Advances in Artificial Intelligence and Machine Learning; Research 2 (1) 303-314 Received 17-03-22; Accepted 27-03-22; Published 31-03-22

Estimating Data Loss At Scale

Wei Zhang wzhang@adobe.com
Ilya Reznik
Adobe Inc., USA.

Corresponding Author: Wei Zhang.

Abstract

For companies that serve corporate customers, Customer Service Outage (CSO) is a catastrophic event that may lead to some loss of their customer data. After each CSO, it is important to have a timely and quantitative measurement of how much data was lost. However, it is impractical for human to do so due to the enormous amount of data. In this paper, we present a robust solution that can return numerical loss report within hours. It handles a variety of challenges that are associated with the data. Consequently, management team can gauge the severity of data loss right after each event and respond accordingly.

Keywords: Machine learning application, Data loss estimation, Predictive modelling, Data analytics.

1. INTRODUCTION

Adobe Analytics is a comprehensive solution for real-time analytics across marketing channels. The data from Adobe Analytics powers Adobe Digital Economy Index [1], which provides a comprehensive view of US eCommerce by analyzing direct consumer transactions online. Many corporate customers use Adobe Analytics for marketing decision. We have seen tremendous growth over the past couple of years. Nevertheless, the ever-increasing data load makes it more prone to service interruptions. Customer Service Outage (CSO) is the most severe case, which may cause some loss of customer data. Prompt and accurate estimation of data loss is critical after each CSO. On one hand, we want to quickly gauge the severity of data loss. In addition, impacted customers may contact us right after an incident. We have to be prepared for the communication and our loss estimation needs to be readily available. In some cases, we may need to financially compensate our customers and the compensation is based on the amount of data loss. We don't want to understate the loss so to be fair to our customers. While overestimation is not acceptable either, as it will result in extra payment and more financial loss.

Citation: Wei Zhang and Ilya Reznik. Estimating Data Loss At Scale. Advances in Artificial Intelligence and Machine Learning. 2022;2(1):20.

2. PROBLEM DESCRIPTION

2.1 Data Collection

Adobe Analytics data collection pipeline starts at the edge servers scattered throughout the world. Our customers (e.g., retailers and communication service providers) all serve a large numbers of end users/consumers. A single interaction of their users with their service is called a *hit*. Higher number of hits means higher traffic to their service. Each hit is then ingested in a particular data center in Adobe and assigned to customer-defined bucket, which is called a *report suite*. One customer can have one or many report suites. Upon arrival in the data center each hit is processed through an elaborate pipeline until customer data are ready for reporting services.

During a customer service outage event which causes data loss, the problem can occur in any of the steps of the above-mentioned pipeline or at the edge. Thus, for this analysis we collect data at the very end of the pipeline when there is a record in the internal database that the hit will appear when queried from any of our reporting services.

A hit has a timestamp indicating when the interaction with customer's service happened. There is also a timestamp of when the hit completed processing and became available to the reporting services in Adobe Analytics. For data ingested in real-time (online interactions) the difference between these two timestamps is equivalent to the latency in Adobe Analytics system.

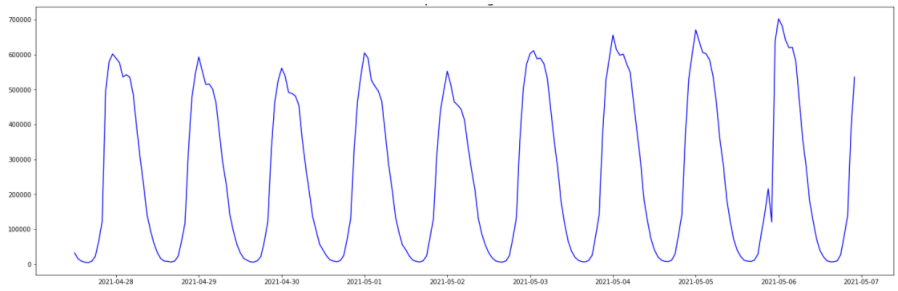
Our customers have abundant flexibility in reporting their traffic. The traffic data usually updates hourly. But in many cases, customers choose to report just once per day (called *bulk ingestion*). Alternatively, a customer can use a report suite to track interactions on a mobile native application which does not require internet connection. When their users connect to the internet they send the entire interaction history at once.

