Anomaly Detection in Dynamic Networks using Multi-view Time-Series Hypersphere Learning

Xian Teng
University of Pittsburgh
135 North Bellefield Avenue
Pittsburgh, Pennsylvania 15260
xian.teng@pitt.edu

Yu-Ru Lin
University of Pittsburgh
135 North Bellefield Avenue
Pittsburgh, Pennsylvania 15260
yurulin@pitt.edu

Xidao Wen
University of Pittsburgh
135 North Bellefield Avenue
Pittsburgh, Pennsylvania 15260
xidao.wen@pitt.edu

ABSTRACT

Detecting anomalous patterns from dynamic and multi-attributed network systems has been a challenging problem due to the complication of temporal dynamics and the variations reflected in multiple data sources. We propose a Multi-view Time-Series Hypersphere Learning (MTHL) approach that leverages multi-view learning and support vector description to tackle this problem. Given a dynamic network with time-varying edge and node properties, MTHL projects multi-view time-series data into a shared latent subspace, and then learns a compact hypersphere surrounding normal samples with soft constraints. The learned hypersphere allows for effectively distinguishing normal and abnormal cases. We further propose an efficient, two-stage alternating optimization algorithm as a solution to the MTHL. Extensive experiments are conducted on both synthetic and real datasets. Results demonstrate that our method outperforms the state-of-the-art baseline methods in detecting three types of events that involve (i) time-varying features alone, (ii) time-aggregated features alone, as well as (iii) both features. Moreover, our approach exhibits consistent and good performance in face of issues including noises, anomaly pollution in training phase and data imbalance.

KEYWORDS

Anomaly detection, Dynamic networks, Multi-view learning

ACM Reference format:
Xian Teng, Yu-Ru Lin, and Xidao Wen. 2017. Anomaly Detection in Dynamic Networks using Multi-view Time-Series Hypersphere Learning. In *Proceedings of ACM Conference, Washington,DC, USA, July 2017 (Conference'17),* 11 pages.
https://doi.org/10.475/123_4

1 INTRODUCTION

The problem of anomaly detection in dynamic networks has attracted much attention in a broad range of domains, such as transportation, communication, financial systems, and social networks. Examples include detection of civil unrest using social media data [6, 25], identification of crowd activities or emergencies in cities [5, 27, 33] and discovery of network intrusion or network failures [8, 36]. Particularly with the increasing adoption of ubiquitous sensors and social mobile technologies, it becomes possible to continuously collect datasets from multiple data sources (so-called "multi-view" datasets) in real time. The continuously-gathered data allows us to understand the temporal regularities and irregularities of a dynamic system. Furthermore, data collected from multiple data sources offer complementary information about the same objects from various perspectives, which promise the potential of more effective anomaly detection than those only based on single-view data.

Over the past decade, a variety of anomaly detection methods in dynamic networks have been put forward [6, 15, 18, 25, 27, 32, 33, 35, 36]. These methods complement traditional anomaly detectors, e.g., Support Vector Machines (SVM) [34] and the Local Outlier Factor (LOF) [4], as the dynamic nature and network structure have introduced new types of anomalies and challenges. For example, Non-Parametric Heterogeneous Graph Scan (NPHGS) [6] and EventTree+ [33] find anomalous subgraphs with structural constraint as a way to detect traffic accidents or abnormal crowd activities. In spite of their success under some situations, these approaches mainly focus on static or time-aggregated features and lack the ability of mining time-sensitive anomalous patterns. For example, EventTree+ directly uses the aggregated activity level as an attribute for each node, without consideration of the daily variation of activities. While converting time-varying attributes into aggregated features is convenient, the process tends to lose important information in detecting certain anomalies, e.g., anomalies with temporal irregularities whereas their time-aggregated attributes may seem normal.

In addition to single-view approaches, there have been works dealing with multi-view datasets, including Horizontal Anomaly Detection (HOAD) [12], Multi-view Low Rank Analysis (MLRA) [23], Outliers Ranking (OutRank) based on subspace analysis [28] and anomaly detection by Affinity Propagation (AP) [11, 26]. Most of these methods focus on inconsistent or different behaviors across different sources, which is referred to as "horizontal anomaly detection" [12]. However, methods that exploit multi-view data as complementary information [23] for extracting normal and abnormal patterns are less explored. In this paper, we consider that abnormal events would create a disturbance of regularities across various views, and by mining such consistent irregular patterns in multiple data sources, we can achieve a more reliable detection result than those based on a single view.

Here we propose a novel anomaly detection framework named "Multi-View Time Series Hypersphere Learning" (MTHL) in dynamic networks. Figure 1 illustrates the key idea of our proposed

Conference'17, July 2017, Washington,DC, USA

ACM ISBN 123-4567-24-567/08/06…$15.00
https://doi.org/10.475/123_4

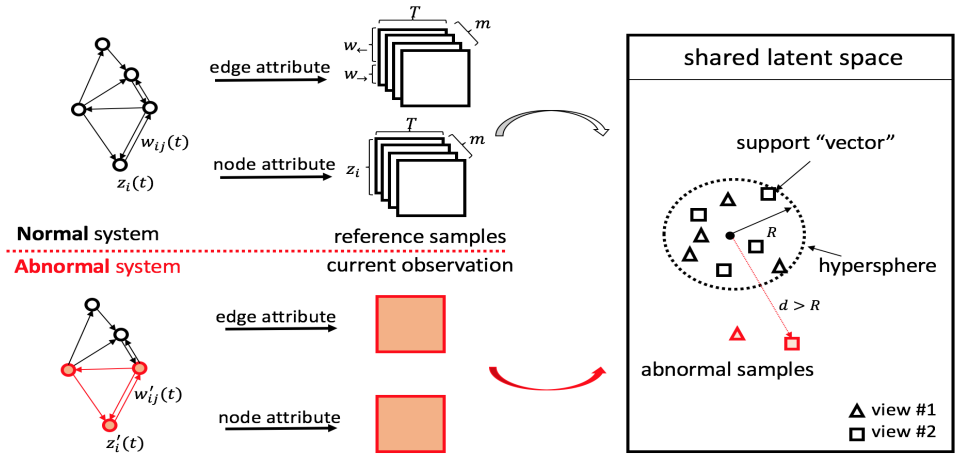

Figure 1: Illustration of our MTHL approach. In terms of the network system, node and edge attributes are taken as two distinct views. The multi-view temporal information is represented as two (or multiple) sets of matrices and then projected into a shared latent space. MTHL mines the normal pattern (soft boundary between normal and abnormal cases) by learning a compact hypersphere surrounding reference samples, and detects outliers based on its distance to the hypersphere centroid.

approach. First, to preserve the temporal variation of multiple attributes, we use multivariate time series representation – a chronologically ordered sequence of feature vectors that capture variation in attribute values. Second, we assume normal samples collected from multiple views would be close to one another in a low-dimensional latent space. To obtain a good representation of normal pattern, we leverage Support Vector Data Description (SVDD) [37] to extract normal patterns. Specifically, MTHL learns a hypersphere around the reference set and distinguishes normal and abnormal samples according to their distances to the hypersphere center. Our contributions can be summarized as follows:

(1) We propose a novel approach called MTHL for anomaly detection in dynamic networks. By full exploitation of multi-view time-series data, MTHL is able to detect events that involve irregular temporal variations, which are easily neglected by traditional approaches that only depend on aggregated features.
(2) By leveraging multi-view learning and support vector description, our approach learns a hypersphere that facilitates the effective identification of anomalies.
(3) We propose an efficient algorithm which involves two alternating optimization stages, by using gradient descent and Lagrangian duality theory. In our runtime comparison, MTHL exhibits the best time performance at the testing phase.
(4) We conduct extensive experiments on both synthetic and real-world datasets. Results demonstrate that our method consistently outperforms the state-of-the-art baseline methods in face of data imbalance as well as noises and anomaly pollution during the training phase.

