ground truth [70]. To reduce the risk of diluting our analysis with unrelated tweets (*e.g.*, personal opinions about quality of show *etc.*), we limit this analysis to providers’ public help-desk Twitter profiles. We do acknowledge the potential bias toward users who are more active on Twitter.
2. Public Downtime Records: We collect uptime reports for Internet services from public databases [6]. Given that video outages can result from many other related services (ISPs, CDNs, content provider portals *etc.*), we collect downtime reports for all of these services.
3. Video QoE Alerts: Our third source of ground truth consists of video QoE alerts that the analytics provider runs already. Given that some QoE issues can lead to viewership drops (as viewers quit their sessions due to poor experience), this can help validate some viewership alerts.
4. Calendar of Holidays and Major Events: Days like Thanksgiving are singular events and tracking these enables interpreting PROTEAS’s alerts accordingly.
5. Blind Review by Expert: Finally, we leverage the expertise of 4 video analysts to confirm the presence of an anomaly. To minimize biases in the labeling process, we set up a 2-phase survey. First, the analyst is presented with the unlabeled timeseries of the date of interest as well as recent timeseries corresponding to the same weekday and is asked to identify anomalous segments (if any). Then, the analyst is presented with the alerts

issued by PROTEAS and the prior works we compare against (§7.2) on that timeseries and is asked to mark the outputs they agree with (again, if any). The events of the second phase of the survey were chosen and presented randomly to ensure there was no association with the data of the first phase.

Labelling Strategy: We compile the ground truth as the union set of incidents from the above sources and check whether a detected anomaly matches with any of the anomalies in this set. If yes, we mark it as a True Positive (TP). Otherwise, we conservatively mark it as a False Positive (FP). For anomalies in the ground truth set that do not match with any of PROTEAS’s anomalies, we manually confirm the presence of a False Negative (FN). This is because an anomaly in a different context (e.g., QoE metrics) might not necessarily translate to a viewership anomaly. Naturally, we acknowledge that our labelling strategy does not provide us with the perfect ground truth, but only with a subset of it.

Evaluation with synthetic trace: To complement the pilot study, we also evaluate PROTEAS with synthetic viewership data. Specifically, we leverage our understanding of viewership anomalies from the real-world dataset and build an anomaly generator that enables injecting custom anomaly signatures on healthy segments of the original viewership trace. This allows us to run a synthetic analysis with perfect knowledge of ground truth.

Accuracy Metrics: To evaluate the accuracy of PROTEAS (and of the baselines), we measure its TP, FP and FN counts. In addition, we also compute the frameworks’ precision ($TP/(TP+FP)$). For our synthetic analysis in §7.4, we also measure recall ($TP/(TP+FN)$) and F-score [8, 13].

Existing QoE alerting workflows: Existing QoE workflows rely on four quality metrics that are commonly used across content providers and are known to be critical in measuring video QoE as well as engagement [42, 53]. These metrics are: (i) **Buffering ratio** i.e., the ratio of seconds a player spends buffering by the total duration of the session; (ii) **Join time** i.e., the time needed for a player to start streaming content after the user clicks ‘play’; (iii) **Average bitrate** i.e., the average of bitrate values (given the prevalent use of adaptive bitrate algorithms; (iv) **Join failures** i.e., a boolean value indicating whether a session failed to start. Detecting anomalies across these QoE metrics is generally more straightforward than detecting viewership anomalies. That is because the values each of these metrics should take lie in a relatively small range and, as a result, often anomaly detection is as simple as setting static or slowly changing upper/lower thresholds.

PROTEAS’s detection sensitivity: As discussed in §4.1, an observation of viewership $s_{g,w}^d(t')$ is marked as anomalous if it doesn’t lie within the interval defined by $\hat{s}_{g,w}(t') \pm c \cdot \hat{u}_{g,w}(t')$. Note that c is a configurable parameter by the operator that determines the width of the confidence bounds (green shaded region in Figure 7). We set $c = 3$ i.e., we set the confidence bounds to the standard of 3 standard deviations from the mean predicted value. A smaller value of c will increase the sensitivity of detection and may result in either increased numbers of false positive alerts or true positives of longer duration. A higher value of c would result in the opposite behavior i.e., missed true positives. Note we apply the same sensitivity configuration across baselines throughout the evaluation.

7.2 Pilot Study: End-to-End Evaluation

	Metric	W1	W2	W3	W4
CP ₁	Alerts	8	10	136	138
	False Positives	1	1	22	50
	Precision (%)	87.5%	90.9%	85.5%	73.4%
CP ₂	Alerts	25	23	63	59
	False Positives	3	4	18	13
	Precision (%)	89.2%	85.1%	77.9%	81.9%
CP ₃	Alerts	85	92	165	211
	False Positives	17	15	42	85
	Precision	83.3%	85.9%	79.7%	71.2%

Table 5: PROTEAS’s key accuracy metrics. We observe that PROTEAS issues a manageable number of alerts, with few FP and no FN. W₃ and W₄ reflect the onset of COVID-19.