2.2 Data Description

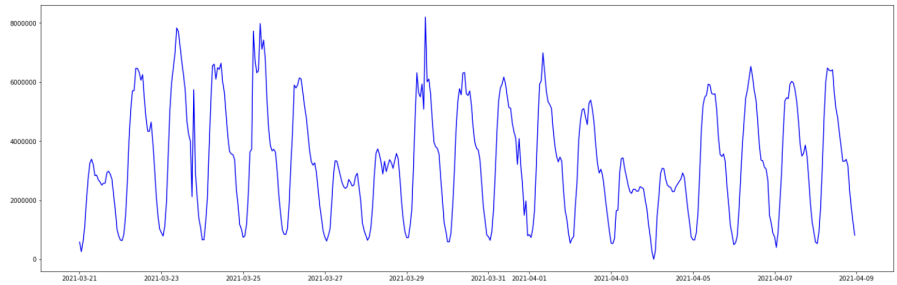
Adobe stores the historical traffic (number of hits) data for each report suite up to a certain length of time. Due to the nature of their businesses, the traffic in each report suite exhibits strong periodic pattern. See FIGURE 1 for some examples. FIGURE 1(d) is an example of bulk ingestion: the number of hits were reported just once per day for a period of time. The weekly pattern, high on weekdays and low on weekends, is fairly clear. It's also interesting to see the customer changed their reporting style on 4/28/2021, from bulk ingestion to regular hourly reporting. It seems the traffic is reduced as the curve is much lower after 4/28/2021. However, this is because when the traffic was reported once per day, the reported number is the sum over the entire day. If we add all 24 (the data is hourly) reported numbers within each day, the daily traffic is similar to previous days.

2.3 Data Loss Definition

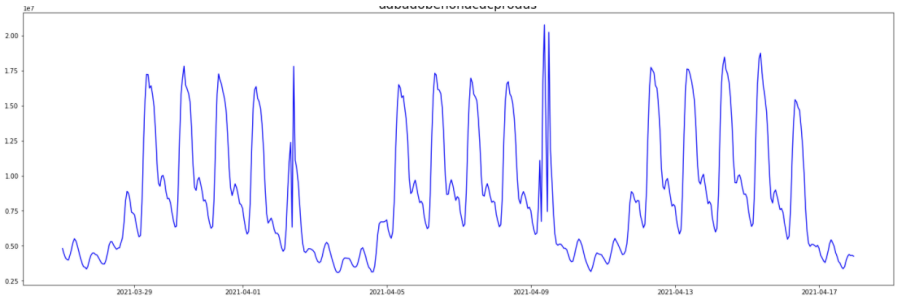
When a Customer Service Outage (CSO) happens, some customer data cannot pass through the data collection pipeline because of the outage (so these data were lost). Consequently, the traffic



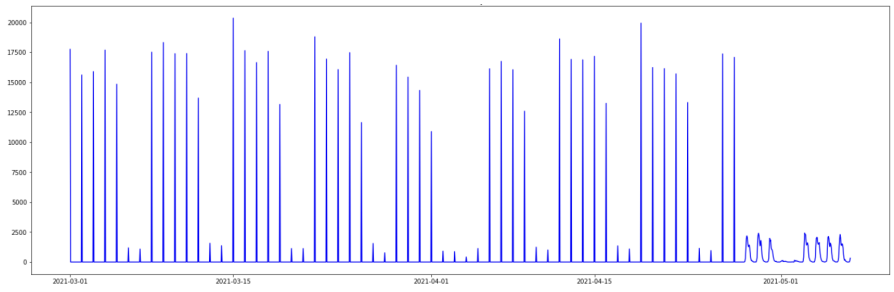
(a) Data with strong daily periodicity (a.k.a. seasonality)



(b) Daily and weekly periodicity



(c) Strong weekly pattern: high in weekdays and much lower during weekends.



(d) Weekly pattern + bulk ingestion + change of reporting

Figure 1: Examples of data.

(number of hits) that Adobe collects will be suddenly lower than usual. When a CSO is reported, the starting and ending time of the outage are reported along with its symptom description. Usually a CSO lasts for an hour or two, occasionally significantly longer. See FIGURE 2 for some examples of data loss. Note that an abrupt drop in traffic does not necessarily mean data loss, see FIGURE 3 for some examples. Cases like FIGURE 3(b) are very common. In a CSO event, most report suites will experience some reduction in the number of hits. However, for many report suits, the reduction is followed by a strong recovery, which means that some traffic get delayed (but not lost) then get through in the following hours. The delayed hits are added to the normal traffic during the recovery period, causing a spike. In other words, many report suits do not experience data loss as their traffic are recovered. We need to make sure to account for both reduction during a CSO and recovery after the CSO period. Otherwise, we may overestimate the loss.