The rest of the paper is organized as follows. In section 2, we briefly review the related work. In section 3, we present problem definition and notations. Section 4 and 5 describe the proposed MTHL approach and its algorithmic solution, respectively. Experimental evaluation is reported in Section 6, with conclusion in section 7.

2 RELATED WORK

Anomaly Detection in Dynamic Networks. Beyond traditional anomaly detection, there has been an increasing interest in anomaly detection in dynamic networks, particularly due to its ability to describe objects and relationships with time-varying properties [2, 14, 32].

In the realm of dynamic networks, what forms up an anomalous object heavily relies on the applications. Detection tasks can span from detecting abnormal vertices [15, 16, 18] and edges [1, 17, 24], to identifying anomalous subgraphs [6, 7, 27, 29, 30, 36] and events [20, 31]. Ji *et al.* [18] detect local evolutionary outliers (vertices) by investigating the shifts in community involvement. Li *et al.* [24] identify abnormal edges in vehicle traffic networks by studying edge weight evolution. In terms of anomalous subgraphs, Chen *et al.* [7] focus on community behaviors and propose to detect six types of community-based anomalies: grown, shrunken, merged, split, born, and vanished communities. Mongiovi *et al.* [27] design a method, called "NetSpot", to find the significant anomalous regions (i.e. a set of adjacent, connected links) and time intervals. A series of scan statistics based approaches [6, 29, 30] are also developed to detect anomalous clusters though subset searching in the spatio-temporal domain.

Many prior works deal with dynamic networks by partitioning the stream data into discrete time windows and then construct aggregated features as the "so-called" temporal properties. Instead, we approach the problem by conducting a fine investigation regarding how the system evolves within each time window. Another difference lies in that our approach is developed from the perspective of multi-view learning, so that it can make use of the mutual-support data sources to achieve better results.

Multi-view Learning. The existence of multiple data sources has inspired a lot of works conducted in the multi-view setting, such as multi-view clustering [3, 11, 21], subspace learning [13, 38, 39], multi-view classification [19, 22] and multi-view outliers detection [12, 23, 26, 28]. The most relevant work to our paper is multi-view outlier detection.

Gao *et al.* [12] were the first to study horizontal anomalies by exploring the inconsistent behaviors across different views. In their work, they proposed a clustered-based approach, called Horizontal Anomaly Detection (HOAD). In specific, HOAD performs clustering simultaneously with all views, and marks those objects belonging to different clusters as outliers. Alvarez *et al.* [26] approached the similar problem by an affinity propagation (AP) based method. However, both HOAD and AP are designed for one type of outliers. As Li *et al.* [23] have pointed out, there are two types of anomalies under the multi-view setting: Type I outlier is the so-called "horizontal outlier" proposed in HOAD paper [12]; Type II outliers refer to the ones that display anomalous patterns in each single view. They develop a Multi-View Low-Rank Analysis (MLRA) approach to simultaneously detect both types of anomalies. Despite that, Type II outlier detection still needs further exploration, so we restrict the scope of this paper to the second category. Furthermore, none of these introduced works are conducted in the realm of time-varying network systems. Therefore, we will make a contribution from this direction.

3 PROBLEM DEFINITION

In this section, we introduce definitions, notations and problem formulation. Table 1 lists the notations used in this paper.

Definition 3.1. **Dynamic Network.** A dynamic network is defined as a directed network $\mathcal{G}(t) = \{\mathcal{V}, \mathcal{E}, z(t), w(t)\}$, where $\mathcal{V}$ denotes the set of vertices, and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ refers to the set of directed edges, z is a vertex mapping function: $\mathcal{V} \rightarrow \mathbb{R}^{d_z}$ that maps vertex i to its d_z-dimensional feature vector $z_i(t)$ at each time step t, w is an edge mapping function: $\mathcal{E} \rightarrow \mathbb{R}$ which associates each edge e_{ij} (from i to j) with a edge-specific value $w_{ij}(t)$ at each time step t.

Such representation of dynamic networks can be used to describe a variety of systems in the real world, such as an urban area consisting of small regions, a financial system connecting banks, and a social network composed of users and institutions. In those networked systems, we are interested in detecting which units (e.g., regions, banks and users) are anomalous compared to their regular norms. Therefore, for each vertex i, we transform edge attributes into a second view of "vertex attribute" by considering all edges connected with it, i.e. $w_i = \{w_{\leftarrow}; w_{\rightarrow}\}^T$ where $w_{\leftarrow}$ and $w_{\rightarrow}$ are features for incoming and outgoing edges. While we focus on two views capturing node and edge attributes in this work, it can be generalized into a more generic framework with more views involved.

Definition 3.2. **Multi-view Multivariate Time-Series.** A multi-view multivariate time series can be denoted as $\{X^v, v = 1, ..., V\}$, where V denotes the number of views, and $X^v \in \mathbb{R}^{d_v \times T}$ represents the time series collected from v-th view. Here d_v is the number of attributes and T is the number of time steps in a time window. Each row of X^v records the temporal variation for each attribute, and each column represents the observation of all attributes at each time step.

Table 1: Notation Definition.

Notation	Definition
$\mathcal{G}(t)$	Attributed dynamic network
$\mathcal{X}$	Reference data set
X^v	Multivariate time-series for view v
d_v, T	Feature and temporal dimensions of X^v
V	Number of views
m	Number of reference samples
P^v, Q^v	Bilinear projection matrices
Y_i^v	Projection in latent space
(Y^*, R)	Hypersphere (center and radius)
L_C	Temporal Laplacian matrix
$f = \Theta + \Phi + \Psi$	Objective function
p, q	Reduced feature and temporal dimensions
ξ	Slack variable
λ_1, λ_2	Penalty and trade-off parameters in f
τ_i^v	Weighting parameter in Θ
$d_{ij}^v, \bar{d_i^v}$	Pairwise and average distance
α, β	Lagrangian multipliers
$\mathcal{L}$	Optimization goal derived by Lagrangian duality
ϕ	Kernel function
σ	Gaussian noise parameter
ν	Anomaly pollution parameter
ω	Data imbalance parameter

In our case of dynamic networks, the view X^1 corresponds to the node attribute $\{z(1), ..., z(T)\} \in \mathbb{R}^{d_z \times T}$, and the view X^2 corresponds to $\{w(1), ..., w(T)\} \in \mathbb{R}^{d_w \times T}$. The multi-view multivariate time-series data captures temporal information for all measurements during each time window T, allowing for discovering anomalies that involve temporal irregularities. We formally formulate our problem as follows.

Definition 3.3. **Dynamic Multi-view Anomaly Detection.** For each vertex in a dynamic network, let $\mathcal{X} = \{X_i^v | i = 1, ..., m, v = 1, ..., V\}$ denote a set of historical observations, also called "reference set". Here m is the number of samples in each view, so that there are mV elements in $\mathcal{X}$. Given a new observation $\{X^v | v = 1, ..., V\}$ for this vertex, our goal is to determine whether it is normal or abnormal in comparison with the reference set $\mathcal{X}$.

4 MULTI-VIEW TIME-SERIES HYPERSPHERE LEARNING (MTHL)

4.1 Motivation

To approach the above defined problem, we first need to learn a good representation of normal patterns from reference data set; based on the learned representation, we can identify anomalous cases or measure the strength of anomalousness. Here we note that several critical points should be carefully considered:

(1) How to extract the intrinsic patterns for both feature and temporal information from high dimensional time-series data;
(2) How to integrate time-series samples from different views to promote anomaly detection;
(3) How to discriminate normal and anomalous cases according to the reference set.