Table 5 summarizes PROTEAS’s key accuracy metrics per testing week (W_x) and content provider (CP_y). W₁ and W₂ reflect PROTEAS’s expected operational scenario. W₃ and W₄ correspond to onset of the shelter-in-place period due to COVID-19 in USA, thus a period of fundamentally altered viewership patterns. Table 5 shows that in the common case, the number of *weekly* alerts issued by PROTEAS is low. CP₁ is a big VoD provider with persistent viewership patterns and incurs the fewest alerts (<10 per week). CP₂ and CP₃ raise higher numbers of alerts in the common case. This is due to their higher percentage of live content offered (which can be more unpredictable in terms of viewership). In addition, in the case of CP₃, alerts are most often traced back to the provider’s many channels (as shown in Figure 11). Indeed, CP₃ offers a high number of channels that (by design) are not grouped together during summarization, which results in higher numbers of True Positives. Nevertheless, across all content providers, we can say that the number of alerts raised results in a manageable workload for human analysts.

We also observe that PROTEAS’s precision is consistently >85% in the expected operational region indicating that PROTEAS incurs few false positives. During the pandemic period, however, we see precision dropping slightly as a result of pervasive changes in viewership behaviors. Regarding FN, we find that PROTEAS does not miss any major incidents.

Comparison with prior work: We compare PROTEAS with (1) Twitter’s offline statistical anomaly detector [49] and (2) PCA-based Surus by Netflix [18]. These prior works only perform anomaly detection, so we combine their outputs with PROTEAS’s summarization routine. Figure 9 shows the distribution of their daily precision values and shows that PROTEAS consistently outperforms prior work. In the cases of CP₁ and CP₂ PROTEAS’s median precision is 88% and 90% respectively whereas for CP₃ it is 79%. On the other hand, Twitter’s median precision for the three providers equals to 68.5%, 78% and 65% respectively, and for Netflix’s Surus these values become 61%, 71% and 59%. Further investigation of these precision gaps indicates that while prior work exhibits similar true positives as PROTEAS, it produces higher numbers of false positives (all frameworks have the same nominal sensitivity).

Figure 9: Comparison of end-to-end precision analysis.

Figure 10: Breakdown of True Positives by source of ground truth.

Trust in PROTEAS’s Alerts: In addition to precision, we measure the “trust” our 4 experts show to each framework’s alerts. Recall that as part of ground truth extraction (§7.1), experts were asked to choose between three different, anonymized outputs per incident (one per framework examined), the ones that subjectively capture best each incident. Among a sample of 30 different incidents, experts chose PROTEAS’s output ~93% of the time and also indicated that they would also agree with a competing framework for ~45% of the samples.

Value to Content Providers: To quantify PROTEAS’s added value to an operator’s toolbox, we measure what percentage of its true positives coincide with alerts issued by existing QoE-tracking tools [52, 53]. Figure 10 illustrates the weekly breakdown of PROTEAS’s true positives by source of ground truth. Indeed, we observe that in CP_1 ’s case, over 50% of viewership alerts were not caught by QoE tracking tools, whereas for CP_2 and CP_3 this percentage was approximately equal to 35%. This shows that PROTEAS substantially complements QoE-based alerting workflows.

7.3 Anatomy of anomalies

Table 6 analyzes PROTEAS’s True Positives. We look at viewership surges and drops and examine the mean duration of each type of alert. Overall, we do not observe strong patterns in favor of surges or drops across the three content providers. In the case of CP_3 , however, we note that the number of drops is generally comparable to the number of surges. We attribute that to the fact that CP_3 is the largest content provider of the three with 164 different channels.

Figure 11 shows a breakdown by group dimensions of PROTEAS’s issued alerts and Figure 12 depicts a breakdown by anomaly signature. For CP_1 , W_1 was marked by a number of surge alerts during the airing of a popular show’s season finale. These alerts were raised as geographically-localized flash crowds. Also, during W_1 , CP_1 experienced a major device failure which resulted in device-related drop alerts. In W_2 , CP_1 ’s viewership patterns changed due to changes

	Dimension	W1	W2	W3	W4
CP_1	Surges	72%	10%	94%	60%
	Drops	28%	90%	6%	40%
	Drop Duration (min)	58	20	23	38
	Surge Duration (min)	36	10	50	88
CP_2	Surges	81%	11%	51%	71%
	Drops	19%	89%	49%	29%
	Drop Duration (min)	38	54	40	55
	Surge Duration (min)	21	24	26	30
CP_3	Surges	53%	52%	65%	71%
	Drops	47%	48%	35%	29%
	Drop Duration (min)	38	54	40	55
	Surge Duration (min)	21	24	26	30

Table 6: Characteristics of True Positives

Figure 11: Breakdown of True Positives by session attributes.

in offered content. This manifested as drops that were correctly classified as model drifts. During W_3 and W_4 , the newly-imposed, geographically-diverse shelter-in-place orders in the US triggered large surges and pervasive changes in viewership patterns (e.g., viewership surges during business hours and in devices generally used in the home). PROTEAS, categorized these surges both as model drift and flash crowds.

The alerts of CP_2 and CP_3 tell us a different story given their frequent live content transmissions. CP_2 airs live shows at fixed time-slots during the day (e.g., daily news, sports). During W_1 , PROTEAS raised multiple flash crowd alerts during the transmission of Champion’s League soccer games. On the other hand, the high numbers of drop alerts during W_2 , were attributed to the cancellations of multiple live transmissions due to COVID-19.⁹ Similar to CP_1 , we also observe shifts in the use of devices, generally found in the home (e.g., AppleTV, Chromecast etc.). W_3 and W_4 are marked by sudden flash crowds when CP_2 aired its evening news. Similarly, the key observation about CP_3 is the rise in surges during W_3 and W_4 due to flash crowds in specific channels dedicated to daily news, stock market etc.