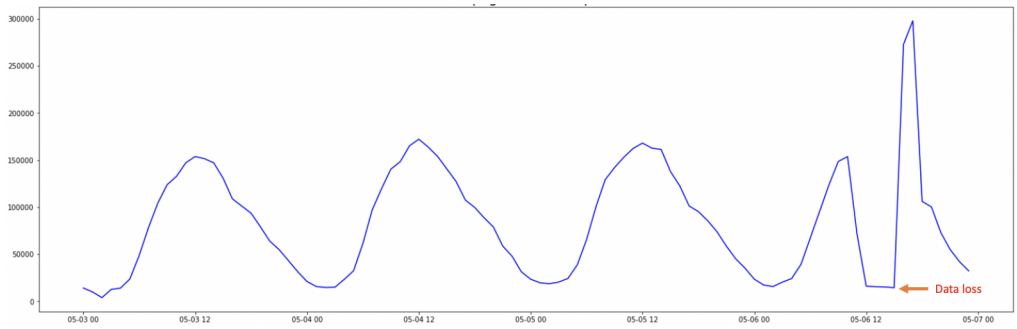
3. OUR SOLUTION

As can be seen from FIGURE 1, the data for each report suit exhibits its unique traffic pattern, with daily and weekly periodicity (seasonality). In the normal situation, the traffic for each report suit remains to be the same pattern. Based on the historical data, we can reliably predict the future traffic. So we propose a data-driven approach to tackle the problem. In general, our data loss estimation service works in three steps:

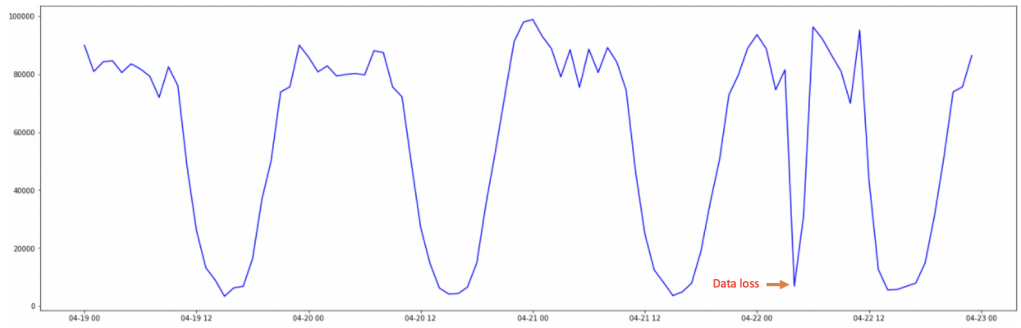
1. Pull historical (hourly) traffic data from reporting service.
2. Build a time-series forecasting model using data before the CSO day and predict the number of hourly hits during the CSO and recovery period.
3. Use the difference between the predicted hourly traffic and observed traffic as the loss estimation.

Many time-series forecasting algorithms can be used for building the model, as long as daily and weekly periodicity (a.k.a. seasonality) can be integrated in the model. Some widely used algorithms are Autoregressive Integrated Moving Average (ARIMA) [2], Seasonal ARIMA (SARIMA) [3] and LSTM [4]. Currently, we use the Prophet [5], library because it provides a robust and easy-to-use library. Given a time-series data $X = \{X_0, X_1, \ldots, X_i\}$, $X_i = (t_i, v_i)$ represents the traffic value v_i at time t_i, Prophet can fit a model M and decompose the data into trend, seasonality and holiday components. Then M can be used to predict the future traffic values. As can be seen in FIGURE 4, we can predict the expected traffic very reasonably based on the Prophet library.

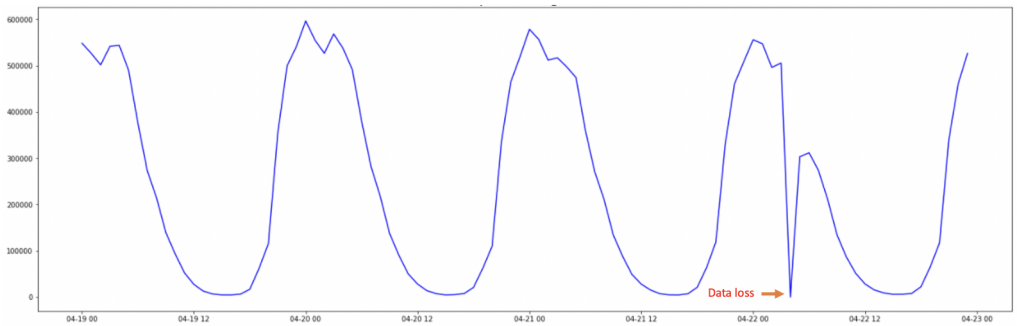
In addition to the choice of forecasting model, there are several critical issues that need to be addressed to make the loss estimation reliable, as we will elaborate next.



