



WINTENDED: WInDowed TENSor decomposition for Densification Event Detection in time-evolving networks

Sofia Fernandes¹ · Hadi Fanaee-T² · João Gama¹ · Leo Tišljarić³ · Tomislav Šmuc⁴

Received: 27 February 2020 / Revised: 11 February 2021 / Accepted: 12 April 2021 /

Published online: 11 May 2021

© The Author(s), under exclusive licence to Springer Science+Business Media LLC, part of Springer Nature 2021

Abstract

Densification events in time-evolving networks refer to instants in which the network density, that is, the number of edges, is substantially larger than in the remaining. These events can occur at a global level, involving the majority of the nodes in the network, or at a local level involving only a subset of nodes. While global densification events affect the overall structure of the network, the same does not hold in local densification events, which may remain undetectable by the existing detection methods. In order to address this issue, we propose WInDowed TENSor decomposition for Densification Event Detection (WINTENDED) for the detection and characterization of both global and local densification events. Our method combines a sliding window decomposition with statistical tools to capture the local dynamics of the network and automatically find the irregular behaviours. According to our experimental evaluation, WINTENDED is able to spot global densification events at least as accurately as its competitors, while also being able to find local densification events, on the contrary to its competitors.

Editor: Petra Kralj Novak

✉ Sofia Fernandes
sdsf@inesctec.pt

Hadi Fanaee-T
hadi.fanaee@hh.se

João Gama
jgama@fep.up.pt

Leo Tišljarić
ltisljaric@fpz.unizg.hr

Tomislav Šmuc
smuc@irb.hr

¹ LIAAD, INESC TEC, University of Porto, Porto, Portugal

² Center for Applied Intelligent Systems Research (CAISR), Halmstad University, Halmstad, Sweden

³ Faculty of Transport and Traffic Sciences, University of Zagreb, Zagreb, Croatia

⁴ Division of Electronics, Laboratory for Machine Learning and Knowledge Representation, Rudjer Boskovic Institute, Zagreb, Croatia

Keywords Time-evolving networks · Tensor decomposition · Event detection

1 Introduction

In the process of mining dynamic networks, the task of finding irregular isolated communication patterns is referred to as event detection. Not only it is important to know at what time an event occurred but also, which nodes were involved in it. For example, if we consider the Facebook wall-posts of a small group of students, then a substantial increase in the overall number of wall-posts at day t is usually unexpected. However, knowing that a substantially larger number of wall-posts was observed at day t is not enough to characterize this potential anomaly. If, additionally, we know that the posts at day t were mainly made at subject's "a" wall then, it is probable that day t corresponded to an event in subject's "a" life such as birthday or graduation (and this peak on the wall-posts is originated by the congratulations messages).

On the other hand, we should also take into account that events may occur at different scales, not necessarily involving all the nodes. As an illustrative example, let us consider a time-evolving network modelling the monthly exchange of emails between employees of a company. Then, the announcement that the company is going to close is expected to generate a lot of discussion between all the employees thus originating an abrupt interaction peak involving the whole network. On the other hand, the scheduling of a meeting between the employees of a specific department is expected to lead to the increasing of the emails exchanged between the employees of such department only. Both of these scenarios are illustrated in Fig. 1: the company shutdown scenario may be regarded as a global event since it involves the majority of the employees and the department meeting corresponds to a local event because it only involves a small group of employees. Since these events correspond to an abnormal increase in the number of edges, they are referred to as densification events.

Currently, the existing densification event detectors fail at incorporating both the previous features (that is, event characterization and local event detection). Thus, in order to tackle such gap, we propose WINDowed TENSOR decomposition for Densification Event Detection (WINTENDED), an event detector that combines tensor decomposition with statistical tools to automatically spot and characterize densification events in time-evolving networks. In this context, we revise and extend our previous work (Fernandes et al., 2019) by providing: (i) a more detailed description of the problem addressed and proposed solution, (ii) a more complete literature review and (iii) a more extensive empirical evaluation (including more baselines and networks) as well as a more detailed and complete analysis of the results. Additionally, we provide some insights on how to adapt our method to detect events associated with an abrupt change in the edges weights, not necessarily associated with densification events.