To deal with the first challenge, we leverage a bilinear dimensionality reduction approach [22]. Dimensionality reduction has been widely used in data mining to extract important properties by filtering out redundancy and noise. Here we seek to preserve both feature and temporal structures that are useful for anomaly detection. Therefore, we learn a pair of bilinear projections to reduce feature and temporal dimensionality, respectively. The second problem requires mitigating the gap between different views and coordinating information across views. For this purpose, we employ a strategy in the multi-view learning field [9, 13, 22]: assuming that multi-view data share the same low-dimensional latent subspace. Finally, for the third question, we leverage support vector data description (SVDD) [37] with latent space projection to distinguish abnormal observations from normal ones.

4.2 Objective function

We introduce the objective function, which contains three components: reconstruction error from bilinear projection, hypersphere learning and temporal smoothing regularization.

Reconstruction error. Given a reference set of time-series samples $\mathcal{X} = \{X_i^v | i = 1, ..., m, v = 1, ..., V\}$, for each view v we seek to learn a pair of bilinear projections, $P^v \in \mathbb{R}^{d_v \times p}$ and $Q^v \in \mathbb{R}^{d_v \times q}$, to reduce both feature and time dimensionality, where p and q are the reduced dimensions. For an arbitrary sample X_i^v in view v, we map it into $Y_i^v = P^{v\,T} X_i^v Q^v$, where $Y_i \in \mathbb{R}^{p \times q}$ is the corresponding low-dimensional representation. To force normal samples to be as close as possible, we impose a strict constraint: all reference samples share the same low-dimensional representation Y^*. To minimize average reconstruction errors, we have the first part in the loss function:

$$\Theta(P^v, Q^v, Y^*) = \tau \sum_v \sum_i \left\| P^{v\,T} X_i^v Q^v - Y^* \right\|_F^2, \tag{1}$$

with the constraints:

$$P^{v\,T} P^v = \mathbf{I}_p, Q^{v\,T} Q^v = \mathbf{I}_q, v = 1, ..., V. \tag{2}$$

where τ is a normalization parameter equal to $1/mV$, and the bilinear projections are semi-orthogonal matrices.

Hypersphere learning. Analogous to SVDD [37], after projecting data into a latent subspace, we try to obtain a compact hypersphere (Y^*, R) via minimizing the radius R:

$$\Phi(P^v, Q^v, Y^*, R) = R^2, \tag{3}$$

with the constraints:

$$\left\| P^{v\,T} X_i^v Q^v - Y^* \right\|_F^2 \le R^2, i = 1, ..., m. \tag{4}$$

To deal with the cases where the given reference set includes a small fraction of anomalies, we revise Φ as follows:

$$\Phi(P^v, Q^v, Y^*, R, \xi) = R^2 + \lambda_1 \sum_v \sum_i \xi_i^v, \tag{5}$$

with the constraints:

$$\left\| P^{v\,T} X_i^v Q^v - Y^* \right\|_F^2 \le R^2 + \xi_i^v, i = 1, ..., m, v = 1, ..., V, \tag{6}$$

$$\xi_i^v \ge 0, i = 1, ..., m, v = 1, ..., V. \tag{7}$$

where the additional slack variables $\xi_i^v \ge 0$ are added to account for data outside the boundary, and the positive parameter λ_1 is to penalize large distance.

Temporal smoothing regularization. In practice, many systems that can be described by dynamic networks, tend to slightly change over time. In fact, temporal fluctuations in many situations can be considered as an indicator of anomalies. To ensure the local smoothness, we incorporate a temporal smoothing regularization. For $P^{v\,T} X^v$, the t-th column $P^{v\,T} X^v(\cdot, t)$ represents the feature vector at time step t, and the discrepancy between consecutive time steps is minimized.

$$\begin{aligned}
\Psi(P^v) &= \frac{1}{2} \sum_{t', t''} C_{t't''} \left\| P^{v\,T} X^v(\cdot, t') - P^{v\,T} X^v(\cdot, t'') \right\|_F^2 \\
&= \sum_{t'} P^{v\,T} X^v(\cdot, t') D_{t't'} X^v(\cdot, t')^T P^v \\
&\quad - \sum_{t', t''} P^{v\,T} X^v(\cdot, t') C_{t't''} X^v(\cdot, t'')^T P^v \\
&= \mathrm{Tr}\left(P^{v\,T} X^v (D - C) X^{v\,T} P^v \right) \\
&= \mathrm{Tr}\left(P^{v\,T} X^v L_C X^{v\,T} P^v \right),
\end{aligned} \tag{8}$$

where C is a predefined matrix with each entry $C_{t't''}$ indicating how much weight is given to penalize the discrepancy between the t'-th and t''-th columns, D is a diagonal matrix with entries $D_{t't'} = \sum_{t''} C_{t't''}$, L_C is the Laplacian matrix associated with C, and $\mathrm{Tr}(\cdot)$ means the trace of a matrix. Here we define the prior weighting matrix C in a simple way:

$$C_{t't''} = \begin{cases} 1, & |t' - t''| \le s, \\ 0, & \text{otherwise.} \end{cases}$$

In this case, the successive columns in $P^v X^v$ within s steps are forced to be similar. In this paper, s is empirically chosen to be 2. Other more sophisticatedly designed weight matrices can also be employed.

MTHL Objective function. By putting Eq. (1), (5), (8) together, we have the following MTHL objective function:

$$\begin{aligned}\min_{\mathcal{P}} f(\mathcal{P}) &= \min_{\mathcal{P}} \{\Theta + \Phi + \Psi\} \\ &= \min_{\mathcal{P}} \Bigg\{ \tau \sum_{v} \sum_{i} \left\| P^{v\,T} X_i^v Q^v - Y^* \right\|_F^2 \\ &+ R^2 + \lambda_1 \sum_{v} \sum_{i} \xi_i^v \\ &+ \lambda_2 \sum_{v} \sum_{i} \mathrm{Tr}\left(P^{v\,T} X_i^v L_p X_i^{v\,T} P^v \right) \Bigg\},\end{aligned} \tag{9}$$

subject to Eq. (2),(6) and (7). The set $\mathcal{P} = \{P^v, Q^v, Y^*, R, \xi\}$ is our optimization goal. In summary, the first term is normalized reconstruction error which encourages samples close to the centroid Y^*, the second term minimizes the volume of hypersphere (Y^*, R), and the third term is a temporal smoothing regularization to prevent dramatic fluctuations. λ_2 is the trade-off parameter that balances the influence of the third term.

4.3 Weighted reconstruction error.

In Eq. (9), the reconstruction error uses a uniform normalization parameter τ, without considering the difference among the time-series in the reference set. However, it is possible that the reference set might contain a small number of anomalous instances, which we refer as "anomaly pollution". To augment the robustness towards anomaly pollution, we extend the objective function in Eq. (9) by rewrite Θ as a weighted reconstruction error:

$$\begin{aligned}\min_{\mathcal{P}} f(\mathcal{P}) = \min_{\mathcal{P}} \Bigg\{ & \sum_{v} \sum_{i} \tau_i^v \left\| P^{v\,T} X_i^v Q^v - Y^* \right\|_F^2 \\ & + R^2 + \lambda_1 \sum_{v} \sum_{i} \xi_i^v \\ & + \lambda_2 \sum_{v} \sum_{i} \mathrm{Tr}\left(P^{v\,T} X_i^v L_p X_i^{v\,T} P^v \right) \Bigg\},\end{aligned} \tag{10}$$

where $\sum_v \sum_i \tau_i^v = 1$.

