# Appendix: Actionable Insights for Multivariate Time-series for Urban Analytics

**Anika Tabassum**[1 3], **Supriya Cinthavali**[2], **Varisara Tansakul**[2], **B. Aditya Prakash**[3]
[1]Department of Computer Science, Virginia Tech
[2]Oak Ridge National Laboratory
[3]College of Computing, Georgia Tech
[1]anikat1@vt.edu, [2]{chinthavalis,tansakulv}@ornl.gov, [3] badityap@cc.gatech.edu

## 1 Additional Details on Sec 3

### 1.1 Proof of Lemma 3.1

*Proof.* Following is the proof for different $i$, $j$, and $k$.

- **Case** $i = 1, k = T$: Trivial. Since, there is only on segment from 1 to $T$ and only one path.
- **Case** $i = 1, k < T$: (By induction). Suppose, $T = k + 1$, then the number of paths is $2^{k+1-k-1} = 2^0 = 1$. This is trivial, because from $s_{jk}$ there is only one possible edge $s_{jk} - s_{k(k+1)}$ and thus there is only one possible path.
  Suppose, with $T = k + q$ where $q > k$, the number of paths is $2^{q-1}$. We need to show, with $T = k + q + 1$, the number of paths become $2^q$. Since $q+1 > q$, all the paths that can be possible to reach from $s_{jk}$ to $s_{j(k+q)}$ can also be possible to reach from $s_{jk}$ to $s_{j(k+q+1)}$ using the edge $s_{j(k+q)} - s_{j(k+q+1)}$. That is, $2^{q-1}$ paths possible from $s_{jk}$ to $s_{j(k+q+1)}$ using the edge $s_{j(k+q)} - s_{j(k+q+1)}$.
  Again, by the edge construction property of $G$, a path ends when end timestamp of a node is $T = k + q + 1$. If we remove the timestamp $k + q$ and consider only $k, k + 1, \dots, k + q - 1, k + q + 1$, then the number of possible paths become $2^{q-1}$ to reach $k + q + 1$ (by induction). Because, all edges that used nodes ended with $k + q$ now can use nodes ended with $k + q + 1$.
  Thus, the total number of possible paths with and without using the edge $s_{j(k+q)} - s_{j(k+q+1)}$ becomes $2^{q-1} + 2^{q-1} = 2^{q-1+1} = 2^q$ (proved).
- **Case** $i > 1, k = T$**:** Following the above if we consider $T = i$ and the $k = 1$. The number of paths is $2^{T-k-1} = 2^{i-1-1} = 2^{i-2}$.
- **Case** $otherwise$**:** Hence, if there are $2^{i-2}$ paths possible to reach a node $s_{ij}$ and $2^{T-k-1}$ possible paths to reach timestamp $T$ from a node $s_{jk}$. The total number of paths possible through the edge $s_{ij} - s_{jk}$ or $e_{ijk}$ is $p_{ijk} = 2^{(i-2)+(T-k-1)}$.

□

### 1.2 Proof of Corollary 3.1

*Proof.* According to the path property of $G$, every path starts from $i = 1$ and ends at timestamp $T$. For $i = 1, k = T$ there is only one path possible (the whole segmentation). For $1 < j < T - 1$, the first node of all other paths in $G$ has to start from some $s_{1j}$. Consider $j = k$, using Lemma **??** we find $p_{1kk}$. Thus, $K = 1 + \sum_{k=2}^{T-1} p_{1kk} = 1 + 2^{T-2} - 1 = 2^{T-2}$. □

### 1.3 Proof of Lemma 3.2

*Proof.* We discuss separate proof for time and space complexity.

**Time complexity:** (i) *Case serial:* The first term is to calculate $\pi_{\text{rest}}$ for each $e_{ijk}$. Total $e_{ijk}$ in $G$ is $O(T^3)$. The second term is to solve $\alpha$ using Gradient Descent learning, and the third term is to solve $\mathbf{r}_j$ for every cutpoint $c_j$.

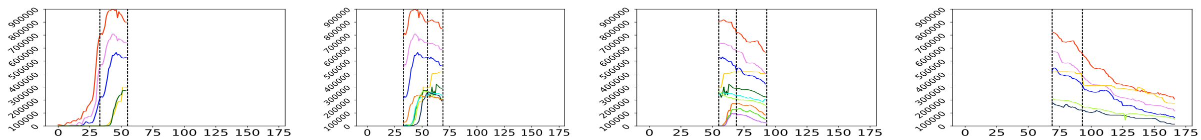
(ii) *Case Parallel:* Parallelization can be applied in Find-RaTSS for $\pi_{\text{rest}}$ computation, then the total time complexity will be distributed among $n$ processors and hence this is $O(\frac{T^3}{n} mf)$. However, time complexity of gradient descent and $\mathbf{r}_j$ computation is similar as in serial cases.