This work is organized as follows. In Sect. 2, we discuss the literature on event detection. In Sect. 3 we provide the background needed to understand this work. The problem addressed is formally formulated in Sect. 4, the proposed method is described in Sect. 5 and the corresponding experiments are presented in Sect. 6. We conclude in Sect. 7.

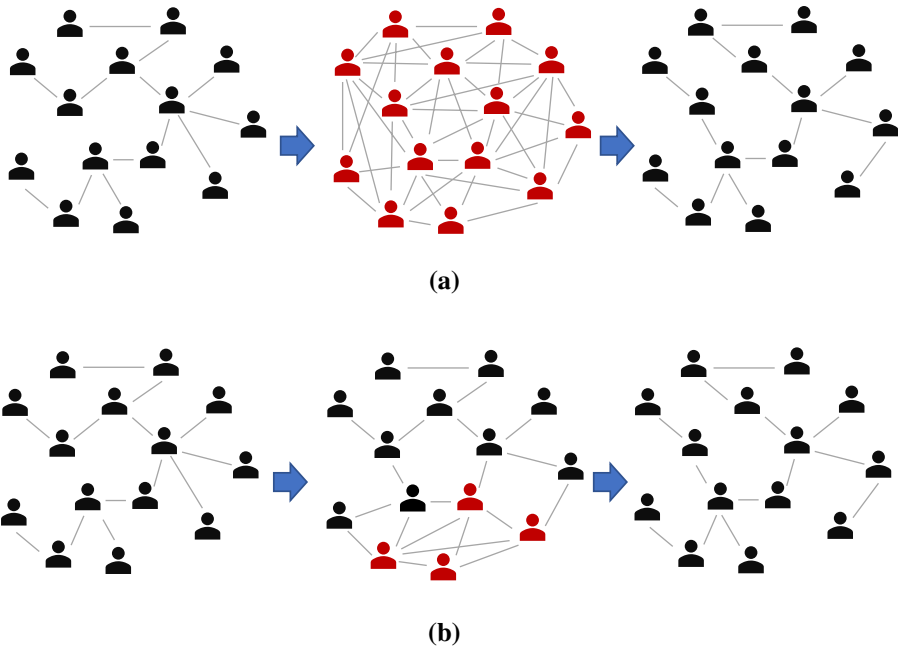


Fig. 1 Illustration of event scale types using the email network of employees in a company: **a** global densification event - there is a substantial increase of communication between all the employees in the company **b** local densification event - there is a substantial increase in the communications between a subset of employees (in red) (Color figure online)

2 Related work

Event detection in time-evolving networks has been traditionally approached via similarity-based approaches. Generally, the idea exploited consists of accounting for the similarity (or dissimilarity) between successive network states. In other words, network state at time t is compared with the previous network state(s), and if a high dissimilarity is observed, then instant t is likely to model an unexpected behaviour. The dissimilarity is interpreted as an anomaly score so that the higher the dissimilarity, the higher the abnormality of the instant. These methods differ mainly on the network representation and similarity metric. In Shoubridge et al. (2002), Kapsabelis et al. (2007), the authors used the graph edit distance and its variations to measure the level of “change” between two network states. In other works, the network state is first summarized and the similarity is measured in the new representation space. The new representation is usually based on node features over time, such as degree or number of triangles in the egonet (Akoglu and Faloutsos, 2010; Berlingerio et al., 2013). It is noteworthy that most of these methods focus on global events detection.

Additionally, this problem has also been approached from matrix and tensor decomposition perspectives. In particular, assuming that the network is modelled by a sequence of (adjacency) matrices, the standard approach consists of measuring the network approximation error in each timestamp. In more detail, if tensor decomposition is being considered, all the network

instants available are (jointly) decomposed and the reconstruction error at time t is obtained (Kolda and Sun, 2008; Koutra et al., 2012). Instants of time for which the reconstruction error is high indicate that the network behaviour at that instants is different from the general behaviour and, consequently, are associated with a high abnormality. In this tensor decomposition-based approach, all the network instants must be available, thus making the method unsuitable for real-time analysis.