We assign such τ_i^v by exploiting pairwise distance between samples in reference data set. We assume that the true normal samples tend to locate closely to each other, whereas the anomalous ones tend to be far away from normal clusters. Therefore, we calculate the pairwise distance (dissimilarity) for each view v via:

$$d_{ij}^v = \left\| X_i^v - X_j^v \right\|_F^2. \tag{11}$$

Given an arbitrary vertex i, we can obtain its average distance to all other samples:

$$\bar{d_i^v} = \frac{1}{m-1} \sum_{j \neq i} \left\| X_i^v - X_j^v \right\|_F^2. \tag{12}$$

Based on $\bar{d_i^v}$, we further define the weighting parameter τ_i^v by the following exponential function:

$$\tau_i^v = \eta e^{-\bar{d_i^v}}, \tag{13}$$

where η is a normalization parameter.

5 SOLUTION: TWO-STAGE ALTERNATIVE OPTIMIZATION ALGORITHM

Eq. (10) is not a jointly convex optimization problem for all variables $\mathcal{P}$, but if the bilinear projections $\{P^v, Q^v\}$ are fixed, it will become a traditional convex optimization in terms of $\{Y^*, R, \xi\}$. Therefore, we determine to divide the problem into two alternating stages. In stage I, we use gradient descent to update $\{P^v, Q^v\}$; In stage II, we keep $\{P^v, Q^v\}$ fixed, and employ Lagrange duality theory to optimize $\{Y^*, R, \xi\}$.

Stage I. Gradient descent. We alternately update P^v and Q^v by following the rules:

$$P^v \leftarrow P^v - \gamma \frac{\partial f}{\partial P^v}, \tag{14}$$

$$Q^v \leftarrow Q^v - \gamma \frac{\partial f}{\partial Q^v}, \tag{15}$$

where γ is the learning rate. The partial derivatives can be represented as:

$$\frac{\partial f}{\partial P^v} = 2 \sum_i \tau_i^v X_i^v Q^v (Q^{v\,T} X_i^{v\,T} P^v - Y^{*\,T}) + 2\lambda_2 \sum_i X_i^v L_p X_i^{v\,T} P^v, \tag{16}$$

$$\frac{\partial f}{\partial Q^v} = 2 \sum_i \tau_i^v X_i^{v\,T} P^v (P^{v\,T} X_i^v Q^v - Y^*). \tag{17}$$

Stage II. Lagrangian duality. Given a fixed pair of bilinear projections $\{P^v, Q^v\}$, we can obtain a set of low-dimensional representations $Y = \{Y_i^v \in \mathbb{R}^{p \times q} | i = 1, ..., m, v = 1, ..., V\}$. In the shared space, we do not distinguish views, hence we remove superscript v and rewrite Y_i^v, τ_i^v, ξ_i^v as $Y = \{Y_i | i = 1, ..., mV\}$, $\{\tau_i | i = 1, ..., mV\}$ and $\{\xi_i | i = 1, ..., mV\}$. The temporal smoothing regularization Ψ only depends on P^v, it will not be included in this stage. In light of *Lagrangian duality theory*, the constraints can be incorporated into Eq. (10) via Lagrangian multipliers:

$$\begin{aligned}\mathcal{L}(R, Y^*, \xi, \alpha, \beta) = & \sum_i \tau_i \left\| Y_i - Y^* \right\|_F^2 + R^2 + \lambda_1 \sum_i \xi_i \\ & - \sum_i \alpha_i \left(R^2 + \xi_i - \left\| Y_i - Y^* \right\|_F^2 \right) - \sum_i \beta_i \xi_i,\end{aligned} \tag{18}$$

where $\alpha_i \geq 0$ and $\beta_i \geq 0$ are Lagrangian multipliers. The dual problem suggests that $\mathcal{L}$ should be minimized with respect to $\{R, Y^*, \xi\}$ and then maximized with respect to $\{\alpha, \beta\}$:

$$\max_{\alpha, \beta} \min_{R, Y^*, \xi} \mathcal{L}(R, Y^*, \xi; \alpha, \beta). \tag{19}$$

Setting partial derivatives to zeros gives the constraints:

$$\frac{\partial \mathcal{L}}{\partial R} = 0 \Rightarrow \sum_i \alpha_i = 1, \tag{20}$$

$$\frac{\partial \mathcal{L}}{\partial Y^*} = 0 \Rightarrow Y^* = \frac{1}{2} \sum_i (\tau_i + \alpha_i) Y_i, \tag{21}$$

$$\frac{\partial \mathcal{L}}{\partial \xi_i} = 0 \Rightarrow \lambda_1 - \alpha_i - \beta_i = 0. \tag{22}$$

Re-substituting Eq. (20), (21), (22) into Eq. (18) can result in:

$$\max_{\alpha} \mathcal{L}(\alpha) = \max_{\alpha} \left\{ \sum_{i} (\tau_i + \alpha_i)\phi(Y_i, Y_i) - \frac{1}{2} \sum_{i,j} (\tau_i + \alpha_i)(\tau_j + \alpha_j)\phi(Y_i, Y_j) \right\}, \tag{23}$$

where $\phi(Y_i, Y_j) = \mathrm{Tr}(Y_i {Y_j}^T)$ is our kernel function. To resolve the quadratic optimization problem in Eq. (23) in terms of α, we apply the Sequential Minimal Optimization (SMO)-type decomposition method proposed by Fan *et al.* [10] that achieves linear convergence.

According to the *Kuhn-Tucker conditions* for optimality, we have the following equations:

$$\alpha_i (R^2 + \xi_i - \left\| Y_i - Y^* \right\|_F^2) = 0, \tag{24}$$

$$\beta_i \xi_i = 0. \tag{25}$$

which are equivalent to:

$$\alpha_i = 0, \beta_i = \lambda_1 \iff \left\| Y_i - Y^* \right\|_F^2 \leq R^2, \xi_i = 0, \tag{26}$$

$$0 < \alpha_i < \lambda_1, 0 < \beta_i < \lambda_1 \iff \left\| Y_i - Y^* \right\|_F^2 = R^2, \xi_i = 0, \tag{27}$$

$$\alpha_i = \lambda_1, \beta_i = 0 \iff \left\| Y_i - Y^* \right\|_F^2 \geq R^2, \xi_i \geq 0. \tag{28}$$

From the above three scenarios we can see that: most data are located inside the hypersphere with $\alpha_i = 0$, they make no contribution. Only those samples with $\alpha_i > 0$ play a role in determining the hypersphere (so-called "support vectors"). With the solution of α, we can calculate Y^* based on Eq. (21), and obtain the radius R by:

$$R^2 = \phi(Y_k, Y_k) - \sum_{i} (\tau_i + \alpha_i)\phi(Y_k, Y_i) + \frac{1}{4} \sum_{ij} (\tau_i + \alpha_i)(\tau_j + \alpha_j)\phi(Y_i, Y_j), \tag{29}$$

for any Y_k on the boundary with $0 < \alpha_k < \lambda_1$.

Computational complexity. The whole process for MTHL algorithm is summarized in Table 2. In particular, steps 6-7 describe Stage I, and steps 10-11 describe Stage II. The loop will continue until the objective function converges. Inside the loop, the computational cost can be divided into two parts. In stage I, time is mainly spent on steps 6 and 7, which costs $\boldsymbol{O}\big(m \cdot (d_v T q + d_v p q + d_v T^2)\big)$ and $\boldsymbol{O}\big(m \cdot (d_v T p + T p q)\big)$ for each view v, respectively. As $\max(p, q) \ll \min(d_v, T)$, the time for each view v reduces to $\boldsymbol{O}\big(m \cdot (d_v T^2 + d_v T)\big)$. When the sample size is way larger than data dimensions $m \gg \max(d_v, T)$, stage I (steps 6-9) is approximately linear to the total sample size $\boldsymbol{O}(mV)$ with all views. Similarly, the mapping process (step 8) also costs $\boldsymbol{O}(mV)$.