Analysis of anomaly signatures: In the case of CP_1 , PROTEAS correctly classified surges during W_1 as flash crowds. In addition, PROTEAS correctly diagnosed all technical failures CP_1 experienced during the pilot (one device failure and one high-impact platform failure). During the pandemic, PROTEAS classified smooth, persistent viewership surges as cases of model drift whereas sharper surges (especially during the prime time zone) were classified as

⁹ CP_2 is based in Europe and measures were taken earlier than in USA.

Figure 12: Breakdown of True Positives by anomaly signature.

flash crowds. In either case, analysts agreed that either diagnosis would be sufficient for content providers to re-provision resources as and when needed. Regarding CP_2 , >90% of the surges of W_1 were correctly associated with flash crowds during sports events whereas in W_2 , drops were almost exclusively marked as model drift due to the cancellation of live transmissions. Overall, our manual verification of PROTEAS’s root-cause attribution showed that PROTEAS gave an acceptable diagnosis for ~90% of all true positive alerts and mis-classified the remaining 10%.

Analysis of False Positives: Perhaps unsurprisingly, PROTEAS’s false positives rose significantly during W_3 and W_4 across all three content providers. Manual investigation of these alerts showed that >85% of false positives could be traced back to the same cause. Recall that during online detection (§4.4), we estimate the shape of viewership at a new timestamp t' using summary statistics of past weeks. In the common operating scenario, this provides sufficiently accurate estimates but that hypothesis did not always hold during W_3 and W_4 . Due to the shelter-in-place order, total viewership showed a sharp increase by 40% which degraded the quality of online estimates for the shape of viewership.

Impact of COVID-19 on viewership: The COVID-19 pandemic caused massive changes in viewership [36, 44, 63]. PROTEAS detected large viewership surges during business hours. These surges rolled out gradually across the US, first in Seattle, California, NY and finally elsewhere, thus tracking not only the progression of the pandemic but also the political response. Second, we observed swift changes in content preferences, with multiple flash crowd and model-drift alerts being traced back to news channels. Unsurprisingly, we also saw a shift to home-centric players and devices (*i.e.*, AppleTV, FireTV *etc.*), unlike other types of players (mobile devices or custom-players for hotel entertainment systems [7]).

7.4 Component-wise sensitivity analysis

To complement the pilot, we manually injected anomalies onto viewership. Specifically, we generate a synthetic dataset by means of a custom anomaly generator that reproduces anomaly signatures observed in the wild and superimposes them to healthy segments of the original trace (Appendix C).

Detection Accuracy: Figure 13 compares PROTEAS with several additional baselines: Holt-Winters [39], ARIMA [88], regression using Fourier, Wavelet basis functions as well as combinations thereof [25]. For Holt-Winters and ARIMA we used Python’s StatsModels package [16]. Regarding Fourier and Wavelet (Debauchies and

Haar) transforms we chose the smallest set of coefficients that collectively accounted for 98% of the original signal’s energy [26, 30]. We set the confidence bounds to 3 standard deviations from the mean value. Figure 13 shows the resulting F-score (combining precision/recall [8]) and also the precision and recall of the simulation.

Figure 13: Detection Accuracy using synthetic traces

Model Accuracy vs. Compactness: As discussed in §4 there is a trade-off between the size of the detection model and accuracy. Indeed, our experiments (shown in Table 7) indicate that accuracy drops sharply when the distance between two observations exceeds 5 minutes, making that the optimal sampling candidate for timestamps in X in terms of accuracy and compactness.

Sampling period (min)	1	5	10	30	60
F-score	0.88	0.86	0.67	0.39	0.38

Table 7: Accuracy vs Compactness tradeoff

Summarization accuracy: Table 8 shows accuracy of the summarization heuristics as a function of k , the number of critical groups PROTEAS issues alerts for. These results represent 500 randomized tests across providers where the critical group was chosen at random and an anomaly was injected and propagated as discussed in §5.1. In Table 8, we compare our results with two competing approaches: (1) output the group that exhibits the highest nominal change in viewership value (Naive Summarization) and (2) Adtributor [34].

Number of Top- k Groups	1	2	3	4	5
PROTEAS	0.5	0.92	0.98	1.0	1.0
Adtributor [34]	0.23	0.67	0.8	0.91	0.93
Naive Summarization	0.22	0.29	0.41	0.66	0.71

Table 8: Summarization Accuracy Results

Indeed, when $k = 3$, PROTEAS is almost guaranteed to identify the critical group. We attribute the jump from $k = 1$ to $k = 2$ to the fact that the generator often picks a group with only one descendant. That descendant (whose viewership is equal to that of its critical parent) was the top choice as our heuristics would favor it over the parent. In these cases, the critical parent node had the second highest EP score.

8 DISCUSSION AND FUTURE WORK

In this section, we offer suggestions for future work based on observations collected while designing PROTEAS.

Applicability of PROTEAS’s insights across video streaming services: PROTEAS leverages domain-specific insights that are derived from viewership patterns of large TV-providers. Nevertheless, with the proliferation of streaming video on social media and on dedicated platforms, such as Twitch and YouTube, a natural question is whether PROTEAS’s insights and workflow can efficiently transfer to these non-TV domains. We hypothesize that PROTEAS’s insights (e.g., long-term shape persistence of viewership) could transfer well to these domains, under the right assumptions. A key differentiator between TV-providers and streaming services like Twitch is the ad-hoc nature of the content offered. That is, in the case of TV-providers, there is some determinism with respect to when different pieces of content are offered, something that does not apply to other streaming services. Therefore, while in PROTEAS we leverage shape persistence as a function of time (e.g., viewership patterns on Monday evenings), we would argue in favor of exploring shape persistence as a function of content type (e.g., viewership patterns during a football game).