Other tensor decomposition-based approaches include the exploratory analysis of the decomposition factors (Papalexakis et al., 2014) and the combination with multivariate statistics (Fanaee-T and Gama, 2016). The main limitation of the exploratory analysis approach is its supervised and non-automatic nature. On the other hand, the method proposed in Fanaee-T and Gama (2016) is automatic but was not designed for neither local event detection nor characterization of events.

Another related problem is the dense block detection in sparse tensors. In this context Shin et al. (2016, 2017) developed methods for finding periods of time in which a subnetwork densification occurs. The main difference between this problem and the one addressed in this work is that in our setting the densification occurs at a single time instant, while in dense block detection, the densification may occur in a period of multiple instants.

Recently, an ensemble approach, SELECT, has been proposed by Rayana and Akoglu (2016), which combines some of the previous approaches (including similarity Akoglu and Faloutsos 2010 and matrix decomposition approaches Papadimitriou et al. 2005), thus being able to improve accuracy.

Our method stands out from the standard tensor decomposition-based approaches in the sense that (i) it processes the network in sliding window mode, thus being able to capture local dynamics as it is demonstrated in this work, and (ii) it is automatic. Besides, it is able to overcome (non-tensor-based) state-of-the-art predictors limitations, such as the inability to spot local events and to identify the source of the anomalous instant. Nonetheless, it should be noted that our method was designed to spot densification events only.

For a complete overview of the methods in event detection and their limitations, the reader may refer to Akoglu et al. (2015), Ranshous et al. (2015).

3 Background

3.1 Tensors

Multi-dimensional numerical arrays are referred to as tensors. In this context, a M -dimensional tensor is an array: $\mathcal{X} \in \mathbb{R}^{N_1 \times N_2 \times \dots \times N_M}$, in which the value of the entry (i_1, i_2, \dots, i_M) is represented as $\mathcal{X}(i_1, i_2, \dots, i_M)$. Moreover $\mathcal{X}(:, :, \dots, I_d, \dots, :)$ represents the restriction of \mathcal{X} to dimension I_d in mode d . In other words, it represents the tensor obtained by fixing the d^{th} dimension to I_d . The tensor size is given by $N_1 \times N_2 \times \dots \times N_M$.

Similarly to matrices, there are decomposition methods for tensors. In this work, we consider the well-known CANDECOMP/PARAFAC (CP) (Kolda and Bader, 2009). Given a 3-order tensor, $\mathcal{X} \in \mathbb{R}^{N_1 \times N_2 \times N_3}$, the goal of CP is to find R (factor) vectors $\mathbf{a}_r \in \mathbb{R}^{N_1}$, $\mathbf{b}_r \in \mathbb{R}^{N_2}$, $\mathbf{c}_r \in \mathbb{R}^{N_3}$, where $1 \leq r \leq R$, such that:

$$\mathcal{X} \approx \sum_{r=1}^R \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r \quad (1)$$

The vectors associated with each mode may be column-wise arranged into the so called factor matrices \mathbf{A}, \mathbf{B} and \mathbf{C} : $\mathbf{A} = [\mathbf{a}_1 | \mathbf{a}_2 | \dots | \mathbf{a}_R] \in \mathbb{R}^{N_1 \times R}$, $\mathbf{B} = [\mathbf{b}_1 | \mathbf{b}_2 | \dots | \mathbf{b}_R] \in \mathbb{R}^{N_2 \times R}$ and $\mathbf{C} = [\mathbf{c}_1 | \mathbf{c}_2 | \dots | \mathbf{c}_R] \in \mathbb{R}^{N_3 \times R}$.