In stage II, the most expensive calculation is step 10, resolving the quadratic optimization problem. As we employ the SMO-type decomposition method that modifies two elements in α per iteration, its time complexity heavily depends on the selection of those two elements, referred to as Working Set Selection (WSS). Fan *et al.* [10] propose a WSS technique using second order information, which has time complexity $\boldsymbol{O}(l)$ where l is sample size ($l = mV$). And the WSS also guarantees linear convergence. To summarize, stage II does not cost a lot more than $\boldsymbol{O}(mV)$.

Provided the estimated linear time complexity in each loop, along with the fact that our algorithm always converges after several iterations, we come to the conclusion that our method can be applied in large-scale datasets.

Table 2: MTHL Algorithm.

MTHL Algorithm: Optimize Eq. (10)	
Input:	Multi-view time-series X^v, parameters $\gamma, \lambda_1, \lambda_2, s, p, q$, and maximum iteration *maxIter*;
Output:	Bilinear projections $\{P^v, Q^v\}$, hypersphere $\{Y^*, R\}$;
1:	Normalize time-series samples X^v for each view v;
2:	Compute the Laplacian matrix L_C using s;
3:	Initialize P^v, Q^v, Y^*;
4:	**For** loop *iter* from 1 to *maxIter* **do**
5:	/* *Stage I: gradient descent* */
6:	Compute new P^v for each v via Eq. (16), orthogonalize it;
7:	Compute new Q^v for each v via Eq. (17), orthogonalize it;
8:	Mapping all X^v into the latent space by ${P^v}^T X^v Q^v$;
9:	/**Stage II: Lagrangian duality**/
10:	Optimize Eq. (23) to obtain α;
11:	Calculate objective function f based on Eq. (10);
12:	**If** f converges, **do**
13:	Compute Y^* according to Eq. (21);
14:	Compute R according to Eq. (29);
15:	**return** P^v, Q^v, Y^*, R;
16:	**else**
17:	**Continue**;
18:	**end if**
19:	**end for**

6 EXPERIMENTS

In this section, we first introduce the datasets, performance evaluation, and then report our experimental results.

6.1 Dataset

Synthetic data. We simulate a dynamic network $\mathcal{G}(t)$ to produce synthetic multi-view time-series data. For example, $\mathcal{G}(t)$ can be considered as a city that consists of a set of zones denoted by $\mathcal{V}$, and $\mathcal{E}$ are edges reflecting the traffic, $z(t)$ can represent any location-specific feature, such as a location's functionalities or topics, $w(t)$ can be taken as temporal traffic flows, and T denotes one day (24 hours). Based on *Definition* 3.2, we have $\{z(1), ..., z(T)\} \in \mathbb{R}^{d_z \times T}$ in which each column is snapshot of topic distribution at time t, as well as $\{w(1), ..., w(T)\} \in \mathbb{R}^{d_w \times T}$ in which each column is the snapshot of transportations at time t. From z and w, we can form up two aggregated features: one is aggregated topic vector $Z \in \mathbb{R}^{d_z}$ by summing up z's rows, and the other one is aggregated traffic vector $W \in \mathbb{R}^{d_w}$ by summing up w's rows.

We generate data in three steps: (1) build a network, (2) assign normal attributes, and (3) insert anomalies. In step (1), we apply a random graph generator to construct an underlying network structure; For step (2), we assign normal attributes for each edge and each vertex. Given an arbitrary vertex, we generate a d_z-dimensional aggregated topic distribution Z from a predefined Dirichelet distribution $\mathrm{Dir}(\alpha_z)$ and then divide each topic share into T time steps according to another Dirichelet distribution $\mathrm{Dir}(\alpha_T)$. Here $\mathrm{Dir}(\alpha_z)$ determines topic distribution while $\mathrm{Dir}(\alpha_T)$ determines temporal separation. Traffic data are generated by first assigning daily flow to each edge from a uniform distribution with range $[0, 1]$, and then

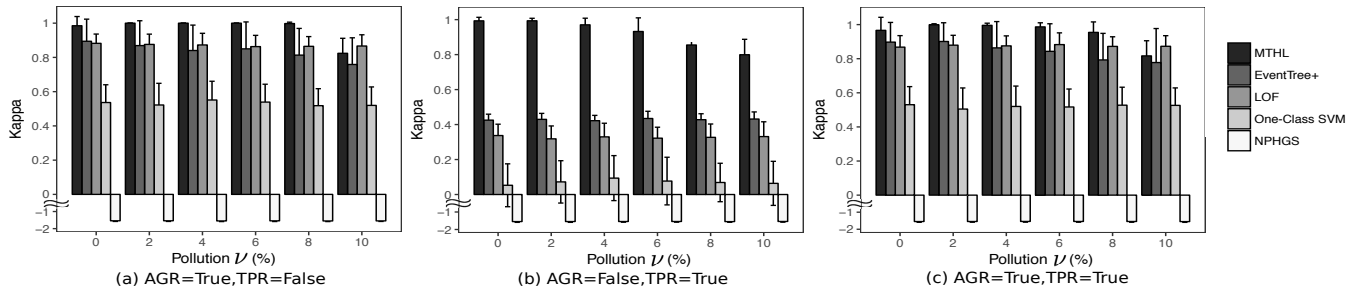

Figure 2: Performances versus anomaly pollution ν for three types of anomalies. There are 50 samples in the reference dataset. Parameters are $\lambda_1 = 0.1, \lambda_2 = 1.0, \sigma = 0.1, \omega = 20\%$.

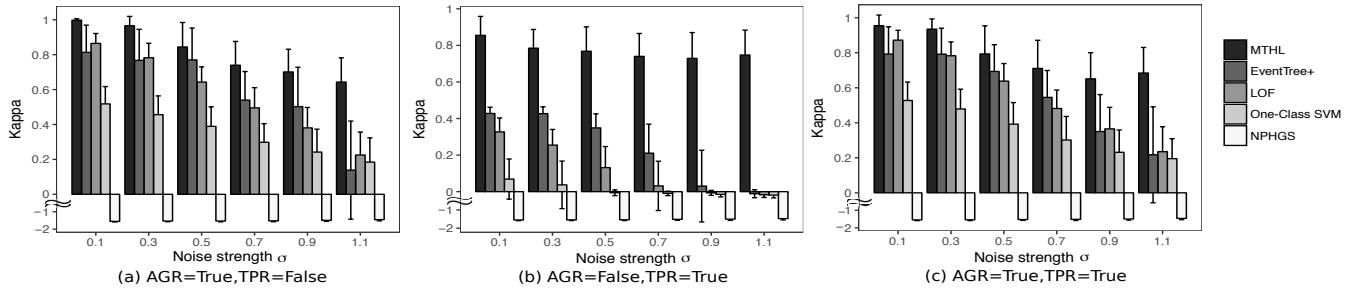

Figure 3: Performance versus noise strength σ for three types of anomalies. There are 50 samples in the reference dataset. Parameters are $\lambda_1 = 0.1, \lambda_2 = 1.0, \nu = 8\%, \omega = 20\%$.

