EasyChair Preprint

№ 2925

TIMESIGHT: Discovering Time-driven Insights Automatically and Fairly

Yohan Bae, Suyeong Lee and Yeonghun Nam

March 11, 2020

TIMESIGHT: Discovering Time-driven Insights Automatically and Fairly

Yohan Bae
Samsung Research, Samsung Electronics
Seoul, South Korea
yhan.bae@samsung.com

Suyeong Lee
Samsung Research, Samsung Electronics
Seoul, South Korea
sy710.lee@samsung.com

Yeonghun Nam
Samsung Research, Samsung Electronics
Seoul, South Korea
yeonghun.nam@samsung.com

***Abstract*— Exploratory data analysis (EDA) on time-series data is an indispensable and important process for not only data analysts but also non-expert users. It helps them make data-driven decisions by discovering important patterns of a certain phenomenon. However, it poses 2 challenges for data analysts and decision-makers. First, although a lot of business intelligence tools have been introduced that can help explore the data, they require repeated analytic procedures and most of the procedures rely on users intuition, knowledge and efforts. Second, even though there have been several attempts to quantify insights to automatically detect interesting patterns, they do not consider score fairness among detected patterns. Therefore, they are not suitable when data has the heterogeneity of insight types, attributes scales and time intervals. We attack these challenges by introducing our new proposed system *Timesight*, which explores data through all possible time units and all attributes automatically. *Timesight* evaluates various types of time-driven insight, matching the fairness among each type of insight, each attribute, and each time interval. We verify our system using internal application log dataset. Our experiment with data analysts working the same dataset shows that *Timesight* alleviates the tedious work and is effective in discovering insight.**

Keywords-component; Data exploration; Insight discovery; Data mining; Time-series data.

I. INTRODUCTION

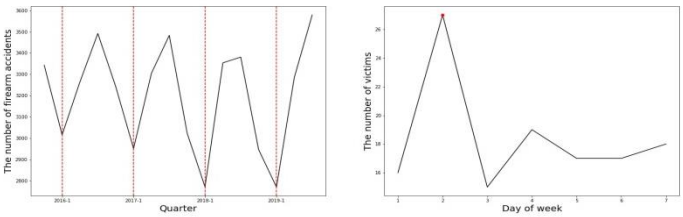
Nowadays, efforts to make data-driven decisions are continuously increasing in various industries [11] to reduce costs or improve productivity. For instance, a manufacturer who wants to increase productivity can adjust the time on the assembly line, observing delays calculated from the data. After introducing a solution to the bottleneck, the decision-maker would measure the time of the process again to see if the change results in time-saving. Thus, exploratory data analysis (EDA) is an indispensable and important process to discover potential meaning or important patterns (called *insight*) from a data [12]. Based on the EDA process, data analysts are able to understand the data to predict or prevent certain phenomenon via building statistical methods or machine learning models. It is also important for non-experts users who are in an important position and should make a critical decision from the data without any data scientific knowledge such as statistics and probability theory.

TABLE I. CRIME DATASET

timestamp	district	# of victims	crime category
2019-09-29 06:39:00	A3	12	firearm
2018-12-31 18:42:00	B2	24	knife
2017-03-05 13:15:00	C1	5	vehicle

Among many kinds of a dataset, time-series data that contains time information about certain events is more valuable than other simple multi-dimensional data. Because, it is important to discover striking patterns of specific phenomena or events over various time units (e.g., year, month, weekday, etc.) in many domains such as business, manufacturing, healthcare, economy, sociology, and even government.

Suppose that we have a crime in a city dataset with the schema (timestamp, district, number of victims, crime category) as shown in Table 1. Figure 1 presents some examples of insights. It can be very helpful for people who are looking for hidden insight in the data, if we get the information that the number of firearm accident has seasonality for each quarter (as Figure 1(a)) or the number of victims is most pronounced in the second day of week as shown in Figure 1(b).



