# Entropy-Based Anomaly Detection in Evolving Graph Streams

Satoshi Kayano
Supervised by: Koki Kawabata, Yasuko Matsubara and Yasushi Sakurai
SANKEN, The University of Osaka, Osaka, Japan
kayano88@sanken.osaka-u.ac.jp

## ABSTRACT

Given a stream of evolving graphs, such as those generated by a bike sharing service, each event can be represented as a tuple of the form (*departure*, *destination*, *time*; *count*). How effectively can we squeeze insights about user behavior? How efficiently can we maintain patterns to adapt to recent characteristics? In this paper, we present a new streaming tensor factorization approach, namely EntMine, for information-theoretic pattern discovery in large evolving graphs. Our method has the following properties: it is *scalable:* our method scales linearly with the number of events at any time to maintain important patterns and infer node anomaly scores; it is *accurate:* our method outperforms the state-of-the-art methods by accurately detecting changes in event distribution; and it is *effective:* it reveals meaningful seasonal and node-based patterns. Experiments using three real-world shared cycle datasets demonstrate that our proposed method achieves the highest accuracy in terms of AUC-PR and discovers interpretable seasonal patterns, while existing methods cannot find them.

**VLDB Workshop Reference Format:**
Satoshi Kayano. Entropy-Based Anomaly Detection in Evolving Graph Streams . VLDB 2025 Workshop: PhD Workshop.

## 1 INTRODUCTION

Given a stream of evolving graphs, such as those generated by a bike-sharing service, each event can be represented as a tuple of the form (*departure*, *destination*, *time*; *count*). How effectively can we uncover any patterns in edge (i.e., demand) distributions by leveraging their graph structural information? Discovering important components that well represent the original evolving graphs are essential to understanding user behavior, allowing us to spot abnormal events by the deviation from detected components. Considering edge directions further supports the structured relationships between nodes in more detail. [5, 16] have addressed edge-based streaming graph anomaly detection and can handle such sparse events efficiently, but they can be deceived by unfamiliar connections. Another limitation is that they can find only extreme behaviors because their criteria for rating anomalies rely on event counts. To address the aforementioned limitations, we propose a novel streaming tensor factorization method, EntMine, designed for information-theoretic pattern discovery in large evolving graphs, enabling streaming anomaly detection. Specifically, our contributions are summarized as follows:

- **Scalability:** our method scales linearly with the number of events at any time and maintains important patterns using constant memory, as mentioned in Lemma 1 and Lemma 2.
- **Accuracy:** our method surpasses the state-of-the-art baselines in detecting distributional changes in streaming graphs, achieving the best performance consistently in AUC-PR.
- **Effectiveness:** it uncovers meaningful seasonal and node-specific patterns in three real-world bicycle datasets which existing methods fail to detect.


Proceedings of the VLDB Endowment. ISSN 2150-8097.

## 2 RELATED WORK

Here, we review prior work in three key areas: (1) information-theoretic approaches for pattern discovery, (2) anomaly detection in graphs, and (3) streaming tensor and matrix factorization.

*Information-theoretic analysis.* Information and entropy are fundamental concepts in information theory and have been playing a vital role in machine learning and data mining applications [14]. Our work is closely related to studies that employ entropy for feature extraction, such as for network intrusion detection [3, 12] and fault detection [2]. PenMiner [4] introduced an entropy-based approach to measure the persistence of activities in edge streams over important communities, referred to as activity snippets.

*Streaming graph anomaly detection.* Recent deep learning methods, such as graph neural networks [7], offer expressive modeling power but suffer from high latency and retraining costs. MIDAS [5] and its variant [6] combined the chi-square test and the count-min sketch to detect abnormal micro clusters over edge streams with a theoretical error bound.

*Tensor factorization.* Tensor (or matrix) factorization is a powerful tool for uncovering latent components that represent "normal" behavior, thereby enabling unsupervised detection of anomalies. Recent advancements have enhanced its capability to handle data sparsity [13], temporal-mode constraints [8, 18], and incorporate auxiliary information [1, 9]. Streaming approaches [11, 17] have further extended their applicability to keep track of evolving components. However, none of the above approaches consider the dynamics of entropy in graph streams.