(a) A CSO of 4 hours on 05/06.

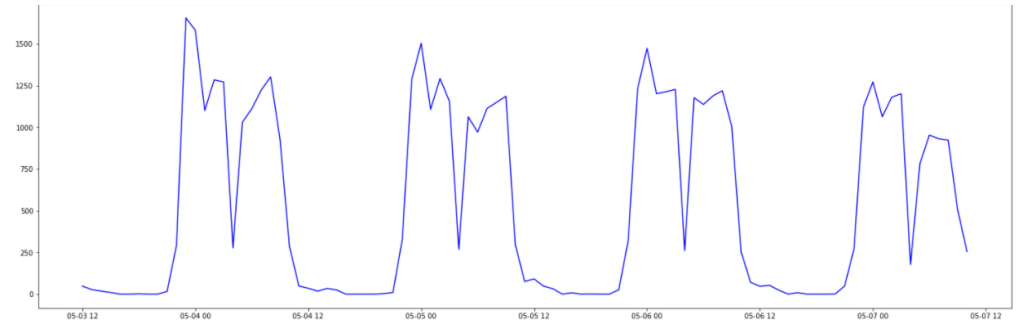


(b) A CSO of two hours on 04/22.

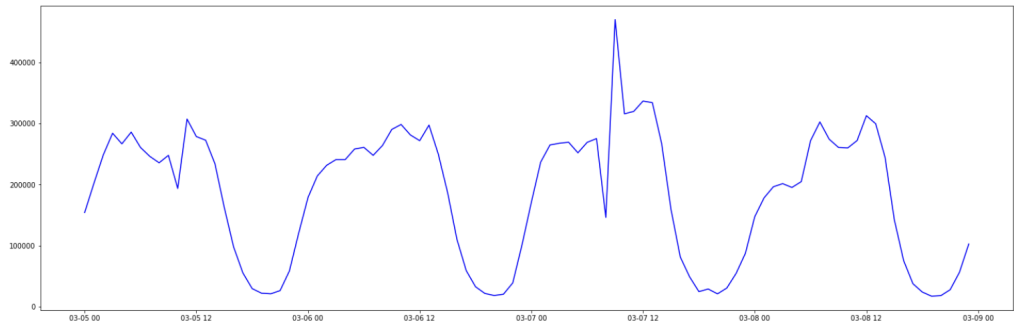


(c) Same CSO on 04/22, but different report suit.

Figure 2: Examples of data loss.



(a) Traffic drop due to daily periodicity



(b) Traffic drop then recovered.

Figure 3: Sudden traffic drops are not necessarily data loss.

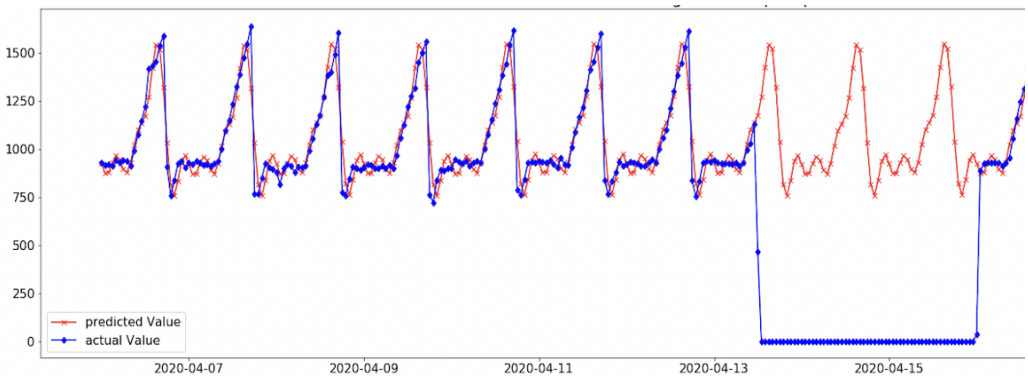


Figure 4: Predicted traffic (red curve) and the actual value (blue curve). There were a period of total data loss (observed values dropped to 0).