(a) # of firearm accidents for each quarter (b) # of victims by day of week

Figure 1. Example of insights

EDA Challenge. Although lots of business intelligence tools like Tableau [9] and Qlik [10] have been introduced that can help explore the data, they require repeated analytic procedures and most of the procedures rely on users intuition, knowledge and efforts. Users have to repeat trial-and-error procedures: building their own hypothesis, selecting appropriate attributes and possible time units, entering the

formula, plotting results with the suitable visualization and exploring remarkable patterns (e.g., trend, spike point). It has the advantage of offering a high degree of freedom for advanced users. However, it`s mentally and physically tedious and exhausting for both non-expert users and professionals in data analysis. If the data has more than thousands of attributes and millions of records, it becomes impossible to evaluate all the assumptions, regardless of users' expertise.

Fairness Challenge. Even though there have been several attempts [1] [2] to quantify insights to automatically detect interesting patterns, they do not consider score fairness among detected patterns. It is difficult to match the fairness because each type of insight needs different score function to calculate interestingness, most real-world dataset has different scales and units among attributes, and each time unit might have different intervals. Thus, it is challenging to provide fair scores that users can easily and reasonably accept.

We attack these challenges by introducing our new proposed system *Timesight*, which explores data through all possible time units and all attributes automatically. *Timesight* defines and evaluates various types of time-driven insight, matching the fairness among each type of insight, each attribute, and each time interval. Therefore, the contributions of this paper are:

- We demonstrate the way that automatically prepares the time-series data to well extract hidden insights.
- We propose the normalization technique to make them fairly comparable among diverse time intervals and attribute scales.
- We define 4 types of time-driven insight and unified formulation of each type to assess the magnitude of interestingness.

The rest of the paper is organized as follows: Section 2 presents an overview of our data modeling procedure. Section 3 provides 4 types of insight and score functions and Section 4 describes the pseudo-code of *Timesight* and optimization techniques. Section 5 demonstrates our experiment using real-world dataset. Section 6 discusses related work, followed by the conclusion and future work in Section 7.

II. DATA MODELING

In this section, we present the data preparation procedures to calculate data insights. It is assumed that a multi-dimensional dataset D is given as a tabular format consisting of a series of rows, and each row is represented by a set of attributes (columns). We assume that D contains 3 types of attribute sets T, N and C: $T=\{t_1,t_2,...,t_\alpha\}$ is a timestamp attribute set, $N=\{n_1,n_2,...,n_\beta\}$ is a numerical attribute set, and $C=\{c_1,c_2,...,c_\gamma\}$ is a categorical attribute set where α, β and γ are the number of timestamp, numerical, and categorical attributes, respectively. In this paper, the data modeling process is divided into two phases: *timestamp decomposition* and *data aggregation and normalization*.

Timestamp Decomposition. Suppose that each timestamp attribute has 'yyyy-MM-dd HH:mm:ss' format. we define a set $O=\{o_1,o_2,\cdots,o_\delta\}$ that includes extracted time units according to the analytics objective (in this section, it is assumed that there is one timestamp attribute in D for the convenience of derivation). We illustrate an example in Figure 2 where the original timestamp attribute, and the extracted attributes are in Figure 2(a) and Figure 2(b), respectively. In this paper, we define 7 (i.e., δ=7) extracted attributes for each timestamp attribute, i.e., O={'yyyy', 'MM', 'dd', 'HH', 'yyyy-qq (year-quarter)', 'yyyy-MM', 'yyyy-MM-dd'}. The elements of O depend on the analytics objective and can be declared dynamically (e.g., 'yyyy HH', 'HH:mm', etc.).

timestamp
2019-09-29 06:39:00
2018-12-31 18:42:00
2017-03-05 13:15:00

(a)

Year	month	day	hour	year-quarter	year-month	year-month-day
2019	09	29	06	2019-3	2019-09	2019-09-29
2018	12	31	18	2018-4	2018-12	2018-12-31
2017	03	05	13	2017-2	2017-03	2017-03-05

(b)

Figure 2. A timestamp decomposition example of (a) the original attribute, and (b) the extracted attributes from (a).