## 3 PROPOSED METHOD: ENTMINE

In this section, we propose an efficient online tensor factorization algorithm, namely, EntMine, that realizes information-theoretic anomaly detection on evolving graphs.

**Dataset and notations.** In this paper, we consider an endless event stream of $\mathbf{A}(t) \in \mathbb{N}^{n \times n}$, which continuously arrives at every time point $t$. We assume that the number of nodes $n$ is fixed over time, and each element $a_{ij} \in \mathbf{A}(t)$ shows the number of directed events;

thus, $\mathbf{A}(t) \neq \mathbf{A}(t)^T$, $\exists t, t \in \{1, 2, \ldots, \infty\}$. We follow the Matlab-like notation. $\mathbf{A}_{i,j}$ denotes the $(i, j)$-th element of matrix $\mathbf{A}$, whereas $\mathbf{A}_{i,:}$ and $\mathbf{A}_{:,j}$ denote the $i$-th row and $j$-th column of $\mathbf{A}$, respectively.

**Streaming entropy extraction.** The first goal is to extract entropies from a slice of the original stream $\mathbf{A}(t)$. Given a distribution $P = \{p_1, p_2, \ldots, p_n\}$, where $p_i$ denotes the probability of the $i$-th outcome, the entropy [10] is defined as $H(P) = -\sum_{p_i \in P} p_i \log_2 p_i$, with the convention that $0 \log_2 0 = 0$. We normalize them by the maximum value, $\log_2 n$, the value when the distribution is uniform.

Since $\mathbf{A}(t)$ is the adjacency matrix of a directed graph, we can calculate entropies for two directions. The procedure vectorizes the original matrix in two ways, which preserves the structural relationships of a given graph while relaxing certain connection information by utilizing entropies. Concerning the sparsity of $\mathbf{A}(t)$, we further propose to augment the number of events for empirical distributions by introducing the number of lookback steps $h$. However, consecutive graphs evolve with several unknown factors, making it harder to find underlying patterns by mixing their edge counts. Instead, we assume the periodicity of length $p$, a user-defined hyperparameter, and exploit the past $h$ seasons. Specifically, $\mathbf{A}^{(i)}(t)$ denotes the cumulative sum of the graphs in the past $i$ seasons, i.e., $\mathbf{A}^{(i)}(t) = \sum_{j=1}^{i} \mathbf{A}(t - (j-1)p)$, where $i = \{1, \ldots, h\}$. Calculating the two-way entropies with each $\mathbf{A}^{(i)}(t)$ enables the robust multi-scale analysis of how events distribute. Overall, we will obtain $h$-dimensional entropy vectors for $n$ nodes at a time point $t$.

Definition 1 (Entropy matrix). *Let $\mathbf{H}(t) \in \mathbb{R}^{n \times h}$ be the slice of the entropy tensor at time point $t$, where each element is a normalized entropy, i.e., a nonnegative real value ranging from 0 to 1. We use $\mathbf{H}^{(s)}(t)$ and $\mathbf{H}^{(e)}(t)$ to denote entropy matrices for outgoing and incoming events, respectively.*

It can be incrementally computed without any overhead by retaining the most recent events, as defined below.

Definition 2 (Current tensor). *Let $\mathcal{A}(t) \in \mathbb{N}^{n \times n \times hp}$ be the current tensor, which is composed of adjacency matrices during the last $h$ seasons, each of length $p$.*

**Tensor factorization and online component update.** Given a couple of entropy matrices, $\mathbf{H}^{(s)}(t)$ and $\mathbf{H}^{(e)}(t)$, how can we discover important patterns hidden in them? We propose a factorization model tailored for the coupled matrices as well as periodic pattern discovery. For simplicity, we first consider approximating an entropy matrix with three compact matrices as follows.