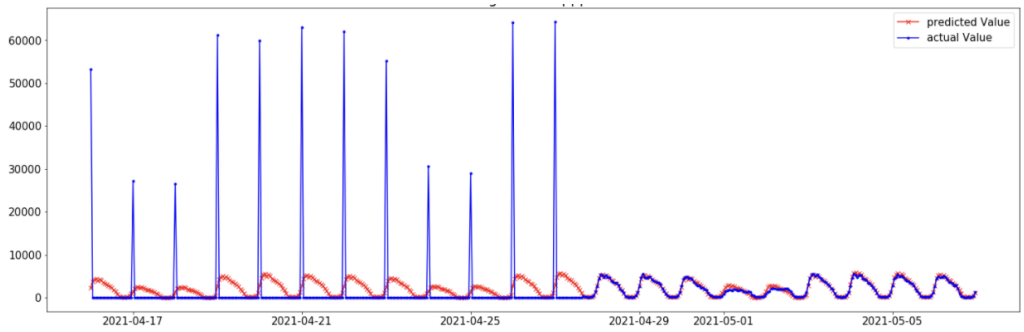


Figure 5: Decomposing bulk ingestion data. We can estimate how hourly traffic should be based on the bulk ingested daily number. The red curve represents the decomposed hours traffic. It has a much smaller range than the blue curve (raw data), because it represents hourly hits that are decomposed from daily hits. The weekly pattern is also preserved in the decomposed data: the red curve is lower during weekends just like the original blue curve.

3.1 Handling Bulk Ingestion

We can not build a high quality model directly using data that has bulk ingestion like FIGURE 1(d). So we always check if a data contains bulk ingested data points. This is relatively straightforward, as bulk ingested data points appear only once per day, instead of hourly. Once we determined that a data contains bulk ingestion, we use Algorithm 1 for loss estimation.

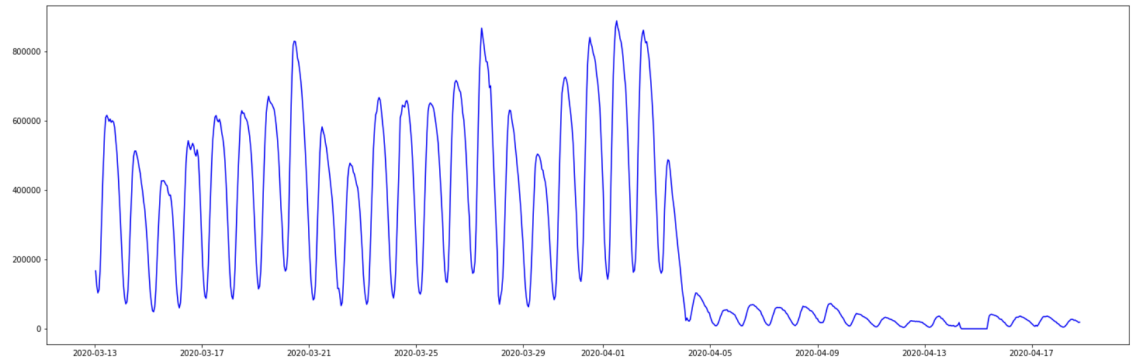
Algorithm 1 Loss estimation for bulk ingestion data

1. Convert hourly data to daily data by combining the total hits in each day.
2. Build forecasting model using the daily data before the CSO day, predict the total number of hits in the CSO day.
3. Estimate the hourly hits distribution.
4. Decompose the predicted daily hits to hourly.
5. Use the difference between the decomposed hourly hits and observed hits as the loss estimation.

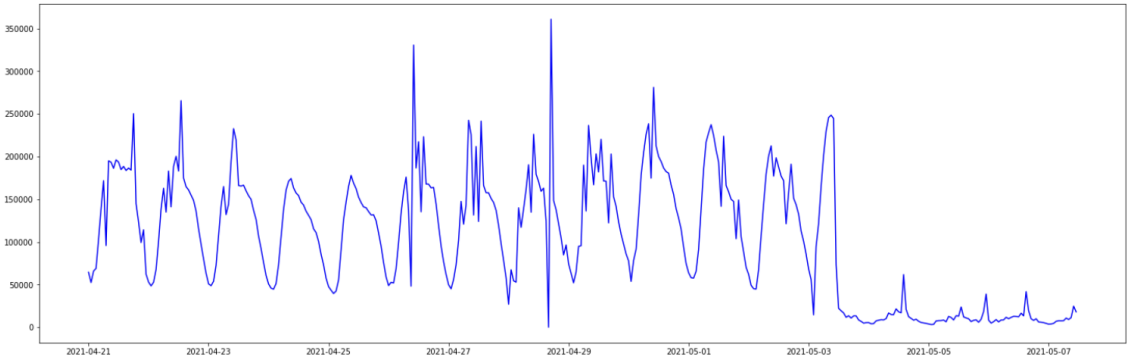
When trying to estimate the hourly hits distribution in Alg. 1 (3), we find all days that are not bulk ingested. Then we use the average hit distribution over these days as the hits distribution for each day. Consequently, we can decompose bulk ingested data into hourly data and make it possible to estimate data loss reliably, as shown in FIGURE 9(c).