Extracting ground truth: A key challenge in evaluating PROTEAS was the lack of ground truth and the need to crowdsource information needed to verify the accuracy of the framework. We identify an opportunity in automating this process through the design of a labeling framework for viewership anomalies. Such a framework would crowdsource relevant metadata based on the alerts issued by PROTEAS, request the feedback of viewership analysts and, over time, build a comprehensive library of events and event signatures for the analysts to use.

Usability of alerts: As discussed throughout this work, efficiently monitoring video streaming services relies on operators having at their disposal multiple ‘lenses’, offering multiple complementary views on their data. Nevertheless, we also observed that, given the diversity of monitoring workflows, the burden of combining these complementary views falls predominantly on the analyst. Specifically, in the evaluation of PROTEAS, the process of comparing PROTEAS’s alerts with alerts from other QoE workflows was often a manual one. Therefore, we believe there is opportunity in exploring the design of tools that take as input alerts from various alerting workflows and further facilitate the diagnosis process in the video streaming landscape [41, 57, 81, 82].

9 RELATED WORK

Understanding viewership: Abrahamsson *et al.* examine how content popularity for VoD content providers can impact caching strategies [21]. Moon *et al.*, analyze the evolution of viewership of user-generated content [37]. Other efforts focus on understanding specific event characteristics (e.g., 2008 Olympics) and contrast differential consumption of live vs. VoD content [85]. These efforts are complementary to our work and can potentially be used in conjunction with PROTEAS’s root cause attribution framework.

Timeseries anomaly detection: Timeseries anomaly detection is a classical problem that has been extensively studied in many

diverse domains and there are several detailed surveys on this topic (e.g., [23, 38, 48, 50, 68]). Techniques like spectral processing or wavelets, have been used in the field of signal processing and for internet traffic patterns [26, 30, 51]. Other techniques e.g., Kalman filtering, PCA, variational autoencoders or extreme value theory have also been proposed as detection candidates [18, 59, 66, 74, 75, 83]. Twitter, Facebook, Snapchat recently released their own statistical technique for timeseries anomaly detection [14, 49, 79]. Among these tools, Surus and Twitter are the closest match for viewership but as we saw in §7, they offer low precision. While the range of options is very wide as we discussed, Gaussian Processes appear a natural fit for capturing the semantics of video viewership.

Internet video: In the context of video streaming, other efforts have focused on QoE. CFA leverages domain specific insights that enable accurate prediction of video quality [52]. Other prior work has shown correlations between video quality metrics and user engagement [29, 76]. There is also large literature in measuring video quality in the wild [72, 85], quality issues [53], server selection [78] as well as techniques to improve user experience [54, 86]. While the types of datasets we consider are similar, our work focuses on the orthogonal problem of viewership anomaly detection.

Anomaly summarization and root-cause analysis: Previous research has also focused on extensively analyzing root-causing in systems and networks [22, 24, 28, 31, 53, 55, 56, 61, 67, 69]. The types of data and problems these focus on is orthogonal to our focus. More closely related to our work is prior work on root-cause attribution and anomalies summarization in the context of ad platforms [34]. Our notion of anomaly signatures and generating compact summaries builds from this literature but customizes it for viewership.

10 CONCLUSIONS

Accurate detection and diagnosis of viewership anomalies can complement existing management tasks for Internet video providers, as it can provide unique insights that QoE measurements cannot capture. However, the complex spatiotemporal relationships in video viewership make this problem challenging, preventing the use of many conventional techniques. PROTEAS identifies key structural insights about viewership to enable accurate and real-time detection and diagnosis. Our evaluation and pilot studies with operators suggest that PROTEAS outperforms many existing solutions.

ACKNOWLEDGEMENTS

We thank our shepherd and the reviewers for their constructive feedback. This work was supported in part by the CONIX Research Center, one of six centers in JUMP, a Semiconductor Research Corporation (SRC) program sponsored by DARPA, ERDF Project AIDA (POCI-01-0247-FEDER-045907) and NSF awards No. CNS-1513764 and CNS-1565343 the Kavcic-Moura research award.