$$\mathbf{H}(t) \approx \mathbf{U}(t)\mathbf{W}(t)\mathbf{V}(t)^T. \tag{1}$$

$\mathbf{U} \in \mathbb{R}_+^{n \times k}$ and $\mathbf{V} \in \mathbb{R}_+^{h \times k}$ summarize node and hierarchical patterns with $k$ components. $\mathbf{W} \in \mathbb{R}_+^{k \times k}$ is a diagonal matrix consisting of $k$ multipliers for seasonal adjustments. So, we retain $p$ distinct $k$-dimensional vectors to produce $\mathbf{W}$ for corresponding seasons. We impose nonnegativity constraints on all the components because the input entropies are still nonnegative values, and nonnegativity improves the interpretability of the components. Next, to apply the factorization to our bi-modal entropy matrices and get more concise yet reasonable components, we take advantage of the coupled matrix factorization framework. Specifically, the node component is shared for modeling both of the entropy matrices to reduce the time and memory complexity to solve and make it easier to understand their behavior. In contrast, it is a natural assumption that seasonal patterns on outgoing and incoming events should be modeled separately. The hierarchy component should also follow the assumption because its structure is derived from the seasonal aspect. Therefore, we define the full parameter set as $\mathcal{M} = \{\mathbf{U}, \mathbf{V}^{(\mathbf{s})}, \mathbf{V}^{(\mathbf{e})}, \mathbf{W}^{(\mathbf{s})}, \mathbf{W}^{(\mathbf{e})}\}$ to model entropy metrices $\mathbf{H}^{(s)}(t)$ and $\mathbf{H}^{(e)}(t)$, simultaneously. Our goal is to find $\mathcal{M}(t)$ that can minimize the errors between the predicted values by $\mathcal{M}(t)$ and the two entropy matrices. We define the objective function $\mathcal{L}$ as follows.

$$\mathcal{L}(\mathcal{M}(t)) = \sum_{r \in \{s,e\}} \|\mathbf{H}^{(r)}(t) - \mathbf{U}(t)\mathbf{W}^{(r)}(t)\mathbf{V}^{(r)}(t)^T\|_F \tag{2}$$

This is a coupled matrix/tensor factorization problem [1] over the node components $\mathbf{U}(t)$ and can be solved by using gradient-based optimization techniques. In a streaming setting, we assume the two components $\mathbf{U}(t)$ and $\mathbf{V}(t)$ smoothly evolve, i.e., $\mathbf{U}(t) \approx \mathbf{U}(t-1)$ and $\mathbf{V}(t) \approx \mathbf{V}(t-1)$, while the seasonal components $\mathbf{W}(t)$ is similar to the past same season, i.e., $\mathbf{W}(t) \approx \mathbf{W}(t-p)$. Letting $\gamma > 0$ be a small learning rate, the gradient update to one of $k$ components $\mathbf{u}_{\text{new}}$ or $\mathbf{v}_{\text{new}}$ is given by differentiating Equation (2) with respect to the corresponding parameters as:

$$\mathbf{u}_{\text{new}} \leftarrow \mathbf{u}_{\text{prev}} + \frac{\gamma}{2} \sum_{r \in \{s,e\}} (\mathbf{H}^{(r)} - \mathbf{U}_{\text{prev}}\mathbf{W}_{\text{prev}}^{(r)}\mathbf{V}_{\text{prev}}^{(r)\,T})\mathbf{v}_{\text{prev}}^{(r)} w_{\text{prev}}^{(r)},$$

$$\mathbf{v}_{\text{new}}^{(r)} \leftarrow \mathbf{v}_{\text{prev}}^{(r)} + \gamma(\mathbf{H}^{(r)} - \mathbf{U}_{\text{prev}}\mathbf{W}_{\text{prev}}^{(r)}\mathbf{V}_{\text{prev}}^{(r)\,T})^T \mathbf{u}_{\text{prev}} w_{\text{prev}}^{(r)},$$