3.2 Calibrating Based On The Previous day

Due to the flexibility in reporting, the activeness of any report suit can change over time. It is possible that a report suit gets less hits for other reasons rather than a CSO. See FIGURE 6 for some examples.



(a) CSO reported on Apr. 13^{th}, but traffic shrinked 10 days before.



(b) CSO reported on May. 6^{th}, but traffic lowered two days before.

Figure 6: Traffic were already significantly lower before CSO happened. So the reduction in traffic was not caused by CSO, meaning there were no data loss during the CSO period.

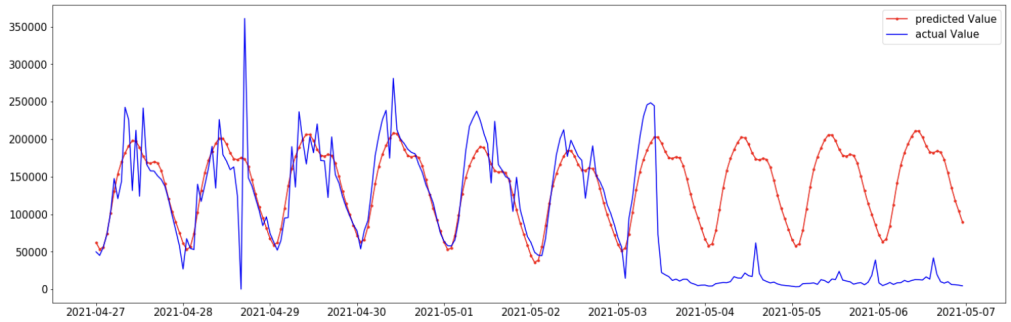


Figure 7: Predictive model based on historical data will continue forecasting the previous traffic pattern, because the customer behavior changed abruptly.

However, the prediction based on the historical data will keep the previous pattern, as illustrated in FIGURE 7. This will lead to overestimation of data loss for the CSO day.

In order to solve this problem, we propose to do calibration based on the day before the CSO. We first calculate the difference between the prediction and actual values in the day before the CSO, in exactly the same way as we estimating loss for the CSO period. So we get 24 hourly "calibration loss" estimation for the day before. If traffic reduction happened before the CSO, these "calibration loss" will be big. If traffic reduction only happens during the CSO period, these "calibration loss" numbers will be very small. Then we subtract these "calibration loss" from the corresponding loss estimation over the CSO period, so to get the true loss estimation. For the case in FIGURE 7, although the initial loss estimation is high, the reported data loss is negligible. Because the "calibration loss" offsets the initial loss estimation.

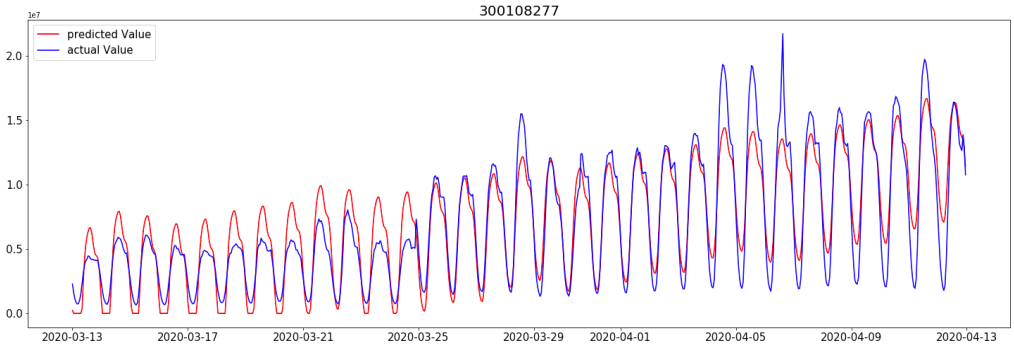
3.3 Choice of Seasonality Model

When building time-series forecasting models, seasonalities are typically treated as additive, which means that the effect of a seasonality is added to the trend to get the forecast. However, our data are special: they represent traffics volume of services. Volumes can grow over time and have significant difference between weekdays and weekends. As a result, a predictive model based on the additive seasonality does not work well for our data. We adopted the multiplicative seasonality and obtained better models. FIGURE 8 illustrates the difference. Based on our experiments, we achieved around 10% improvement by using multiplicative rather than additive seasonality.