REFERENCES

- [1] Comcast outage. <https://www.miamiherald.com/news/nation-world/national/article214067869.html>.
- [2] Compira Labs. <https://www.compirlabs.com/>.
- [3] Confidence Interval Critical Values. <https://bit.ly/3bRM9aK>.
- [4] Conviva - Real-time Streaming Video Intelligence. <https://www.conviva.com/>.
- [5] Cosine Distance. <https://docs.scipy.org/doc/scipy-0.14.0/reference/generated/scipy.spatial.distance.cosine.html>.
- [6] Down Detector. <https://www.downdetector.com/>.
- [7] ENSEO systems. <https://www.enseo.com/>.
- [8] F-score. <https://deeptai.org/machine-learning-glossary-and-terms/f-score>.
- [9] LightUp AI. <https://www.lightup.ai/>.
- [10] Matern Kernel. https://en.wikipedia.org/wiki/Mat%C3%A9rn_covariance_function.
- [11] Ooyala. <https://ooyala.dalet.com/>.
- [12] Periodic Kernel. http://ghamrick.github.io/gaussian_processes/gp.kernels.PeriodicKernel.html.
- [13] Precision Recall Analysis. <https://towardsdatascience.com/a-look-at-precision-recall-and-f1-score-36b5fd0dd3ec>.
- [14] Prophet. <https://facebook.github.io/prophet/>.
- [15] Squared Exponential Kernel. <http://evelinag.com/Ariadne/covarianceFunctions.html>.
- [16] StatsModels. <https://www.statsmodels.org/stable/index.html>.
- [17] Streaming Video Alliance. <https://www.streamingvideoalliance.org/>.
- [18] SURUS - Anomaly detection at Netflix. <https://netflixtechblog.com/rad-outlier-detection-on-big-data-d6b0494371cc>.
- [19] Verizon Outage. <https://bgr.com/2019/06/24/internet-outage-2019-google-amazon-reddit-down/>.
- [20] White noise Kernel. https://scikit-learn.org/stable/modules/generated/sklearn.gaussian_process.kernels.WhiteKernel.html.
- [21] H. Abrahamsson and M. Nordmark. Program popularity and viewer behaviour in a large tv-on-demand system. In *Proceedings of the 2012 Internet Measurement Conference, IMC '12*, pages 199–210, New York, NY, USA, 2012. ACM.
- [22] B. Agarwal, R. Bhagwan, T. Das, S. Eswaran, V. N. Padmanabhan, and G. M. Voelker. Netprints: Diagnosing home network misconfigurations using shared knowledge. In *NSDI*, volume 9, pages 349–364, 2009.
- [23] C. C. Aggarwal. *Outlier Analysis*. Springer Publishing Company, Incorporated, 2013.
- [24] F. Ahmed, J. Erman, Z. Ge, A. X. Liu, J. Wang, and H. Yan. Detecting and localizing end-to-end performance degradation for cellular data services. In *IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications*, pages 1–9. IEEE, 2016.
- [25] W. Aiello, A. Gilbert, B. Rexroad, and V. Sekar. Sparse approximations for high fidelity compression of network traffic data. In *Proceedings of the 5th ACM SIGCOMM conference on Internet measurement*, pages 22–22. USENIX Association, 2005.
- [26] V. Alarcon-Aquino and J. A. Barria. Anomaly detection in communication networks using wavelets. 2001.
- [27] S. Aminikhanghahi and D. J. Cook. A survey of methods for time series change point detection. *Knowl. Inf. Syst.*, 51(2):339–367, May 2017.
- [28] P. Bahl, R. Chandra, A. Greenberg, S. Kandula, D. A. Maltz, and M. Zhang. Towards highly reliable enterprise network services via inference of multi-level dependencies. *ACM SIGCOMM Computer Communication Review*, 37(4):13–24, 2007.
- [29] A. Balachandran, V. Sekar, A. Akella, S. Seshan, I. Stoica, and H. Zhang. Developing a predictive model of quality of experience for internet video. *SIGCOMM Comput. Commun. Rev.*, 43(4):339–350, Aug. 2013.
- [30] P. Barford, J. Kline, D. Plonka, and A. Ron. A signal analysis of network traffic anomalies. In *Proceedings of the 2Nd ACM SIGCOMM Workshop on Internet Measurement, IMW '02*, pages 71–82, New York, NY, USA, 2002. ACM.
- [31] P. Barham, A. Donnelly, R. Isaacs, and R. Mortier. Using magpie for request extraction and workload modelling. In *OSDI*, volume 4, pages 18–18, 2004.
- [32] C. Bergmeir, R. J. Hyndman, and J. M. Benítez. Bagging exponential smoothing methods using stl decomposition and box-cox transformation. *International journal of forecasting*, 32(2):303–312, 2016.
- [33] D. J. Berndt and J. Clifford. Using dynamic time warping to find patterns in time series. In *KDD workshop*, volume 10, pages 359–370. Seattle, WA, 1994.
- [34] R. Bhagwan, R. Kumar, R. Ramjee, G. Varghese, S. Mohapatra, H. Manoharan, and P. Shah. Adtributor: Revenue debugging in advertising systems. In *11th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 14)*, pages 43–55, 2014.
- [35] C. M. Bishop. *Pattern recognition and machine learning*. springer, 2006.
- [36] T. Böttger, G. Ibrahim, and B. Vallis. How the internet reacted to covid-19: A perspective from facebook's edge network. In *Proceedings of the ACM Internet Measurement Conference, IMC '20*, page 34–41, New York, NY, USA, 2020. Association for Computing Machinery.
- [37] M. Cha, H. Kwak, P. Rodriguez, Y.-Y. Ahn, and S. Moon. Analyzing the video popularity characteristics of large-scale user generated content systems. *IEEE/ACM Trans. Netw.*, 17(5):1357–1370, Oct. 2009.