**Space complexity:** (i) Case serial: The first term $O(m)$ is to store $\pi_{\text{rest}}$ and $\pi_B$. Since we need $O(m)$ $\pi_{\text{rest}}$ and $O(m)$ $\pi_B$ for $m$ time-series. Hence $O(m) + O(m) = O(2m) \approx O(m)$.

(ii) Case parallel: To store final computation of $\pi_{\text{rest}}$ and $\pi_B$ it takes $O(m)$. Whereas the second term $O(mn)$ is to store temporary $\boldsymbol{\pi}_{\text{rest}}$ computation for each processor among $n$ processors, $O(mn)$ is the number of possible edges in a segment graph $G$. Hence temporary space for $n$ processor and final computation of $\boldsymbol{\pi}_B$ and $\boldsymbol{\pi}_{\text{rest}}$ is $O(mn) + O(m) = O(m + mn)$. □

## 2 Additional Details on Sec. 4

**Additional experiments: Feature ablation test** For feature robustness by we use feature ablation test. Tbl. 1 shows how rationalization affected on the Synthetic, ChickenDance and GrandMal dataset by removing each feature used in RaTSS.




(a) Irma $c_1$ (b) Irma $c_2$ (c) Irma $c_3$ (d) Irma $c_4$

(e) Irma $r_1$ (f) Irma $r_2$ (g) Irma $r_3$ (h) Irma $r_4$

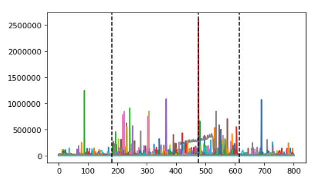
Figure 1: **Find-RaTSS finds the most affected counties facing damage. The rationalizations $\mathbf{r}_j$ for corresponding $c_j$ are mapped in US county map (Higher $r_j^u$ with brighter red).**

Table 1: **Feature Ablation test in terms of F1-score for Ground-truth datasets.** $f_1$**: mean,** $f_2$**: variance,** $f_3$**: min,** $f_4$**: max**

| Dataset | Features removed | | | |
|---|---|---|---|---|
| | $f_1$ | $f_2$ | $f_3$ | $f_4$ |
| *Gaussian* | 0.52 | 0.44 | 0.49 | 0.64 |
| *ChickenDance* | 0.73 | 0.86 | 0.401 | 0.643 |
| *Great Barbet* | 0.5 | 0.5 | 0 | 1.0 |

## 2.1 Additional Case-studies: Hurricane Irma

Fig. 1 shows our rationalizations for Hurricane Irma.




(a) Time-series with segmentation

| Rationalizations for $\mathbf{c_1}$ | Avg. traffic across $\mathbf{c_1}$ |
|---|---|
| Calculator | 6819 |
| Ahrar Ah-shams | 4589 |
| Christopher Paul Neil | 3850 |
| 88th academy awards | 1813 |
| World largest companies by sector | 223 |
| Joker Comics | 142 |

(b) Rationalization

Figure 2: **Find-RaTSS finds the rationalization from Wikipedia during the period 2015-2017.**

**Additional Case-study in general dataset (Wikipedia)** To understand the importance of content and improve advertisement strategies, we consider the web traffic of 3000 Wikipedia articles from 2015 July-2017 October. The segmentation mark a different season of a year (around the end of 2015, middle of 2016, and beginning of 2017 in Fig. 2(a)). We find overall, the majority of articles/rationalizations (64%) by Find-RaTSS are eventful. Fig. 2(b) shows an example of the rationalizations by Find-RaTSS across the first cutpoint $c_1$. We observe Find-RaTSS considers some low-traffic articles, e.g., 'World's largest companies by sector' within top 10 along with the high traffic ones. The reason is, 'Forbes Global 2000' for worlds largest companies was published around June 2016 after $c_1$[1]. This is not easy to find such low traffic easily by visualizing the time-series or using any other baselines from Sec. **??** (as buried with other high traffic articles).

[1]https://www.forbes.com/sites/forbespr/2016/05/25/forbes-14th-annual-global-2000-the-worlds-biggest-public-companies-2/#21ca87643c44