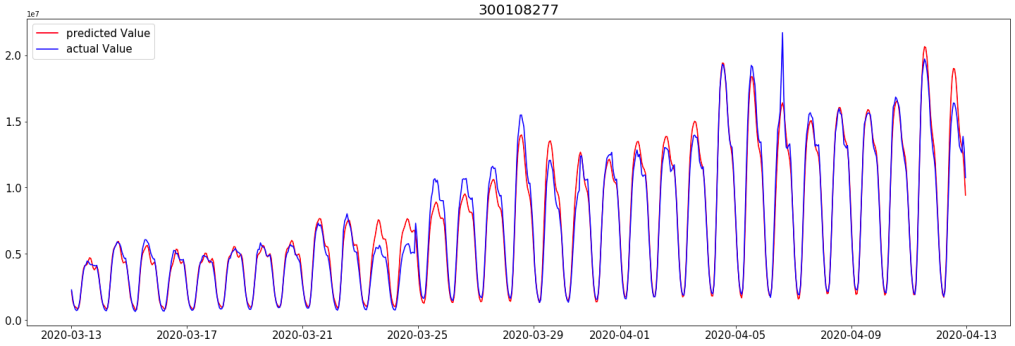
4. RESULTS AND DISCUSSION

The proposed solution has been used for multiple rounds of data loss estimations. As illustrated in FIGURE 9, our predictions of normal traffic are convincing even though traffic data exhibit all kinds of variations. We can quickly provide loss estimation for any individual report suit, e.g., "the report suit *ABCDE* experienced significant loss of 61.8%". The loss severity, measured by the percentage of traffic that is less than usual, is defined in Equation 1, where P is the predicted traffic value and O is the observed value. When O is higher than the expectation P, there is no loss so it is set to be 0.

$$\frac{\max(P - O, 0)}{P} \tag{1}$$

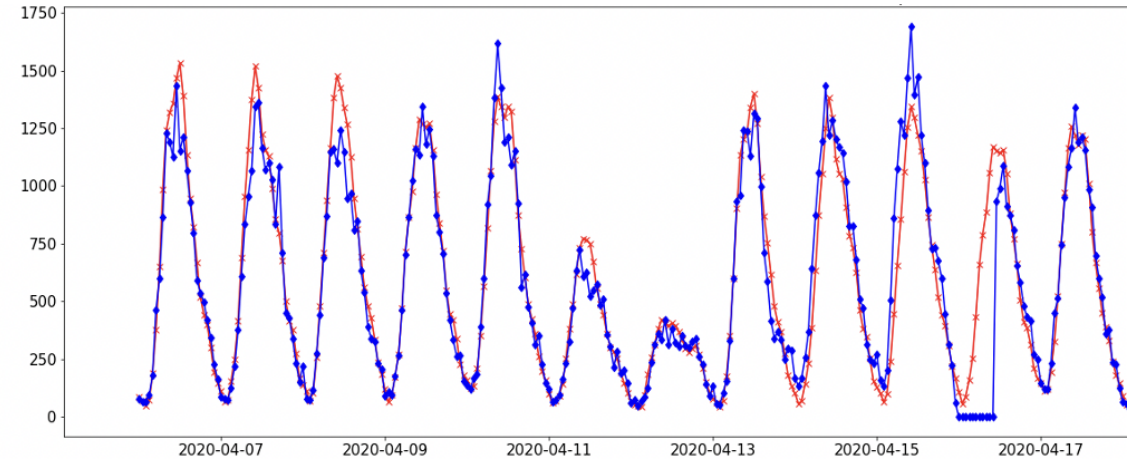


(a) Standard model based on additive seasonality can not closely fit the data.

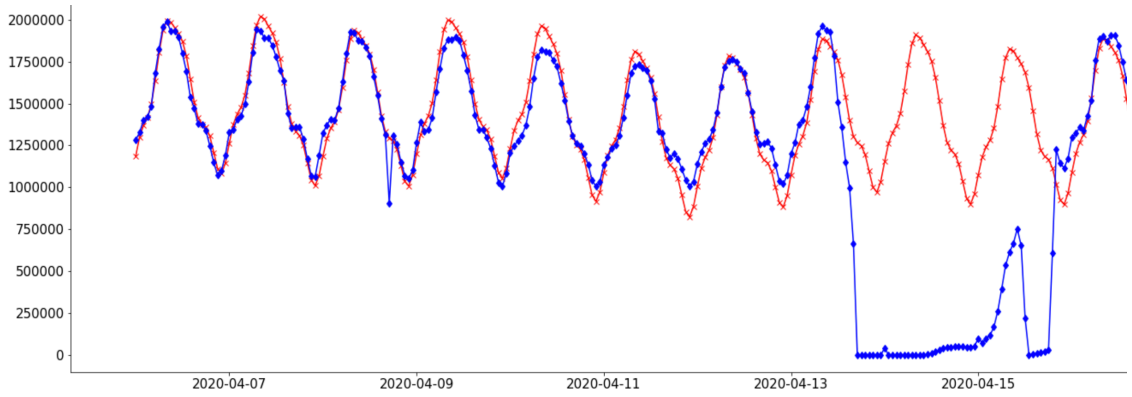


(b) Model based on multiplicative weekly seasonality fits same data much better.