- [38] V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection: A survey. *ACM Comput. Surv.*, 41(3):15:1–15:58, July 2009.
- [39] C. Chatfield. The holt-winters forecasting procedure. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 27(3):264–279, 1978.
- [40] R. B. Cleveland, W. S. Cleveland, J. E. McRae, and I. Terpenning. Stl: A seasonal-trend decomposition. *Journal of official statistics*, 6(1):3–73, 1990.
- [41] A. Crotty, A. Galakatos, E. Zraggen, C. Binnig, and T. Kraska. Vizdom: interactive analytics through pen and touch. *Proceedings of the VLDB Endowment*, 8(12):2024–2027, 2015.
- [42] F. Dobrian, V. Sekar, A. Awan, I. Stoica, D. Joseph, A. Ganjam, J. Zhan, and H. Zhang. Understanding the impact of video quality on user engagement. *ACM SIGCOMM Computer Communication Review*, 41(4):362–373, 2011.
- [43] D. Duvenaud. *Automatic model construction with Gaussian processes*. PhD thesis, University of Cambridge, 2014.
- [44] A. Feldmann, O. Gasser, F. Lichtblau, E. Pujol, I. Poese, C. Dietzel, D. Wagner, M. Wichtlhuber, J. Tapiador, N. Vallina-Rodriguez, O. Hohlfeld, and G. Smaragdakis. The lockdown effect: Implications of the covid-19 pandemic on internet traffic. In *Proceedings of the ACM Internet Measurement Conference, IMC '20*, page 1–18, New York, NY, USA, 2020. Association for Computing Machinery.
- [45] W. A. Fuller. *Introduction to statistical time series*, volume 428. John Wiley & Sons, 2009.
- [46] F. A. Gers, J. Schmidhuber, and F. Cummins. Learning to forget: Continual prediction with lstm. 1999.
- [47] J. Görtler, R. Kehlbeck, and O. Deussen. A visual exploration of gaussian processes. *Distill*, 2019. <https://distill.pub/2019/visual-exploration-gaussian-processes>.
- [48] D. Hawkins. *Identification of Outliers*. Chapman and Hall, 1980.
- [49] J. Hochenbaum, O. S. Vallis, and A. Kejariwal. Automatic anomaly detection in the cloud via statistical learning. *arXiv preprint arXiv:1704.07706*, 2017.
- [50] V. J. Hodge and J. Austin. A survey of outlier detection methodologies. *Artificial Intelligence Review*, 22(2):85–126, Oct 2004.
- [51] A. Hussain, J. Heidemann, J. Heidemann, and C. Papadopoulos. A framework for classifying denial of service attacks. In *Proceedings of the 2003 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, SIGCOMM '03*, pages 99–110, New York, NY, USA, 2003. ACM.
- [52] J. Jiang, V. Sekar, H. Milner, D. Shepherd, I. Stoica, and H. Zhang. Cfa: A practical prediction system for video qoe optimization. In *Proceedings of the 13th Usenix Conference on Networked Systems Design and Implementation, NSDI'16*, pages 137–150, Berkeley, CA, USA, 2016. USENIX Association.
- [53] J. Jiang, V. Sekar, I. Stoica, and H. Zhang. Shedding light on the structure of internet video quality problems in the wild. In *Proceedings of the ninth ACM conference on Emerging networking experiments and technologies*, pages 357–368. ACM, 2013.
- [54] J. Jiang, V. Sekar, and H. Zhang. Improving fairness, efficiency, and stability in http-based adaptive video streaming with festive. *IEEE/ACM Transactions on Networking (ToN)*, 22(1):326–340, 2014.
- [55] S. Kandula, R. Mahajan, P. Verkaik, S. Agarwal, J. Padhye, and P. Bahl. Detailed diagnosis in enterprise networks. *ACM SIGCOMM Computer Communication Review*, 39(4):243–254, 2009.
- [56] R. R. Kompella, J. Yates, A. Greenberg, and A. C. Snoeren. Fault localization via risk modeling. *IEEE Transactions on Dependable and Secure Computing*, 7(4):396–409, 2009.
- [57] T. Kraska. Northstar: An interactive data science system. 2021.
- [58] S. S. Krishnan and R. K. Sitaraman. Video stream quality impacts viewer behavior: Inferring causality using quasi-experimental designs. In *Proceedings of the 2012 Internet Measurement Conference, IMC '12*, pages 211–224, New York, NY, USA, 2012. ACM.
- [59] N. Laptev, S. Amizadeh, and I. Flint. Generic and scalable framework for automated time-series anomaly detection. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1939–1947, 2015.
- [60] A. Lazarevic, A. Banerjee, V. Chandola, V. Kumar, and J. Srivastava. Data mining for anomaly detection. In *Tutorial at the European Conference on Principles and Practice of Knowledge Discovery in Databases*, 2008.
- [61] Q. Lin, J.-G. Lou, H. Zhang, and D. Zhang. idice: problem identification for emerging issues. In *Proceedings of the 38th International Conference on Software Engineering*, pages 214–224, 2016.
- [62] X. Liu, F. Dobrian, H. Milner, J. Jiang, V. Sekar, I. Stoica, and H. Zhang. A case for a coordinated internet video control plane. In *Proceedings of the ACM SIGCOMM 2012 conference on Applications, technologies, architectures, and protocols for computer communication*, pages 359–370, 2012.
- [63] A. Lutu, D. Perino, M. Bagnulo, E. Frias-Martinez, and J. Khargosstar. A characterization of the covid-19 pandemic impact on a mobile network operator traffic. In *Proceedings of the ACM Internet Measurement Conference, IMC '20*, page 19–33, New York, NY, USA, 2020. Association for Computing Machinery.