where $r \in \{s, e\}$. Then, we apply the element-wise max operation with zeros to the updated components to preserve nonnegative constraints, and then normalize them. With these updated components, the seasonal component is updated as: $w_{\text{new}}^{(r)} \leftarrow w_{\text{prev}}^{(r)} \cdot \|\mathbf{u}_{\text{new}}\| \cdot \|\mathbf{v}_{\text{new}}^{(r)}\|, r \in \{s, e\}$. Based on the above formulas, we can keep track of better components by updating them from previous ones. However, the problem is that, at time point $t$, the relationship between $\mathbf{U}(t-1)$, $\mathbf{V}(t-1)$, and $\mathbf{W}(t-p)$ is not reliable anymore because the seasonal component is outdated unlike the other two components, which are frequently updated during the last $p-2$ period. So, we avoid using the previous $\mathbf{W}(t-p)$ directly. EntMine first estimates the temporal $\mathbf{W}_{\text{tmp}}$ to fix the outdated seasonal patterns. Subsequently, it updates the previous $\mathbf{U}$ and $\mathbf{V}$ with the pre-updated $\mathbf{W}_{\text{tmp}}$ Then it updates $\mathbf{W}_{\text{tmp}}$ again with the updated $\mathbf{U}$ and $\mathbf{V}$. These alternating updates enable our EntMine to adapt to recent patterns effectively.

*Theoretical Analysis.* We now present the time and memory complexities of EntMine. The key advantage is that both of the complexities are constant with regard to a given tensor length, even if it evolves without bound. We let $z(\cdot)$ denote the number of non-zero elements of a given matrix/tensor.

Lemma 1. *The time complexity of EntMine is $O(hz(\mathcal{A}(t)) + k(z(\mathbf{H}^{(s)}(t)) + z(\mathbf{H}^{(e)}(t))) + k^2(n + h))$ per time point.*

Proof. Omitted for brevity. □

Lemma 2. *The memory complexity of EntMine is $O(k(n + 2h) + k^2 + z(\mathcal{A}(t)))$ per time point.*

Proof. Omitted for brevity. □

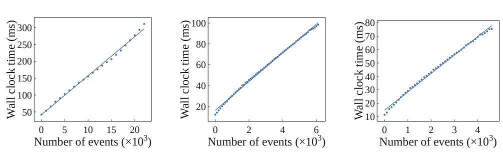
Consequently, the time and memory complexities of EntMine are independent of the tensor stream length, enabling efficient streaming anomaly detection while capturing significant patterns based on entropies.

## 4 EXPERIMENTS

In this section, we compare the performance of the proposed EntMine method with the state-of-the-art streaming anomaly detection methods. We design experiments to answer the following questions: Q1. Scalability: Does our method scale linearly? Q2. Anomaly detection accuracy: How accurate is our method? and Q3. Real-world discoveries: How does it detect abnormal drifting patterns? We implemented our algorithm in Python (ver. 3.13.0) and conducted all experiments on an Intel Xeon Platinum 8268 2.9GHz 24-core CPU with 512GB of memory, running Linux.

*Real-world datasets.* We collected three public transportation datasets spanning three years, from January 1, 2022: **NYC-SC**[1] (NYC CitiBike ride trips between 1328 stations), **WDC-SC**[2] (Capital Bikeshare ride trips between 380 stations), and **TEB-SC**[3] (Bay Wheels ride trips between 285 stations), from three different regions of the United States. For each dataset, we selected a subset of stations (i.e., nodes) that were used as sources and destinations on at least 90% of the days during the observation period. Then, we sum up edge counts on an hourly basis and assume weekly periodicity, i.e., $p = 168$ over the streams, resulting in the sparsity (i.e., the ratio of nonzero elements to the tensor size) of the three datasets being 99% after preprocessing.

**Q1. Scalability.** To evaluate the scalability of our methods, we measure the execution time for each time point in Figure 1. For each dataset, the time complexity scales linearly with the number of events as mentioned in Lemma 1. The result suggests that EntMine can be applied to large real-world datasets because it is scalable enough to decompose an input matrix within millisecond scales, assuming that the data arrive every hour.