The approximation error of the resulting tensor is computed as:

$$\left\| \mathcal{X} - \sum_{r=1}^R \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r \right\|_F \equiv \sqrt{\sum_{i=1}^{N_1} \sum_{j=1}^{N_2} \sum_{k=1}^{N_3} \left(\left[\mathcal{X} - \sum_{r=1}^R \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r \right] (i, j, k) \right)^2}$$

so that the error in slice K is given by

$$\left\| \left[\mathcal{X} - \sum_{r=1}^R \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r \right] (:, :, K) \right\|_F \equiv \sqrt{\sum_{i=1}^{N_1} \sum_{j=1}^{N_2} \left(\left[\mathcal{X} - \sum_{r=1}^R \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r \right] (i, j, K) \right)^2} \quad (2)$$

which, in the case of time-evolving networks, gives the reconstruction error at instant K .

Since in this work we are dealing with networks (modelled as non-negative count data), we exploit Alternating Poisson Regression CP (CP-APR) (Chi and Kolda 2012), which takes into account these constraints.

4 Problem addressed

By adapting the event definition formulated by Ranshous et al. (2015), to refer to densification events only and to also include local events, the problem addressed in this work is as follows:

Given a time-evolving network G , characterized by the sequence of adjacency matrices over time instants 1 to L , $\{A_G^t\}_1^L$, find the instants of time τ , with $1 \leq \tau \leq L$, for which there is a substantial density increase of a subnetwork G' .

In this context, the density of subgraph G' , which corresponds to the ratio of existing edges among all possible edges, may be regarded as a scoring function $s_{G'}$. Therefore, we are interested in finding subgraphs G' such that there exists an instant of time τ satisfying:

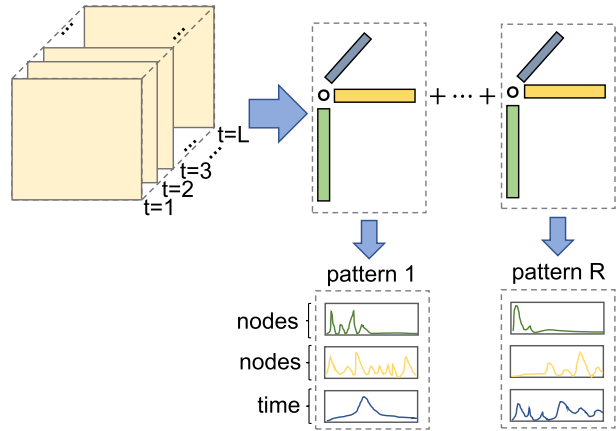
$$s_{G'}^\tau - s_{G'}^t > \delta, \forall t \neq \tau. \quad (3)$$

for $\delta > 0$ and $s_{G'}^t$ denoting the density of subgraph G' at time t . The event is dubbed as global if it involves the majority of the nodes (Fig. 1a) or as local otherwise (Fig. 1b).

It is worth mentioning that, in some contexts, an abrupt and substantial decrease in the interaction level can also be regarded as an anomaly (Ranshous et al., 2015). For example, in a daily proximity network between students in a school, an abrupt lack of communication between one of the students and the others may indicate that the student missed school at the given day. We leave this type of event to be considered in a future extension of the proposed framework.

For the sake of simplicity and since in this work we target only densification events, throughout the rest of the manuscript, whenever we refer to an event we are referring to a densification event (unless explicitly stated differently).

Fig. 2 Illustration of the pattern extraction via tensor decomposition within the time window



5 Proposed method

Idea Tensor decomposition has been successfully applied to detect communication patterns in evolving networks (Papalexakis et al., 2012; Jeon et al., 2016). The concepts unveiled allow us to discover which entities interact and when. Therefore, the idea exploited in our approach is to search for (anomalous) communication peaks in the patterns discovered by tensor decomposition.

Data A network is modelled as a tensor of type *entities* × *entities* × *time* corresponding to the sequence of adjacency matrices over time. The network is processed using a sliding window \mathcal{W} of length L , which corresponds to the number of timestamps in the window.