Data Aggregation and Normalization. The data aggregation and normalization techniques are applied to the dataset to make it fairly comparable among diverse time units and types of insight. In this paper, numerical as well as categorical attributes are used for insight scoring (which is different in that related works only consider numerical attributes) because pattern of categorical attributes can contain important meaning after appropriate aggregations. For the available aggregation functions $agg \in \{SUM, AVG, COUNT, \cdots\}$, we consider agg for the categorical and the numerical attributes separately, because the available aggregations for the categorical and numerical attributes are different. For example, COUNT and PERCENTAGE are for categorical, while SUM and AVG are for numerical attributes. For the arbitrary time unit element o_l in O ($1 \le l \le \delta$), the aggregated datasets can be obtained based on two cases.

Case1: For numerical attributes, the function $GN(o_l,n_j)$ *g*roups D by o_l with certain aggregation on the attribute n_j, which is presented as follows:

$$GN(o_l,n_j) \approx \text{SELECT } agg(n_j) \text{ FROM D GROUP BY } o_l$$

Case2: For categorical attributes, for the arbitrary *i*th categorical attribute, let E_i denote the set of distinct elements of the c_i, assuming that $|E_i| \ge 1$ and $e_{i,m}$ the arbitrary *m*th element of E_i ($1 \le m \le |E_i|$). The function $GC(o_l,e_{i,m})$, which filters D with value $e_{i,m}$ and groups D by o_l with certain aggregation on the attribute c_j can be presented as follows:

$GC(o_l,e_{i,m}) \approx$ SELECT $agg(c_j)$ FROM D WHERE $c_j = e_{i,m}$ GROUP BY o_l

Thus, from the given $GN(o_l,n_j)$ and $GC(o_l,e_{i,m})$, the result set X can be derived as:

$$X = \begin{cases} GN(o_l, n_j) & \text{(for the numeric attributes)} \\ GC(o_l, e_{i,m}) & \text{(for the categorical attributes)} \end{cases} \text{ on D}$$

We obtain aggregated result set $X=\{x_1,x_2,\cdots,x_n\}$ from each categorical and numerical attribute considering multiple time unit elements. Next, we normalize all values in the X using min-max normalization [3]. It maps all values to the range [0, 1], and helps us focus on the relative ratio, improving the balance among other result sets that are measured by different units. It also enhances fairness with different insight types. There is another well-known normalization method called standardization (or Z-score normalization) [3], but it creates new data not bounded to certain interval. Consequently, we get a normalized result set X_{norm} from X. An example of normalization is illustrated in Figure 3 where Figure 3(a) is the original result set X and Figure 3(b) is the normalized result set X_{norm}

$$X_{norm} = \left\{ x_{norm,i} : \frac{x_i - x_{min}}{x_{max} - x_{max}}, i = 1, 2, 3, \dots \right\}$$

5	21	-1	32	-11	13	0	-7	19	29

(a)

0.37	0.74	0.25	1.0	0.0	0.56	0.26	0.09	0.70	0.93

(b)

Figure 3. A normalization example of (a) the original result set, and (b) the normalized result set from (a) which is bounded in range [0, 1].

III. INSIGHT SCORE FUNCTIOIN

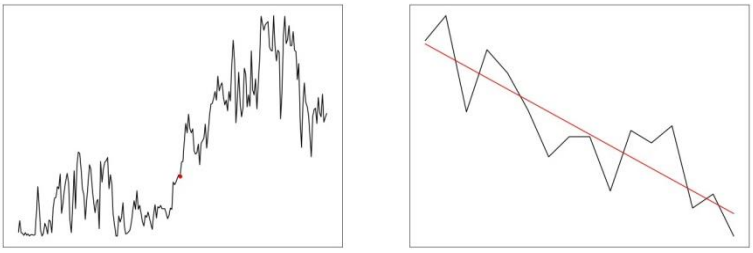
We define 4 types of time-driven insight: *spike point*, *change point*, *seasonality*, and *trend*. We want to score and rank interestingness of each insight based on the *p*-value to discover important insight from the entire result set. In statistics, the *p*-value is the probability of the observation from the null hypothesis and commonly used to determine whether an observation is statistically significant [4]. We use different kinds of null hypotheses for different types of insight to calculate the appropriate *p*-values. In this paper, we use the Gaussian distributions $N(\mu,\sigma^2)$ [5], where μ and σ^2 are constant parameters to model the distribution of observational values of each insight type. We use $\mu=0$ and $\sigma^2=1$ for all types of insights for convenience, but each can be replaced with the appropriate parameters in future studies.