(a) NYC-SC (b) WDC-SC (c) TEB-SC

**Figure 1: EntMine scales linearly.**

**Q2. Accuracy.** Here, we evaluate the accuracy of EntMine with the state-of-the-art methods for streaming anomaly detection. Due to the lack of ground-truth labels for anomalies in real-world datasets, we make synthetic data that can simulate significant distribution shifts over time, such as the difference between holiday and normal seasons. The data creation procedure is as follows. (1) We randomly select 15 cut points of a data stream from the beginning points of seasons with length $p = 100$ to obtain ground-truth labels. (2) To simulate node-specific preferences, we randomly draw $p$ multinomial distributions from the Dirichlet distribution with the parameter $\alpha_i = 1, \forall i \in \{1, \ldots, n\}$ for each node. (3) To generate $\mathbf{A}(t)$ by using the set of multinomial distributions, we repeat drawing a category, which corresponds to a destination, until the total count reaches a specified limit for each node and each time point. (4) We iterate steps 2 and 3 to simulate segment-specific patterns. (5) Finally, we randomly add 1 to 5% of elements in the entire data stream as noise. The synthetic data is thus specified by the stream length (i.e., the density of the anomalies) and the number of events per time point. We set $n = 50$ and created 3 sets of synthetic data, A, B, and C, with parameter sets $\{(5000, 20), (10000, 20), (5000, 40)\}$, respectively.

**Table 1: EntMine spots abnormal pattern shifts accurately: it consistently outperforms its competitors in terms of the AUC-PR averaged from ten runs (higher is better). Bold scores indicate the best performances.**

| | | SYN-A | SYN-B | SYN-C |
|---|---|---|---|---|
| **Entropy-based** | Average entropy | 0.32 ± 0.07 | 0.14 ± 0.03 | 0.40 ± 0.09 |
| **Count-based** | Average degree | 0.33 ± 0.08 | 0.13 ± 0.02 | 0.33 ± 0.09 |
| | MIDAS | 0.33 ± 0.04 | 0.18 ± 0.06 | 0.74 ± 0.06 |
| | SSMF | 0.82 ± 0.05 | 0.76 ± 0.07 | 0.83 ± 0.07 |
| **Proposed** | EntMine-F | 0.40 ± 0.08 | 0.16 ± 0.04 | 0.40 ± 0.08 |
| | EntMine-S | 0.95 ± 0.03 | 0.86 ± 0.07 | 0.97 ± 0.01 |
| | EntMine | **0.96 ± 0.02** | **0.95 ± 0.03** | **0.99 ± 0.00** |

*Baselines.* As the simplest baselines, we incrementally computed the average degree/entropy, and then reported the squared error between the local degree/entropy calculated at each time point and the average as an anomaly score. MIDAS [5] is a micro-cluster-based anomaly detection method for edge streams. We chose it because its latest extension was proposed to incorporate rich edge attributes, and it is equivalent to MIDAS without them. Following the original paper, we set two hash functions for the CMS data structures, and set the number of CMS buckets to 2719. SSMF [13] is an online tensor factorization approach that can detect regime shifts by changing seasonal components. We also compare EntMine with its variants to evaluate how well each component works. EntMine-F is a special case of EntMine when the number of lookback steps $h = 1$, which means it only considers local entropies computed at each time point, while the original EntMine uses $h = 9$. EntMine-S updates each component in $\mathcal{M}(t)$ once at each time point to validate the effectiveness of our pre-update strategy for seasonal components. For these tensor factorization approaches, we set the learning rate to 0.3 and the number of components 30 initialized by non-negative CP decomposition [15]

*Results.* Table 1 summarizes the results of the anomaly detection, where each method processed each synthetic data stream 10 times to report node anomaly scores. The performance metric is the area under the precision-recall curve (AUC-PR). As expected, EntMine outperforms its baselines for all datasets, and specifically, improved up to 25.4% in terms of the average AUC-PR. In the ablation study, EntMine-F failed to spot the correct cut points because local entropies can fluctuate over time. Since EntMine-S requires more graphs than EntMine to update itself sufficiently, it reported false positives after the ground truth cut points, resulting in the second best in the comparison. In conclusion, our method accurately detects anomalies based on our proposed coupled tensor factorization approach and efficient update strategy.