Method Given a tensor modelling the network time window \mathcal{W} , WINTENDED encompasses the following stages, summarized in algorithm 1 and illustrated in Fig. 5.

1. **Decomposing the tensor window:**

We decompose \mathcal{W} by applying CP-APR with R components: $\mathcal{W} \approx \sum_{r=1}^R \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{t}_r$, with $\mathbf{a}_r, \mathbf{b}_r, \mathbf{t}_r \geq 0$. \mathbf{a}_r and \mathbf{b}_r are associated with entities dimensions and \mathbf{t}_r is associated with the time dimension. Each concept r , defined by $\{\mathbf{a}_r, \mathbf{b}_r, \mathbf{t}_r\}$, induces a subgraph G'_r formed by the nodes so that

$$\begin{cases} nodes_{a_r} = \{i : \mathbf{a}_r(i) > 0\} \\ nodes_{b_r} = \{i : \mathbf{b}_r(i) > 0\} \end{cases}; \tag{4}$$

(we refer to these nodes as the active nodes in concept r). Their time activity is described by \mathbf{t}_r .

This process is illustrated in Fig. 2.

2. **Identifying the anomaly candidates:** Since each \mathbf{t}_r may be interpreted as a communication/activity vector, then an isolated abrupt peak corresponds to an isolated outlier in this vector.

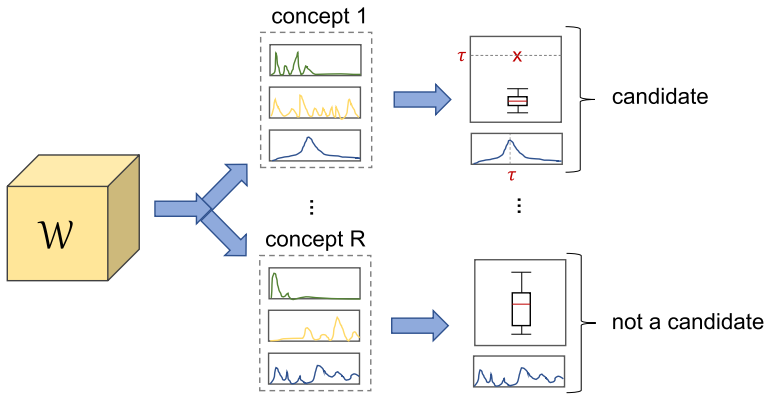


Fig. 3 Illustrative example of the anomaly candidate identification in a given time window \mathcal{W} : candidates are the concepts for which the time factor vector has an isolated extreme outlier

Based on this, for each concept r , we search for the instant of time τ , such that $\tau = \operatorname{argmax}_i t_r(i)$, which is an anomalous instant candidate. Moreover, we verify if the following condition holds

$$t_r(\tau) > Q3 + 3IQR \tag{5}$$

where $Q1$ and $Q3$ are the first and third quartile, respectively, and $IQR = Q3 - Q1$ (Dawson 2011), as illustrated in Fig. 3.

Intuitively, the time points satisfying (5) are far from the expected level of interaction, modelled by t_r . Therefore, if condition (5) does not hold for τ , the maximum activity in the time window falls within the expected range of values and the instant is not an event. Otherwise, since the temporal activity vector approximates a Poisson distribution, it is necessary to check the level of abnormality of τ . In other words, we need to check if τ is associated with an isolated high value.

To meet this end, we must verify if

$$t_r(\tau) - t_r(\bar{\tau}) > \gamma \tag{6}$$

for $\bar{\tau} = \operatorname{argmax}_{i \neq \tau} t_r(i)$ and $\gamma > 0$. In particular, if $t_r(\bar{\tau}) \leq Q3 + 3IQR$, then (6) holds for some γ and τ is flagged as an anomaly instant. Otherwise, since $\sum_{i=1}^R t_r(i) = 1$, it is likely that $Q3 + 3IQR \approx 0$ (as in Fig. 4), in which case either (i) τ and $\bar{\tau}$ fall within a similar range of values (consequently, $t_r(\tau) - t_r(\bar{\tau}) \leq \gamma$ and τ is not an anomalous instant) or (ii) $t_r(\tau)$ is an isolated high value and corresponds to an event.