Spike point. The Spike point means that a certain value in X_{norm} represents a notable difference from others. Data analysts who are exploring a data are attracted by the significant deviations from predictable patterns. The spike point can be discovered by measuring how a certain point is noticeable over other values. The distribution of various domains, such as physics, biology, economics, social science, and other numerous man-made phenomena often follow power-law distribution [6]. Therefore, we set the null hypothesis of spike point as:

H_0: X_{norm} follows power-law distribution.

Let $X=\{x_{norm,1},x_{norm,2},\dots,x_{norm,n}\}$ be the set of values in the result of aggregation as mentioned previous section. We sort X_{norm} in the descending order and get the maximum value $x_{norm,max}$. Then, from the $x_{norm,max}$, we evaluate the magnitude of interestingness against the hypothesis H_0 is true. Hence, we fit values in $X_{norm} \setminus \{ x_{norm,max} \}$ to the power-law distribution (i.e., $a \cdot i^{-b}$). Next, from the $x_{norm,max}$'s prediction error $e_{\max} = x_{norm.max} - \hat{X}_{\text{norm,max}}$, we calculate *p*-value $p_{spike} = P(e > e_{max} | e \sim N(\mu,\sigma^2))$. Consequently, the score of spike point is $1 - p_{spike}$

$$score_{spike} = 1 - p_{spike}$$



(a) Change point (b) Trend

Figure 4. Examples of change point and trend.

Change point. A change point generally indicates an abrupt variation of values between the previous and subsequent intervals [17]. Figure 4(a) shows an example of change point. We use mean value on the window size *K* to obtain representative value. Thus, the null hypothesis of change point is:

H_0: Difference of mean between before and after $x_{norm,i} \approx 0$.

Let $X=\{x_{norm,1},x_{norm,2},\dots,x_{norm,n}\}$ be the time series of values and assume length of window size, $K < n/2$. By shifting the window along the values, we calculate the mean for values in left and right window of size *K* respectively. And their difference d_i is as follows:

$$\text{mean}_{\text{left,i}} = \frac{1}{K}\sum_{j=i}^{i+K-1} x_{norm,j},$$
$$\text{mean}_{\text{right,i}} = \frac{1}{K}\sum_{j=i+K}^{i+2K-1} x_{norm,j},$$

$$d_i = |mean_{left,i} - mean_{right,i}|,\ 1 \le i \le n - 2K + 1.$$

Let d_{max} denote the maximum difference value. Then, we calculate *p*-value $p_{change} = P(d > d_{max} \mid d \sim N(\mu,\sigma^2))$. Thus, the score of change point is

$$score_{change} = 1 - p_{change}$$

Seasonality. If the data show repetitive patterns or fluctuation over a specific period of time, we can say it has seasonality. We use the autocorrelation function (ACF) because it is commonly used to determine whether the data has a dependency on its past [7]. If the strongest correlation appears at particular period *p* for a given o_l (e.g., 4 for 'yyyy-qq', 12 for 'yyyy-mm', etc.), we determine it has seasonality. Thus, we set the null hypothesis as follows:

H_0: $a \in$ ACF(X_{norm}) has maximum value a_{max} at p.

Then, the p-value is $p_{seasonality} = P(a > a_{max} \mid a \sim N(\mu,\sigma^2))$. As a result, the score of seasonality is $1 - p_{seasonality}$

$$score_{seasonality} = 1 - p_{seasonality}$$

Trend. It indicates that the data show continuously rising or falling movement over time, like in Figure 4(b). Those who want to discover explainable patterns in the data are well obsessed when it has a very different slope from 0. Thus, we set the null hypothesis as:

H_0: Slope of the values in X_{norm} over entire time ≈ 0.