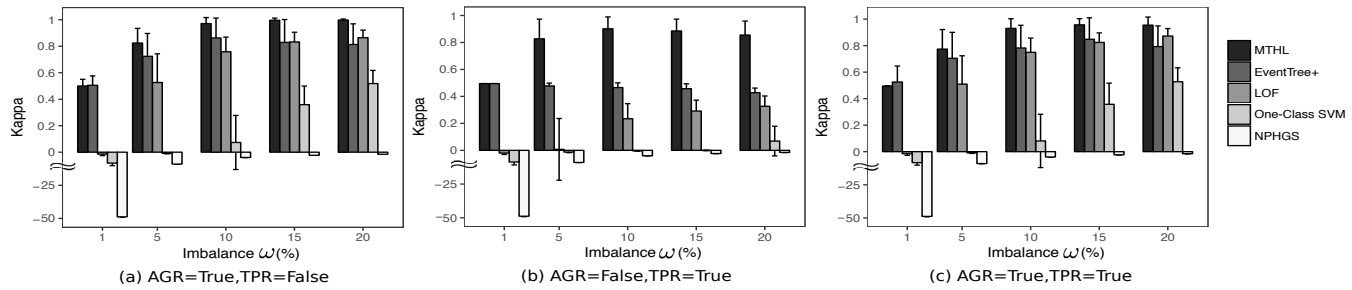

Figure 4: Performances versus data imbalance ω for three types of anomalies. There are 50 samples in the reference dataset. Parameters are $\lambda_1 = 0.1, \lambda_2 = 1.0, \nu = 8\%, \sigma = 0.1$.

segmenting the flow amount into T time steps based on the temporal Dirichelet Dir(α_T). For step (3), we randomly select a subset of vertices and override the assigned attributes in order to inject anomalies.

NYC taxi trips and social media data. We obtain a set of New York City (NYC) taxi trip data from July 2016 to December 2016 [1]. Each trip records the detailed information like pick up time, pick up location, drop off time and drop off location etc. In addition, we also collect Twitter streaming data during the same period of time (from July 2016 to December 2016).

Due to data sparsity, we limit our analysis to the Manhattan area. We extract all trips that are relevant to Manhattan (pick up or drop off in Manhattan), and filter out all geo-tagged tweets posted in this area. In total, there are more than 41 million trips, and about 11 million tweets. Based on administrative boundaries, Manhattan borough can be partitioned into 69 zones (taken as vertices), and the taxi trips are used to construct directed edges. Tweets are allocated into corresponding zones by coordinate information (i.e., longitude and latitude), so that zone-specific topic distribution can be obtained. Topic distribution and taxi trip constitute two different views for mining normal patterns and detecting anomalous phenomena. Finally, we divide a day into 6 slices (4 hours per slice), and obtain two types of multivariate time-series samples.

6.2 Performance Evaluation

Baseline methods. We compare our MTHL algorithm with the following approaches: One-Class Support Vector Machines (One-Class

[1] www.nyc.gov/html/tlc/html/about/tri_record_data.shtml

SVM) [34], the Local Outlier Factor (LOF) [4], Non-Parametric Heterogeneous Graph Scan (NPHGS) [6] and EventTree+ [33]. Among them, One-Class SVM and LOF are two domain-independent methodologies as they take common feature vectors as inputs. In our case, we construct the feature vectors by combining both vertex and edge attributes. NPHGS and EventTree+ are two existing state-of-the-art event detection algorithms that operate on dynamic networks. Both of them define events as subgraphs. NPHGS detects events by finding a connected subgraph that optimizes a nonparametric scan statistic. Although NPHGS is designed for heterogeneous networks, the algorithm can also be employed in homogeneous ones [6]. Besides, EventTree+ detects events through finding a compact subset of vertices that have short distances but high activity level. To apply this algorithm to our synthetic data, we set the distances between all pairs of vertices as 1, and compute the dissimilarity of the current feature vector with the average vector of reference data as node activity level.

Evaluation matrics. The study focuses on detection of three types of anomalies: Type I anomaly only involves changes in aggregated attributes while temporal feature is normal (AGR=True, TPR=False), Type II anomaly only involves changes in temporal attributes but aggregated features are constant (AGR=False, TPR=True), and Type III anomaly involves changes in both types of attributes (AGR=True, TPR=True). In the field of anomaly detection, data is usually highly imbalanced (a small number of outliers). Therefore, in this paper we choose to use Kappa statistic as evaluation matric. Kappa statistic is a comparison of the overall accuracy to the expected random chance accuracy:

$$\text{Kappa} = \frac{(\text{accuracy} - \text{expected accuracy})}{1 - \text{expected accuracy}}. \tag{30}$$

Positive value implies that the proposed algorithm performs better than random guessing, while negative value shows the other way around.

6.3 Experimental Results

We study MTHL's performance from three aspects: performance versus anomaly pollution, performance versus noise, and performance versus data imbalance. The degree of anomaly pollution is denoted as ν to indicate the percentage of abnormal samples in the reference data. The strength of noise is denoted by σ which is the Gaussian deviation. Data imbalance ω is measured by the percentage of anomalies we have injected in the networks.

Performance versus anomaly pollution. We first examine the performance of MTHL and baselines in terms of anomaly pollution. Figure 2(a-c) show the comparison results for three types of anomalies. Bar chart represents average value and error bars represent standard deviation. Each result is obtained from 100 trials (10 networks and 10 trials per network). Generally, MTHL algorithm has the highest Kappa statistic (nearly 1 when $\nu \leq 8\%$) over all baseline methods across all types of anomalies. In particular, the baselines have limited capabilities in terms of Type II anomaly detection, in contrast, our proposed MTHL algorithm can give very promising results. Among all methods, NPHGS obtains negative Kappa statistic (worse than random guessing). The reason is that, NPHGS needs a large number of historical records to obtain good results, but in our case only 50 instances are provided. Additional

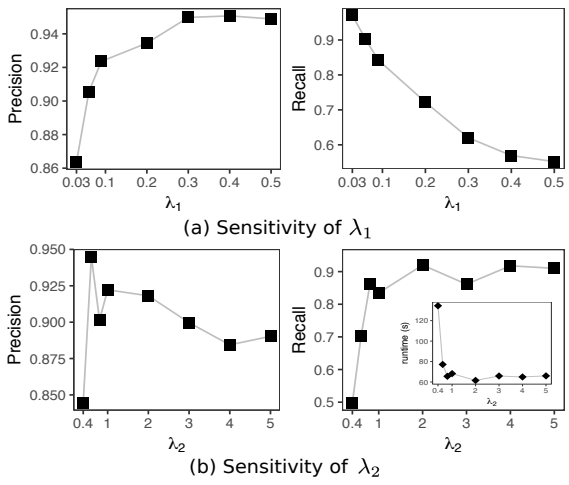

Figure 5: Parameter sensitivity of λ_1 (a) and λ_2 (b). The results are obtained under the Type II case with a balanced dataset. The other parameter are set as $\sigma = 4.0, \nu = 6\%, \omega = 50\%$.

experiments show that if we increase the volume of the reference dataset, NPHGS will exhibit better performance. This observation highlights another advantage of our method, i.e., it only requires a small number of historical records to give satisfactory results.

Performance versus noise. We also examine the robustness of different methods towards noise. The comparison result is shown in Figure 3. Similarly, MTHL outperforms all other baselines under all cases. In particular, its superiority is more prominent under Type II situation. When σ becomes larger than 0.9, EventTree+, LOF and One-Class SVM become nearly equivalent to random guessing, whereas our MTHL approach still shows great advantage with approximately 0.75 Kappa value.

Performance versus data imbalance. As in many applications, anomalies tend to exist for only a small fraction. Hence, we seek to examine MTHL's performance in dealing with imbalanced data. Figure 4 provides the comparison results in terms of different imbalance levels. It reveals that it is more difficult for MTHL and baselines to cope with highly imbalanced data (for the case of 1% anomalies). But in general, MTHL still obtains better detection accuracy than all other baselines, and such superiority is more evident in Type II case.