Therefore, τ is flagged as anomaly candidate if it satisfies (5) and (6). This process is applied through *ispeak* in algorithm 1.

In order to improve WINTENDED, further research should be carried out to derive a rule to spot also sparsification events. This feature is left as future work.

3. **Characterizing and verifying the anomaly candidates:** The goal of this stage is to discard the anomaly candidates that do not represent a densification pattern. To achieve our aim, we construct the graph induced from the anomaly candidate and quantify its presence in the remaining timestamps.

In more detail, let $\hat{G}'_r = (V'_r, E'_r)$ be the subgraph associated with the anomaly candidate, where:

- (a) the nodes set is given by $V'_r = nodes_{a_r} \cup nodes_{b_r}$;
- (b) the adjacency matrix is given by $A_{\hat{G}'_r}(nodes_{a_r}, nodes_{b_r})$ so that $A_{\hat{G}} = \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{t}_r(\tau)$ (with τ being the anomalous instant).

Then, we carry out a “cleaning procedure” by discarding from the subgraph \hat{G}'_r the nodes which are less active in the original networks as they do not substantially contribute to the densification event.

Finally, we quantify the anomaly level of the candidate \hat{G}'_r based on three statistics, measured in each timestamp of the time span:

- the density of the subgraph induced by V'_r in the original network;
- the average weighted node degree of the graph induced by V'_r in the original network;
- the rate of edges in the anomaly candidate subgraph \hat{G}'_r that are also present in the original graph.

The candidate is flagged as event if for all those three vectors, the anomalous timestamp τ is an isolated extreme outlier. Otherwise, it means that the candidate does not substantially deviate from the behavior observed in the time window. This process is carried out when executing *isanomaloussubgraph* in algorithm 1.

For each component flagged as event, we store the anomalous instant τ and an activity score which accounts for the size of the corresponding subgraph, computed as:

$$activity_score_{G'} = \frac{|nodes_{a_r} \cup nodes_{b_r}|}{N}, \quad (7)$$

where N is the total number of nodes in the network. Additionally, we also store the event pattern corresponding to the concept $r: \{\mathbf{a}_r, \mathbf{b}_r, \mathbf{c}_r\}$.

Ensemble The main parameters of WINTENDED are the number of factors to consider in tensor decomposition and the window length.

Regarding the number of components, the choice of an appropriate value is not straightforward. In particular, the suitable number of components depends not only on the type of data (Papalexakis 2016) but also on the task at hands (da Silva Fernandes et al. 2017). Consequently, it is not guaranteed that a given state-of-the-art estimator provides the most adequate estimate for our scenario. Based on this, and inspired by Dunlavy et al. (2011), Hosseinimotlagh and Papalexakis (2018), we considered models with varying number of components so that we are able to understand the “strength”/

Table 1 Illustrative example on how to rank the events detected by the ensemble based on the number of models detecting them and the number of nodes in the anomalous subgraph (activity score). Higher ranks are associated with more abnormality

	τ_1	τ_2	τ_3	τ_4
number of models detecting the event	4	2	3	2
<i>activity_score</i>	0.5	0.53	0.33	0.67
final abnormality rank	1	4	2	3

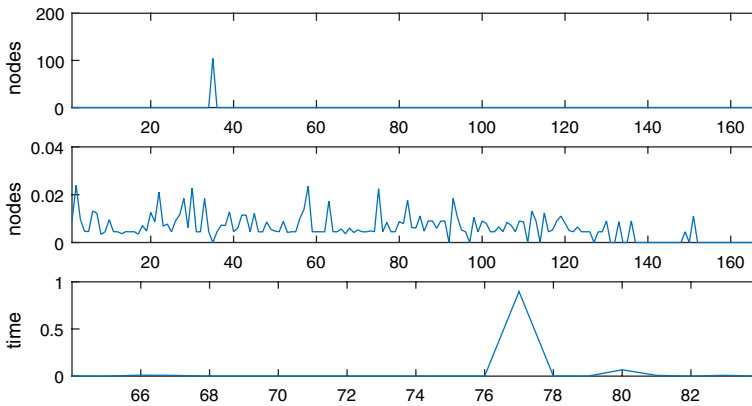


Fig. 4 Event detected by WINTENDED in the manufacturing e-mail network

relevance of the pattern: if a pattern is predominant, then it is expected to be present regardless of the number of components used.