Frist, we fit X_{norm} to a line by linear regression as shown in Figure 4(b). We also normalize the x-axis values using min-max normalization, for fairness with another result set. As a result, we can concentrate on the change of values, not the length (time interval) of the data. Then we calculate its slope s^* and coefficient of determination, r^2. The r^2 represents how well the line fits the data [8]. Also, we compute the p-value as $p_{trend} = P(s > |s^*| \mid s \sim N(\mu,\sigma^2))$. Finally, we can obtain the trend score $r^2 * (1 - p_{trend})$ where r^2 is used to reflect the accuracy of the regression.

$$score_{trend} = r^2 * (1 - p_{trend})$$

IV. FRAMEWORK

In this section, we firstly describe the full procedure of *Timesight* using pseudo-code, then discuss the pruning-based optimization techniques that can reduce search space and running time, improving overall performance of *Timesight*.

Algorithm 1 InsightDiscovery(T, N, C)

```
1:  max-heap ℍ ← {}
2:  O ← extract all possible time units from T
3:  for o_l in O do
4:      for n_j in N do
5:          X_norm ← normalize(GN(o_l, n_j))
6:          CalculateInsights(X_norm, ℍ)
7:      for c_i in C do
8:          for e_i,m in c_i do
9:              X_norm ← normalize(GC(o_i, e_i,m))
10:             CalculateInsights(X_norm, ℍ)
11: return ℍ
```

Function: *CalculateInsights*(*X*, ℍ)

```
12: for each insight type I do
13:     score_I ← calculate_I(X)
14:     insert (score_I, X) to ℍ
```

A. Psuedo Code

The Algorithm 1 presents the full procedure of our insight discovery system. We assume that there is one timestamp attribute in D for the convenience of derivation.

First, we initiate the max heap ℍ to store insights in descending order (Line 1) and extract all possible time units (Line 2). Then iterating over all time units (Line 3), we repeatedly generate X_{norm} to score insights for all numerical attributes N (Lines 4, 5). At the same time, we generate X_{norm} for all categorical attributes C (Lines 7-9). Using the generated result set X_{norm}, function '*CalculateInsights*' calculates scores of all insight types and updates ℍ (Lines 12-14).

B. Pruning-Based Optimization Technique

Searching and computing all possible time units and all attributes takes a lot of time and degrade performance. Therefore, we suggest the three pruning methods that can save time performance.

1) We pass calculating the score if multiple X_{norm} sets are identical for different time units. For instance, if the data is only for 2019, the result sets of '*yyyy-mm*' and '*mm*' have the same values. This can be applied equally on '*yyyy-qq*' with '*qq*', '*yyyy-MM-dd*' with '*dd*' and so on.

2) If the length of X_{norm} is too short, the data cannot represent a particular pattern properly. As a result, we set the minimum length ζ (e.g., ζ = 4, because '*qq*' can have a maximum length of 4.) and if the length of X_{norm} is shorter than ζ, then we do not calculate all insight score.

3) If $|E_i|$ is too large, the search space grows exponentially and the performance is degraded. Also, if $|E_i| = 1$, it may not meaningful to apply aggregation on c_i. Consequently, we set the minimum and maximum length θ (e.g., 70) and calculate score only if $1<|E_i|<\theta$.

V. EXPERIMENT

TABLE II. SUMMARY OF EXPERIMENT DATA

Range of date	2009.06.26 00:29:12 ~ 2019.07.12 23:50:02
# of rows	100000
# of numerical attributes	10
# of categorical attributes	49

In this section, we apply the real-world time-series data to evaluate the effectiveness of *Timesight*. This data is an internal application log dataset that is de-identified for research purposes. The summary of the dataset is shown in Table 2. The data has a timestamp attribute that represents the access date for each user from 2009 to 2019. And 10 numerical attributes and 49 categorical attributes contain a variety of information.