- [64] V. Morgenstern, B. Upadhyaya, and M. Benedetti. Signal anomaly detection using modified cusum method. In *Proceedings of the 27th IEEE Conference on Decision and Control*, pages 2340–2341. IEEE, 1988.
- [65] G. Nason. *Stationary and non-stationary time series*, pages 129 – 142. Geological Society of London, United Kingdom, 2006.
- [66] J. Ndong and K. Salamatian. Signal processing-based anomaly detection techniques: A comparative analysis. 06 2011.
- [67] B. Nguyen, Z. Ge, J. Van der Merwe, H. Yan, and J. Yates. Absence: Usage-based failure detection in mobile networks. In *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*, pages 464–476, 2015.
- [68] K. Ord. Outliers in statistical data : V. Barnett and T. Lewis, 1994, 3rd edition, (John Wiley & Sons, Chichester), 584 pp., [UK pound]55.00, ISBN 0-471-93094-6. *International Journal of Forecasting*, 12(1):175–176, March 1996.
- [69] X. Pan, J. Tan, S. Kavulya, R. Gandhi, and P. Narasimhan. Ganesha: Blackbox diagnosis of mapreduce systems. *ACM SIGMETRICS Performance Evaluation Review*, 37(3):8–13, 2010.
- [70] B. Pang, L. Lee, et al. Opinion mining and sentiment analysis. *Foundations and Trends® in Information Retrieval*, 2(1–2):1–135, 2008.
- [71] D.-S. Pham, S. Venkatesh, M. Lazarescu, and S. Budhaditya. Anomaly detection in large-scale data stream networks. *Data Mining and Knowledge Discovery*, 28(1):145–189, 2014.
- [72] L. Plissonneau and E. Biersack. A longitudinal view of http video streaming performance. In *Proceedings of the 3rd Multimedia Systems Conference*, pages 203–214. ACM, 2012.
- [73] C. E. Rasmussen. Gaussian processes in machine learning. In *Summer School on Machine Learning*, pages 63–71. Springer, 2003.
- [74] H. Ren, B. Xu, Y. Wang, C. Yi, C. Huang, X. Kou, T. Xing, M. Yang, J. Tong, and Q. Zhang. Time-series anomaly detection service at microsoft. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 3009–3017, 2019.
- [75] A. Siffer, P.-A. Fouque, A. Termier, and C. Largouet. Anomaly detection in streams with extreme value theory. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1067–1075, 2017.
- [76] H. H. Song, Z. Ge, A. Mahimkar, J. Wang, J. Yates, Y. Zhang, A. Basso, and M. Chen. Q-score: Proactive service quality assessment in a large iptv system. In *Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference*, pages 195–208. ACM, 2011.
- [77] M. Theodosiou. Forecasting monthly and quarterly time series using stl decomposition. *International Journal of Forecasting*, 27(4):1178–1195, 2011.
- [78] R. Torres, A. Finamore, J. R. Kim, M. Mellia, M. M. Munafo, and S. Rao. Dissecting video server selection strategies in the youtube cdn. In *2011 31st International Conference on Distributed Computing Systems*, pages 248–257. IEEE, 2011.
- [79] W. Wang, Z. Liu, X. Shi, and L. Pierce. Online fdr controlled anomaly detection for streaming time series. 2019.
- [80] W.-K. Wong and D. B. Neill. Tutorial on event detection kdd 2009.
- [81] D. Xin, H. Miao, A. Parameswaran, and N. Polyzotis. Production machine learning pipelines: Empirical analysis and optimization opportunities, 2021.
- [82] D. Xin, E. Y. Wu, D. J.-L. Lee, N. Salehi, and A. Parameswaran. Whither automl? understanding the role of automation in machine learning workflows. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, pages 1–16, 2021.
- [83] H. Xu, W. Chen, N. Zhao, Z. Li, J. Bu, Z. Li, Y. Liu, Y. Zhao, D. Pei, Y. Feng, et al. Unsupervised anomaly detection via variational auto-encoder for seasonal kpis in web applications. In *Proceedings of the 2018 World Wide Web Conference*, pages 187–196, 2018.
- [84] Z. Xu, K. Kersting, and L. Von Ritter. Stochastic online anomaly analysis for streaming time series. In *IJCAI*, pages 3189–3195, 2017.
- [85] H. Yin, X. Liu, F. Qiu, N. Xia, C. Lin, H. Zhang, V. Sekar, and G. Min. Inside the bird’s nest: Measurements of large-scale live vod from the 2008 olympics. In *Proceedings of the 9th ACM SIGCOMM Conference on Internet Measurement, IMC ’09*, pages 442–455, New York, NY, USA, 2009. ACM.
- [86] X. Yin, A. Jindal, V. Sekar, and B. Sinopoli. A control-theoretic approach for dynamic adaptive video streaming over http. In *ACM SIGCOMM Computer Communication Review*, volume 45, pages 325–338. ACM, 2015.
- [87] M. Zaharia, R. S. Xin, P. Wendell, T. Das, M. Armbrust, A. Dave, X. Meng, J. Rosen, S. Venkataraman, M. J. Franklin, et al. Apache spark: a unified engine for big data processing. *Communications of the ACM*, 59(11):56–65, 2016.
- [88] G. P. Zhang. Time series forecasting using a hybrid arima and neural network model. *Neurocomputing*, 50:159–175, 2003.
- [89] Z.-G. Zhou and P. Tang. Improving time series anomaly detection based on exponentially weighted moving average (ewma) of season-trend model residuals. In *2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pages 3414–3417. IEEE, 2016.