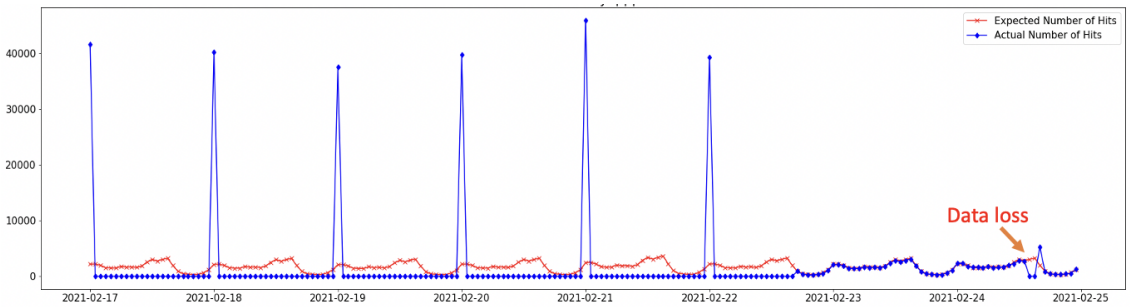
Figure 8: Model fitting: additive vs. multiplicative.



(a) Relative shorter period of data loss, but still evident on 04/16/20.



(b) One report suite suffered an extended period of loss.



(c) A case with bulk ingested data. Since we can decompose daily traffic to hourly, we can accurately estimate loss even though the customer changed its way of reporting.

Figure 9: Some examples for loss estimation: estimated normal traffic (red curve) vs. the actual value (blue curve)

The visualized results in Figure 9 only provide qualitative illustration. We also evaluated our solution quantitatively by doing "loss estimation" for the non-CSO time. During the non-CSO

time, the traffic is supposed to be normal and the estimated loss should be small. It will not be zero, however, due to natural data variations. In addition, the loss is non-negative by definition so above-normal traffic value cannot offset the loss value. We randomly selected 1000 report suits and ran the data loss estimation procedure for the day before the CSO day. The average loss estimation over these 1000 results is 1.9%, with a standard deviation of 5.4%. So the loss estimation is indeed small for normal cases. When we do observe data loss, the reported loss estimation will be a lot higher.

It is clear that our loss estimation algorithm can run in parallel for different report suits. When in urgent needs, we can spin up dozens of virtual machines (VM) in AWS or Azure and spread computations across different VMs. Once the loss estimations for all report suits are done, we can aggregate them and provide a big-picture view of each company or data center, for example: "the impact of this CSO is minor on the company *ABC*, the data loss is less than 1.5%", or "the data loss is significant at 26.3%".

5. CONCLUSION

In this paper, we present a robust solution for data loss estimation. It utilizes historical data to forecast the normal traffic for the CSO day, thus we can estimate the expected loss between the regular traffic and the observed traffic. We further improve loss estimation by adopting the multiplicative seasonality rather than the standard choice of the additive seasonality. Our solution is versatile and can handle many variations in the data. Specifically, we proposed Algorithm 1 to address the bulk ingestion problem. In addition, a calibration scheme is introduced to ensure that changes in traffic pattern will not cause over-estimation. The proposed solution has enabled us to fulfill multiple emergency requests for data loss estimations. We can return the loss report for any particular customer in a matter of minutes, which ensures that we are ready to talk with any customer after each CSO. In addition, We can finish the loss estimation of all our customers (tens of thousands of report suits) within hours, so we can quantitatively evaluate the severity of each CSO right after the incident.

References

[1] https://business.adobe.com/resources/digital-economy-index.html

[2] Ho S, Xie M. The use of arima models for reliability forecasting and analysis. Computers industrial engineering. 1998; 35:213–216.

[3] Nobre F, Williamson GD. Dynamic linear model and sarima: a comparison of their forecasting performance in epidemiology. Statistics in medicine. 2001;20:3051-69.

[4] Hochreiter S, Schmidhuber J. Long short-term memory. Neural computation. 1997; 9;1735–1780.

[5] Taylor SJ, Letham B. Forecasting at scale. The American Statistician. 2018;72:37–45.