[1]https://citibikenyc.com/system-data
[2]https://capitalbikeshare.com/system-data
[3]https://www.lyft.com/bikes/bay-wheels/system-data

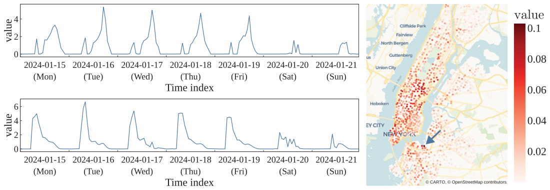



**Figure 2: EntMine uncovers interpretable components: one set of the components of NYC-SC: (left) seasonal components of departure and destination views in the top and bottom, respectively, and (right): a heatmap of the node component.**

**Q3. Real-world Discoveries.** In this section, we demonstrate the effectiveness of EntMine on real-world datasets. Figure 2 shows one of the decomposed components of NYC-SC. We can see a series of seasonal components for the departure exhibit weekday patterns including peaks every morning and every evening. In contrast, one for the destination shows strong peaks every morning. This component suggests a commuting pattern: in the mornings, bicycles arrive from various locations and are returned at stations related to this component, while in the evenings, bicycles are rented from this location and returned elsewhere. The heatmap based on the node component values shows increased values at stations near ferry terminals and train stations. This implies that many users utilize shared bicycles in the morning to reach transportation hubs on their way to work, and in the evening to return home, which aligns with common commuting behavior. By jointly decomposing coupled entropy matrices, EntMine successfully captures the shared structure in such station components and two distinct temporal dynamics of their usage.

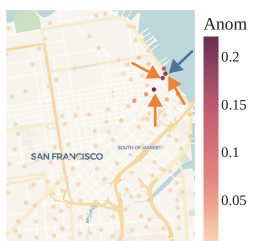



**Figure 3: Visualization of anomaly scores for TEB-SC at 2024-07-28 4 a.m.**

Figure 3 shows the heatmap of node anomaly scores for the TEB-SC dataset on July 28th, 2024, at 4 a.m.– the time when the largest anomaly score was observed. Notably, the San Francisco Marathon was held on this day, starting at 5:15 a.m., with the start location (Embarcadero at Market, indicated by blue arrow), located near the top three most anomalous stations (indicated by orange arrows). This suggests an unusual distribution of activity likely caused by marathon participants or their supporters. For comparison, we also analyzed the same dataset by MIDAS, but it ranked this time point as only the 160141st most anomalous. This contrast highlights the effectiveness of our entropy-based method in detecting meaningful anomalies that are overlooked by existing count-based approaches.

## 5 CONCLUSION AND FUTURE WORK

In this paper, we present a new streaming tensor factorization approach, namely EntMine, for information-theoretic pattern discovery in large evolving graphs. Our method has the following properties: it is *scalable:* our method scales linearly with the number of events at any time to maintain important patterns and infer node anomaly scores; it is *accurate:* our method outperforms the state-of-the-art methods by accurately detecting changes of event distribution; and it is *effective:* it reveals meaningful seasonal and node-based patterns hidden in Experiments using three real-world shared cycle datasets demonstrate that our proposed method can efficiently factorize evolving graphs at any time, and achieved the highest AUC-PR score, and discover interpretable seasonal patterns, while existing count-based methods fail to find out them.

We are considering the following future works. **Other domains.** We used only three shared-cycle datasets for real-world discoveries. Although we can gain interesting insights from them, we don't think our method can be applied solely to shared-cycle datasets. We're considering using EntMine to incorporate more diverse datasets and gain deeper insights from them. **Classification.** We're trying to classify the nodes and reveal their hidden similarities to efficiently grasp their properties and make them more interpretable.

## ACKNOWLEDGMENTS

This work was supported by "Program for Leading Graduate Schools" of the Osaka University, Japan, JSPS KAKENHI Grant-in-Aid for Scientific Research Number JP20910053, JST CREST JPMJCR23M3, JST COI-NEXT JPMJPF2009, JPMJPF2115 .

## REFERENCES

[1] Evrim Acar, Tamara G Kolda, and Daniel M Dunlavy. 2011. All-at-once optimization for coupled matrix and tensor factorizations. *arXiv preprint arXiv:1105.3422* (2011).