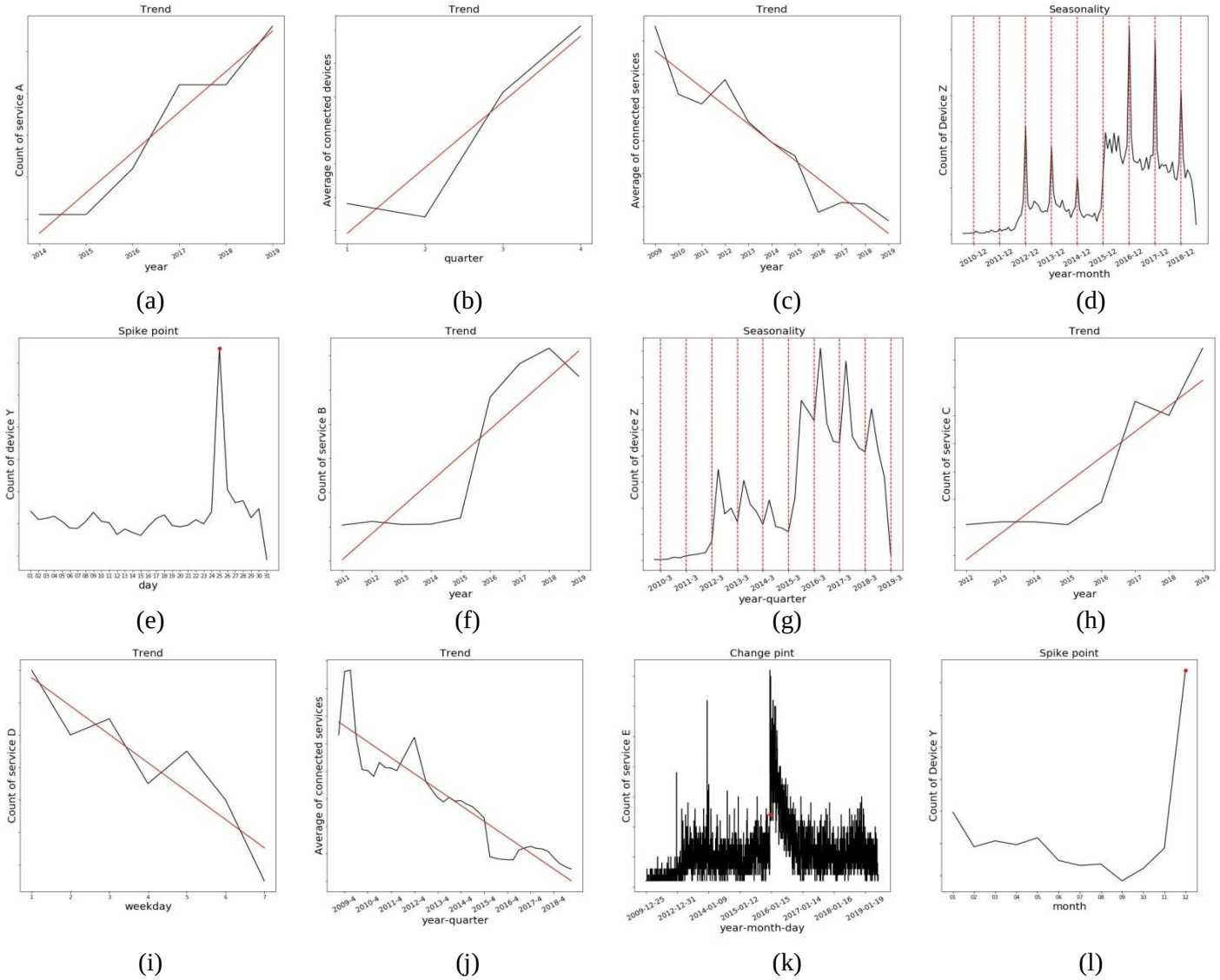


Figure 5. Result of our experiments. The x-axis is each time unit and the y-axis is each aggregated result set. The black lines indicate actual result set, and the red lines indicate fitted trends or periods of seasonality. The red points indicate spike points or change points.