A ETHICAL CONSIDERATIONS

To the extent that the dataset tracks actual viewer behavior, our dataset does concern real Internet users. All viewership data used in this work are covered by NDAs prohibiting any re-sharing with 3rd parties even for research purposes. Further, raw viewership data have been reviewed and validated by the operator with respect to GDPR compliance (e.g., no identifier can be associated to person), and data processing only looks at viewership counts per combination of session dimension values. No personal and/or contact information was available in the viewership data used for this study. The data for the ground truth collection (e.g., Twitter) also entail user data but they are in the public domain. We do not use user/poster names, only the contents relevant to the incidents under evaluation.

B GAUSSIAN PROCESS FORMULATION

Recall from §4 that we define random variable $S_{g,w}(t)$ as follows:

$$S_{g,w}(t) \equiv \bar{S}_{g,w}(t) + \bar{\epsilon}_{g,w}(t) \quad \forall t \in T, \quad \text{where} \quad (8)$$

$$\begin{aligned} \bar{S}_{g,w}(t) &\sim \mathcal{N}\left(\frac{1}{|D|} \sum_i s_{g,w}^{d_i}(t), 0\right), \\ \bar{\epsilon}_{g,w}(t) &\sim \mathcal{N}(0, \sigma_t^2) \end{aligned}$$

$\bar{S}_{g,w}(t)$ is the random variable whose mean for every $t \in X$ is the empirical mean of historical shape observations at that time. σ_t^2 is the standard deviation of $s_{g,w}^d(t) \quad \forall d \in D$.

To extract the desired objectives $\hat{s}_{g,w}(t')$, $t' \in Y$ and $\hat{u}_{g,w}(t')$, we first model the joint Gaussian distribution of $S_{g,w}(X)$ and $S_{g,w}(Y)$. We use the notation K_{XY} to refer to the covariance matrix of $S_{g,w}(X)$ and $S_{g,w}(Y)$ (and apply the same notational logic to all four possible combinations of $S_{g,w}$ sets). Formally:

$$\begin{aligned} \begin{bmatrix} S_{g,w}(X) \\ S_{g,w}(Y) \end{bmatrix} &\sim \mathcal{N}(\mathbf{0}, \Sigma), \quad \text{where} \\ \Sigma &:= \begin{bmatrix} K_X + \sigma_{g,w}^2(X)I & K_{XY} \\ K_{XY} & K_{YY} \end{bmatrix}. \end{aligned} \quad (9)$$

Note that according to GP theory, the mean of the joint distribution can be safely set for notational simplicity to 0. Using (3), the formulas for $\hat{s}_{g,w}(t')$ and $\hat{u}_{g,w}(t')$ become:

$$\begin{aligned} \hat{s}_{g,w}(t') &= \mathbb{E}[S_{g,w}(Y) \mid S_{g,w}(X), X, Y] \\ &= K_{XY}^T [K_X + \sigma_{g,w}^2(X)I]^{-1} \mathbb{E}[S_{g,w}(X)] \end{aligned} \quad (10)$$

$$\begin{aligned} \hat{u}_{g,w}(t') &= \mathbb{V}[S_{g,w}(Y) \mid S_{g,w}(X), X, Y] \\ &= K_Y - K_{XY}^T [K_X + \sigma_{g,w}^2(X)I]^{-1} K_{XY} \end{aligned} \quad (11)$$

Definition of Kernels: As per (5), our choice of kernels is as follows:

$$\kappa_{\text{periodic}}(t, t') = \alpha^2 \exp\left(-\frac{2 \sin^2(\pi |t - t'|/p)}{\ell^2}\right) \quad (12)$$

$$\kappa_{\text{sqExp}}(t, t') = \gamma^2 \exp\left(-\frac{(t, t')^2}{2\ell^2}\right) \quad (13)$$

$$\kappa_{\text{Matern}}(t, t') = \frac{\beta^2}{\Gamma(\nu)2^{\nu-1}} \left(\frac{\sqrt{2\nu}(t, t')}{l}\right)^\nu B_\nu\left(\frac{\sqrt{2\nu}(t, t')}{l}\right) \quad (14)$$

$$\kappa_{\text{wNoise}}(t, t') = \sigma_{g,w}^2(X)I \quad (15)$$

Proof Of Validity For Noise Kernel: Typically, when defining a valid noise kernel the resulting covariance matrix needs to be positive semidefinite. Below we prove that the modified noise kernel above (where $\sigma_{g,w}^2(X)I$ is a function of the training timestamp) is positive semidefinite.

We now argue that the matrix inverse above is legitimate. First, it is well-known that a covariance matrix is always a positive semidefinite matrix [73]. Second, since $\sigma > 0$, $\sigma^2 I$ is a positive definite matrix. As the sum of a positive semidefinite matrix with a positive definite matrix is positive definite, XYZ is positive definite and therefore invertible.

C SYNTHETIC ANOMALY GENERATION

To synthetic anomaly generator that we use for our evaluation configures both the temporal and the spatial dimension of each generated anomaly.

Temporal Dimension: With respect to the temporal dimension our generator has the following configurable parameters:

1. **Start time:** The start time of the anomaly.
2. **Duration:** To configure the duration of the anomaly, we use a Markovian model where we define the probability of remaining in an anomalous state for the next timestamp
3. **Directionality:** This parameter determines whether the anomaly is going to be a surge, drop or oscillate over time.
4. **Change in magnitude:** This parameter determines how far the viewership deviates from the expected normalized w.r.t. the standard deviation;
5. **Gradient:** to simulate how sharp or smooth is the onset/recovery of the anomaly.

Spatial Dimension: Once we generate the temporal signature, we leverage the propagation mechanism discussed in §5.1 to spatially propagate the change to other viewership groups in the hierarchy.