Parameter sensitivity. In our model, there are two major parameters λ_1 and λ_2. The first one λ_1 controls the strength of penalty we impose on anomalies in the reference data; the second parameter λ_2 controls the influence of temporal smoothing regularization. Figure 5 shows the sensitivity of two parameters under Type II case with balanced data. We jointly present precision and recall. From the figure, we can observe that decreasing λ_1 induces significant decline in precision, while increasing λ_1 results in significant drop in recall. This phenomenon can be well explained. If the λ_1 is set too large, the samples would to be forced to be inside the hypersphere and the radius R would be very large. In this way, MTHL is more likely to take true anomalies as normal samples and thus obtains a low recall value. One the other hand, if λ_1 is set too small,

Table 3: Runtime Results.

Method	Train Time (sec)	Test Time (sec)
MTHL	78.60 (±8.33)	**0.007** ($\pm 6 \times 10^{-4}$)
EventTree+	0.32 (±0.08)	238.18 (±22.87)
NPHGS	150.17 (±33.37)	3.05 (±0.68)
One-Class SVM	0.016 ($\pm 1 \times 10^{-3}$)	0.067 ($\pm 5 \times 10^{-4}$)
LOF	0.14 ($\pm 4 \times 10^{-3}$)	

there is nearly no penalty effect on samples located outside the hypersphere and a small radius R is learned. With a small radius R, MTHL would probably take true normal samples as abnormal ones, and thus obtains a low precision. To balance the effect of precision and recall, we choose $\lambda_1 = 0.1$ in our algorithm. In terms of λ_2, we also observe mixed results. In a similar way, we select a value $\lambda_2 = 1.0$ by jointly considering precision and recall performance. In addition, we find that if λ_2 is less than 1.0, the convergence time would significantly increase to a higher level. This observation also validates our choice of $\lambda_2 = 1.0$.

Runtime results. Table 3 shows the runtime comparison between our MTHL and baseline methods. In particular, we report training and test runtime separately for each method (except for LOF). For MTHL, training time refers to the time spent on hypersphere learning using the reference dataset, and test time refers to the time used to decide whether a new observation is abnormal or not. It reveals that MTHL can make a very fast decision within one millisecond. This observation is crucial because many applications require timely and fast decision making. In contrast, EventTree+ spends much more time (nearly 4 minutes) in test stage. Although NPHGS costs little time to test new observations, it costs twice as much time as MTHL in the training stage. Actually, in the training stage, NPHGS needs to calculate empirical p-values by comparing the current observation to each historical record. That means NPHGS runtime is highly dependent on the size of the reference dataset. One-Class SVM and LOF are two fast approaches, taking the least total time in the anomaly detection process. To summarize, our proposed algorithm can obtain the best performance by spending the comparable least time like One-Class SVM and LOF.

Case study in real-world data. In Figure 6, we show two important events occurred in Manhattan: (a) Post-Election Day on November 10, 2016 and (b) New Year's Eve on December 31, 2016. In each example, we shows the anomalous zones detected by our proposed algorithm MTHL (left panel) and by EventTree+ (right panel). As MTHL can output anomaly scores, we use dark color to indicate a large value. For EventTree+ algorithm, we integrate traffic and topic features into one single vector, and consider each zone's dissimilarity to its regular norm as the so-called attribute "activity level". Unlike MTHL, it outputs binary labels. For both methods, we take the preceding 30 days as a basis to construct reference dataset.

Figure 6(a) shows the anomalous regions detected for the post-election day (November 10, 2016). Donald Trump was elected to be the 45th president of the United States on November 9, 2016. Trump's victory sparked nationwide Anti-Trump protests during the following days [2]. Trump's opponents either took the street or turned to social media to express their opposition to Trump's policies. By comparison, we can see that MTHL obtains more meaningful detection results. In specifically, MTHL (left panel) suggests that Midtown Center, Midtown East, Upper Manhattan and Greenwich Village exhibit anomalous activities. Considering that those zones are either near to Trump real estate or the places where universities and colleges are located (marked on maps), anomalous behaviors are more likely to emerge. However, EventTree+ (right panel) fails to detect those critical zones.

Figure 6(b) shows the anomalous regions detected for the New Year's Eve (December 31, 2016). MTHL (left panel) tells that Times Square and the nearby zones seem to be anomalous. This observation is probably related to the traditional event "Ball Drop" held at Times Square (marked on map) every year. It is reported that an estimated one million people gather in Times Square to celebrate the festival and watch musical performances at that night [3]. This large-scale gathering of people would influence the traffics in neighboring areas; therefore, we can observe anomalous phenomena in Midtown Manhattan and Upper East Side. In contrast, EventTree+ (right panel) considers half of Manhattan area as anomalous zones, which provides less practical value.

To summarize, the two events in New York City have demonstrated that our proposed MTHL can obtain reliable and meaningful detection results, which suggests its potential application in a real-world domain.

7 CONCLUSIONS

In this paper, we develop a novel MTHL framework for anomaly detection in dynamic networks. Compared to traditional techniques, our proposed MTHL has prominent superiority in detecting events that involves anomalous temporal dynamics. Our work highlights the necessity of the extraction of temporal patterns, and the exploitation of multiple data sources. As part of future work, we plan to relax the assumption that the streaming data can be partitioned into short, periodic and well-aligned temporal segments having similar patterns. Instead, we seek to mine the evolution pattern in the infinite time span in order to detect more potential anomalies. In addition, we plan to incorporate the interplay among individual objects (e.g., vertices or edges) into analysis, so as to detect large-scale anomalies across regions or to predict anomaly spreading in networks.

ACKNOWLEDGMENTS

The authors would like to acknowledge the support from the NSF Grants No. 1634944 and No. 1637067. Any opinions, findings, and conclusions or recommendations expressed in this material do not necessarily reflect the views of the funding sources.

REFERENCES

[1] James Abello, Tina Eliassi-Rad, and Nishchal Devanur. 2010. Detecting novel discrepancies in communication networks. In *Data Mining (ICDM), 2010 IEEE 10th International Conference on*. IEEE, 8–17.

[2] https://www.nytimes.com/interactive/2016/11/12/us/elections/photographs-from-anti-trump-protests.html

[3] http://www.timessquarenyc.org/index.aspx

(a) Post-Election Day (b) New Year's Eve

Figure 6: Two events detected in Manhattan: (a) Post-Election Day (Nov 10, 2016) and (b) New Year's Eve (Dec 31, 2016). It shows the results obtained by MTHL (left panel) and also by EventTree+ (right panel). Parameters are set $\lambda_1 = 0.1$ and $\lambda_2 = 1.0$ for MTHL, and $\gamma = 1.0$ for EventTree+.

[2] Leman Akoglu, Hanghang Tong, and Danai Koutra. 2015. Graph based anomaly detection and description: a survey. *Data Mining and Knowledge Discovery* 29, 3 (2015), 626–688.

[3] Steffen Bickel and Tobias Scheffer. 2004. Multi-View Clustering.. In *ICDM*, Vol. 4. 19–26.

[4] Markus M Breunig, Hans-Peter Kriegel, Raymond T Ng, and Jörg Sander. 2000. LOF: identifying density-based local outliers. In *ACM sigmod record*, Vol. 29. ACM, 93–104.

[5] Nan Cao, Chaoguang Lin, Qiuhan Zhu, Yu-Ru Lin, Xian Teng, and Xidao Wen. 2017. Voila: Visual Anomaly Detection and Monitoring with Streaming Spatiotemporal Data. In *IEEE Symposium on Visual Analytics Science and Technology (VAST 2017)*.