[2] Sarahi Aguayo-Tapia, Gerardo Avalos-Almazan, and Jose de Jesus Rangel-Magdaleno. 2024. Entropy-based methods for motor fault detection: a review. *Entropy* 26, 4 (2024), 299.

[3] Gianmarco Baldini. 2020. On the application of entropy measures with sliding window for intrusion detection in automotive in-vehicle networks. *Entropy* 22, 9 (Sept. 2020), 1044.

[4] Caleb Belth, Xinyi Zheng, and Danai Koutra. 2020. Mining persistent activity in continually evolving networks. In *KDD*. ACM, New York, NY, USA.

[5] Siddharth Bhatia, Bryan Hooi, Minji Yoon, Kijung Shin, and Christos Faloutsos. 2020. Midas: Microcluster-Based Detector of Anomalies in Edge Streams. *AAAI* 34, 04 (April 2020), 3242–3249.

[6] Siddharth Bhatia, Arjit Jain, Pan Li, Ritesh Kumar, and Bryan Hooi. 2021. Mstream: Fast anomaly detection in multi-aspect streams. In *WWW'21*. 3371–3382.

[7] Lei Cai, Zhengzhang Chen, Chen Luo, Jiaping Gui, Jingchao Ni, Ding Li, and Haifeng Chen. 2021. Structural temporal graph neural networks for anomaly detection in dynamic graphs. In *CIKM'21*. 3747–3756.

[8] Xinyu Chen, Mengying Lei, Nicolas Saunier, and Lijun Sun. 2021. Low-rank autoregressive tensor completion for spatiotemporal traffic data imputation. *IEEE Transactions on Intelligent Transportation Systems* 23, 8 (2021), 12301–12310.

[9] Dongjin Choi, Jun-Gi Jang, and U Kang. 2019. S3CMTF: Fast, accurate, and scalable method for incomplete coupled matrix-tensor factorization. *PLoS One* 14, 6 (June 2019), e0217316.

[10] T.M. Cover and J.A. Thomas. 2005. *Entropy, Relative Entropy, and Mutual Information.* John Wiley & Sons, Ltd, Chapter 2, 13–55.

[11] Shikai Fang, Xin Yu, Shibo Li, Zheng Wang, Mike Kirby, and Shandian Zhe. 2023. Streaming factor trajectory learning for temporal tensor decomposition. *Advances in Neural Information Processing Systems* 36 (2023), 56849–56870.

[12] Pierdomenico Fiadino, Alessandro D'Alconzo, Mirko Schiavone, and Pedro Casas. 2015. Challenging entropy-based anomaly detection and diagnosis in cellular networks. *Comput. Commun. Rev.* 45, 4 (Sept. 2015), 87–88.

[13] Koki Kawabata, Siddharth Bhatia, Rui Liu, Mohit Wadhwa, and Bryan Hooi. 2021. SSMF: shifting seasonal matrix factorization. *Advances in Neural Information Processing Systems* 34 (2021), 3863–3873.

[14] Claude E. Shannon. 1948. A mathematical theory of communication. *Bell System Technical Journal* 27, 3 (1948), 379–423.

[15] Amnon Shashua and Tamir Hazan. 2005. Non-negative tensor factorization with applications to statistics and computer vision. In *ICML* (Bonn, Germany). Association for Computing Machinery, New York, NY, USA, 792–799.

[16] Kijung Shin, Bryan Hooi, Jisu Kim, and Christos Faloutsos. 2017. Densealert: Incremental dense-subtensor detection in tensor streams. In *KDD*. 1057–1066.

[17] Qingquan Song, Xiao Huang, Hancheng Ge, James Caverlee, and Xia Hu. 2017. Multi-aspect streaming tensor completion. In *KDD*. 435–443.

[18] Hsiang-Fu Yu, Nikhil Rao, and Inderjit S Dhillon. 2016. Temporal regularized matrix factorization for high-dimensional time series prediction. *Advances in neural information processing systems* 29 (2016).