Figure 5 presents 12 result insights from *Timesight*. *Timesight* analyzes time stamp attributes by decomposing it in many ways such as *year*, *quarter*, *year-quarter*, *month*, *year-month*, *day*, *weekday*, *year-month-day* and so on. As a result, users can examine the data from various time perspectives. Each of A, B, C, D, and E is a service name in the application and each Y, Z is a device name that user used to access the application.

Figure 5(d) and Figure 5(g) show that the number of people who accessed the application using device Z had a clear seasonality for *year-month* and *year-quarter*. Also, we can see that most people accessed the application using device Y on the 25th of every month and every December, as shown in Figure 5(e) and Figure 5(l), respectively. Furthermore, Figure 5(k) represents that the number of a person who signed in through service E had a significant change point before and after 2016-01-15. The rest of the Figure 5 shows us the clear tends of each aggregated attribute for *year*, *quarter*, *year-quarter*, *weekday*.

It is very convenient for analysts and decision-makers in the aspect of getting these insights from the data without the effort to explore it in person. Furthermore, they can make immediate decisions like: 1) From Figure 5(d) and Figure 5(g) they can try to find out the reason that the usage of device Z has seasonality and is most prominent in December and look for a solution that can balance monthly usage to improve overall usage. 2) From Figure 5(k), they can investigate why service E shows dramatic changes in usage around 2016-01-15 and apply that factor on other services. 3) From Figure 5(i), decision-maker can promote about service D on weekends so that the usage of service D does not decrease on weekends. Analysts who are working on the same dataset observed the effectiveness of results in alleviating their tedious works and discovering meaningful insights.

VI. RELATED WORK

In this section, we discuss prior works from multiple areas related to our research. We review the related works and describe how they differ from *Timesight*.

Business Intelligence Tools. Business intelligence tools, such as Tableau [9], Qilk [10], and Sisense [13] have recently improved capabilities in EDA and gained in popularity. These tools allow data analysts and decision-makers who lack programming skills to easily select attributes and build visualizations based on their abilities. However, they require repeated analytic procedures and most of the procedures rely on users' intuition, knowledge and efforts. Furthermore, if the data has more than thousands of attributes, it becomes impossible to evaluate all the assumptions regardless of users' expertise. On the contrary, *Timesight* discover insight automatically, exploring all attributes and all time units based on unified formulations of various insight types. Therefore, *Timesight* alleviates the tedious trial-and-error process of users.

Automated Exploratory Analysis. There have been several researches that attempt to quantify insights to automatically detect interesting patterns. The SeeDB [14], which is the visualization recommendation system, identified charts that are largely deviated from a given reference and considers them as insight. But, it is difficult to use for not-expert users or service manager because they have to select and put queries in person. Whereas *Timesight* extracts insight automatically using unified formulations of various insight types. In Foresight [1], the authors defined about 6 insight types and their score functions to facilitate the rapid discovery of insights from large, high-dimensional datasets. However, first, because they tried to define insights into the general attribute domain, they did not consider time-driven insights such as trends, change point, seasonality. Also, they did not consider the difference between each attribute`s scales that might affect a huge effect on calculated insight scores. QuickInsight [2] [15] which is the most recent research, is automatic insight discovering system released in Microsoft Power BI [16]. QuickInsight proposes a unified formulation of important patterns, and introduce an insight mining framework to automatically mine insight from given data. However, it also did not consider fairness among detected patterns because of heterogeneity of attributes scales, time intervals, and formulations. On the other hand, *Timesight* normalize attributes and time intervals to provide fair scores that users can easily and reasonably accommodate.

VII. CONCLUSION AND FUTURE WORK

We introduce a novel approach to automatically discover interesting insight from multi-dimensional time-series data to offer invaluable hidden information to both data analysts and decision makers. We decompose a timestamp attribute in several ways to examine the data at various time perspectives, and use both numerical attributes and categorical attributes as targets by applying appropriate aggregations. And we normalize values in the result set to obtain a fair score between each insight type, each attribute and even each time interval. And then, we define 4 types of time-driven insight and unified score functions to assess the interestingness of each result set. Furthermore, we demonstrate *Timesight* using pseudo-code and propose several pruning techniques to improve the performance of *Timesight*. Lastly, we present our experimental result based on internal service log dataset that help data analysts to discover hidden insight easily.