[6] Feng Chen and Daniel B Neill. 2014. Non-parametric scan statistics for event detection and forecasting in heterogeneous social media graphs. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 1166–1175.

[7] Zhengzhang Chen, William Hendrix, and Nagiza F Samatova. 2012. Community-based anomaly detection in evolutionary networks. *Journal of Intelligent Information Systems* 39, 1 (2012), 59–85.

[8] Qi Ding, Natallia Katenka, Paul Barford, Eric Kolaczyk, and Mark Crovella. 2012. Intrusion as (anti) social communication: characterization and detection. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 886–894.

[9] Zhengming Ding and Yun Fu. 2014. Low-rank common subspace for multi-view learning. In *Data Mining (ICDM), 2014 IEEE International Conference on*. IEEE, 110–119.

[10] Rong-En Fan, Pai-Hsuen Chen, and Chih-Jen Lin. 2005. Working set selection using second order information for training support vector machines. *Journal of machine learning research* 6, Dec (2005), 1889–1918.

[11] Brendan J Frey and Delbert Dueck. 2006. Mixture modeling by affinity propagation. *Advances in neural information processing systems* 18 (2006), 379.

[12] Jing Gao, Wei Fan, Deepak Turaga, Srinivasan Parthasarathy, and Jiawei Han. 2011. A spectral framework for detecting inconsistency across multi-source object relationships. In *Data Mining (ICDM), 2011 IEEE 11th International Conference on*. IEEE, 1050–1055.

[13] Yuhong Guo. 2013. Convex Subspace Representation Learning from Multi-View Data.. In *AAAI*, Vol. 1. 2.

[14] Manish Gupta, Jing Gao, Charu Aggarwal, and Jiawei Han. 2014. Outlier detection for temporal data. *Synthesis Lectures on Data Mining and Knowledge Discovery* 5, 1 (2014), 1–129.

[15] Manish Gupta, Jing Gao, Yizhou Sun, and Jiawei Han. 2012. Community trend outlier detection using soft temporal pattern mining. *Machine Learning and Knowledge Discovery in Databases* (2012), 692–708.

[16] Manish Gupta, Jing Gao, Yizhou Sun, and Jiawei Han. 2012. Integrating community matching and outlier detection for mining evolutionary community outliers. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 859–867.

[17] Zan Huang and Daniel D Zeng. 2006. A link prediction approach to anomalous email detection. In *Systems, Man and Cybernetics, 2006. SMC'06. IEEE International Conference on*, Vol. 2. IEEE, 1131–1136.

[18] Tengfei Ji, Dongqing Yang, and Jun Gao. 2013. Incremental local evolutionary outlier detection for dynamic social networks. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 1–15.

[19] Meina Kan, Shiguang Shan, Haihong Zhang, Shihong Lao, and Xilin Chen. 2016. Multi-view discriminant analysis. *IEEE transactions on pattern analysis and machine intelligence* 38, 1 (2016), 188–194.

[20] Danai Koutra, Joshua T Vogelstein, and Christos Faloutsos. 2013. Deltacon: A principled massive-graph similarity function. In *Proceedings of the 2013 SIAM International Conference on Data Mining*. SIAM, 162–170.

[21] Abhishek Kumar, Piyush Rai, and Hal Daume. 2011. Co-regularized multi-view spectral clustering. In *Advances in neural information processing systems*. 1413–1421.

[22] Sheng Li, Yaliang Li, and Yun Fu. 2016. Multi-View Time Series Classification: A Discriminative Bilinear Projection Approach. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*. ACM, 989–998.

[23] Sheng Li, Ming Shao, and Yun Fu. 2015. Multi-view low-rank analysis for outlier detection. In *Proceedings of the 2015 SIAM International Conference on Data Mining*. SIAM, 748–756.

[24] Xiaolei Li, Zhenhui Li, Jiawei Han, and Jae-Gil Lee. 2009. Temporal outlier detection in vehicle traffic data. In *Data Engineering, 2009. ICDE'09. IEEE 25th International Conference on*. IEEE, 1319–1322.

[25] Yu Liu, Baojian Zhou, Feng Chen, and David W Cheung. 2016. Graph Topic Scan Statistic for Spatial Event Detection. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*. ACM, 489–498.

[26] Alejandro Marcos Alvarez, Makoto Yamada, Akisato Kimura, and Tomoharu Iwata. 2013. Clustering-based anomaly detection in multi-view data. In *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*. ACM, 1545–1548.

[27] Misael Mongiovi, Petko Bogdanov, Razvan Ranca, Evangelos E Papalexakis, Christos Faloutsos, and Ambuj K Singh. 2013. Netspot: Spotting significant anomalous regions on dynamic networks. In *Proceedings of the 2013 SIAM International Conference on Data Mining*. SIAM, 28–36.

[28] Emmanuel Muller, Ira Assent, Patricia Iglesias, Yvonne Mulle, and Klemens Bohm. 2012. Outlier ranking via subspace analysis in multiple views of the data. In *Data Mining (ICDM), 2012 IEEE 12th International Conference on*. IEEE, 529–538.

[29] Joshua Neil, Curtis Hash, Alexander Brugh, Mike Fisk, and Curtis B Storlie. 2013. Scan statistics for the online detection of locally anomalous subgraphs. *Technometrics* 55, 4 (2013), 403–414.

[30] Daniel B Neill and Gregory F Cooper. 2010. A multivariate Bayesian scan statistic for early event detection and characterization. *Machine learning* 79, 3 (2010), 261–282.

[31] Arvind Ramanathan, Pratul K Agarwal, Maria Kurnikova, and Christopher J Langmead. 2010. An online approach for mining collective behaviors from

molecular dynamics simulations. *Journal of Computational Biology* 17, 3 (2010), 309–324.

[32] Stephen Ranshous, Shitian Shen, Danai Koutra, Steve Harenberg, Christos Faloutsos, and Nagiza F Samatova. 2015. Anomaly detection in dynamic networks: a survey. *Wiley Interdisciplinary Reviews: Computational Statistics* 7, 3 (2015), 223–247.

[33] Polina Rozenshtein, Aris Anagnostopoulos, Aristides Gionis, and Nikolaj Tatti. 2014. Event detection in activity networks. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 1176–1185.

[34] Bernhard Schölkopf, Robert C Williamson, Alexander J Smola, John Shawe-Taylor, John C Platt, and others. 1999. Support vector method for novelty detection.. In *NIPS*, Vol. 12. 582–588.

[35] Jimeng Sun, Dacheng Tao, and Christos Faloutsos. 2006. Beyond streams and graphs: dynamic tensor analysis. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 374–383.

[36] Jimeng Sun, Yinglian Xie, Hui Zhang, and Christos Faloutsos. 2007. Less is more: Compact matrix decomposition for large sparse graphs. In *Proceedings of the 2007 SIAM International Conference on Data Mining*. SIAM, 366–377.

[37] David MJ Tax and Robert PW Duin. 2004. Support vector data description. *Machine learning* 54, 1 (2004), 45–66.

[38] Xidao Wen, Yu-Ru Lin, and Konstantinos Pelechrinis. 2016. PairFac: Event Analytics through Discriminant Tensor Factorization. In *Proc. of The 25th ACM International Conference on Information and Knowledge Management (CIKM 2016)*. https://doi.org/10.1145/2983323.2983837

[39] Martha White, Xinhua Zhang, Dale Schuurmans, and Yao-liang Yu. 2012. Convex multi-view subspace learning. In *Advances in Neural Information Processing Systems*. 1673–1681.