We want to advance this research through some direction of future work. First, we will develop and supplement additional types of insight such as the correlation between different *X*s. Second, we use uniformed μ and σ^2 for distributions of all insight types in the paper for convenience. But we will investigate lots of datasets to find appropriate μ and σ^2 for each type of insight. Lastly, the limitation of our system is that as the number of rows and attributes in the dataset increase, the search space is extended together. This can increase the time and space it takes to calculate the scores. We need and advanced optimization method to solve these problems.

REFERENCES

[1] Demiralp. C., Haas. P.J., Parthasarathy. S., and Pedapati. T., "Foresight: Recommending visual insight", Proceedings of the VLDB Endowment, Vol. 10, No. 12, pp. 1937-1940, 2017.

[2] Tang. B, Han. S., Yiu. M.L., Ding. R., and Zhang. D., "Extracting Top-K Insights from multi-dimensional Data", In: Proceeding of 2017 ACM International Conference on Management of Data (SIGMOD `17), pp. 1509-1524. ACM, New York, NY, USA, 2017.

[3] Patro. S.G.K., and Sahu. K.K., " Normalization: A Preprocessing Stage", International Advanced Research Journal in Science, Engineering and Technology, Vol. 2, No. 3, pp. 20-22, 2015.

[4] Krzywinski. M., Altman. N., "Points of significance: Significance, p values and t-tests." Nature methods, Vol. 10, pp. 1041-1042, 2015.

[5] Lyon, A., "Why are Normal Distributions Normal?." The British Journal for the Philosophy of Science, Vol. 65, No. 3, pp. 621-649, 2014.

[6] Newman M. E. J., "Power laws, Pareto distributions and Zipf's law." Contemporary Physics, Vol.46, No.5, pp323-351, 2005.

[7] Nopia. Z.M., Lennie. A., Abdullah S., Nuawi. M.Z., Nuryazmin. A.Z., and Baharin. M.N., "The use of autocorrelation function in the seasonality analysis for fatigue strain data." Journal of Asian Scientific Research, Vol. 2, No. 11, pp. 782-788, 2012.

[8] Hamilton. D.F., Ghert. M. and Simpson. A.H., "Interpreting regression models in clinical outcome studies." Bone Joint Res, Vol. 4, No. 9, pp. 152-153, 2015.

[9] Tableau Homepage, https://www.tableau.com/, last accessed 2019/11/14.

[10] Qlik Homepage, https://www.qlik.com, last accessed 2019/11/14.

[11] Brynjolfsson. E., McElheran. K., "The Rapid Adoption of Data-Driven Decision-Making." American Economic Review, Vol. 106, No. 5, pp. 133-139, 2016.

[12] Yu, C.H., "Exploratory data analysis." Methods 2, 2017, pp. 131-160.

[13] Sisense Homepage, https://www.sisense.com/, last accessed 2019/11/14.

[14] Vartak. M., and Rahman. S., Madden. S., "SeeDB: efficient data-driven visualization recommendations to support visual analytics." Proceedings of the VLDB Endowment, Vol. 8, No. 13, pp. 2182-2193, 2015.

[15] Ding. R., Han. S., Xu. Y., Zhang. H., and Zhang. D., "QuickInsights: Quick and Automatic Discovery of Insights from Multi-Dimensional Data." In: Proceeding of the 2019 International Conference on Management of Data (SIGMOD `19), pp. 317-332. ACM, New York, NY, USA, 2019.

[16] Microsoft Power BI Homepage, https://powerbi.microsoft.com/, last accessed 2019/11/14.

[17] Aminikhanghahi. S., and Cook D.J., "A Survey of Methods for Time Series Change Point Detection." Knowledge and information systems, Vol. 51, No. 2, pp. 339- 367, 2017.