With respect to window length, we also considered varying values with the goal of making the method more robust to seasonality effects arising from using a single window length. Note that if there is a periodical interaction peak at each t timestamps, then by considering a window length of less than t timestamps, the peak is isolated within the time window and the method assumes that such peak is abnormal. On the other hand, by considering a window length higher than t then the periodicity is captured and the peaks are not flagged as events. Therefore, the choice of the set of values should take into account the time granularity and periodicity (if known). In particular, by setting L_K as the maximum window length, we are guaranteeing that the events spotted by WINTENDED do not repeat within a time span of length L_K . Still, it should be noted that L_K should be relatively small in order to ensure that tensor decomposition accurately models the network dynamics.

Let $\{R_i\}_{i=1}^M$ and $\{L_i\}_{i=1}^K$ be the set of numbers of components and the set of window lengths to be considered, respectively. Then, we generate a model M_{ij} using R_i

components and a window length of L_j for each possible combination. Finally, the results of the several models are combined. The events are ranked according to the number of models detecting them and their number of participants (that is, their activity score, computed using (7)). In more detail, the anomaly score is as high as the number of models detecting it. Moreover, if two events are detected by the same number of models, then the event involving more nodes (that is, having a higher activity score) is considered to be more anomalous.

As an illustrative example, let us consider an ensemble of 6 models that detected the events in Table 1. Then τ_1 is considered the most anomalous event, followed by τ_3 . Since both τ_2 and τ_4 were detected by the same number of models, we have to check their activity score so that the event associated with a larger subgraph is considered more anomalous. Therefore, τ_4 is considered more anomalous than τ_2 .

Specification For each event detected, WINTENDED provides the factor vectors associated with the event, as the one shown in Fig. 4, thus allowing the identification of the nodes contributing to the event and the event instant (in this case, the anomaly is caused by employee 35 at the seventy-seventh day).

5.1 Notes on complexity

Applying WINTENDED to a given time window of a network requires the decomposition of the time window and the posterior automatic analysis of each of the R factors produced.

According to Hansen et al. (2015), the cost per iteration of the algorithm used to fit a CP-APR model, is approximately proportional to the number of non-zero entries in the tensor. In this context, since we target networks, the number of non-zeros corresponds to the number of edges in the network timestamps. Therefore, CP-APR is efficient when dealing with sparse networks, in which the number of edges is substantially smaller than the number of nodes (Ching et al., 2015). Also, by processing the network in a sliding window mode, we substantially reduce the amount of data to be processed at each time (on the contrary to the traditional tensor decomposition-based approaches Kolda and Sun 2008; Koutra et al., 2012).

Based on this, since the number of non-zero entries in the tensor is usually much larger than the number of components R , then the computational cost of the proposed approach is dominated by the tensor decomposition calculation. In more detail, the cost of processing a given time window in WINTENDED is proportional to the number of edges in the time window. It should be noted that if an edge connecting nodes a and b is repeated in multiple timestamps then each occurrence must be accounted as a distinct edge.

Algorithm 1: Window processing of WINTENDED

Input : network window $\mathcal{W} \in \mathbb{R}^{N \times N \times L}$, number of decomposition components R
Output: event instants (*event_log*), event scores (*event_activity*) and event patterns (*event_patterns*)

```

//Initialize event register
event_log ← {};
event_activity ← {};
event_patterns ← {};

//Decompose tensor window
{A, B, T} ← nmCP(W, R);

//Identify event candidates
candidates ← {};
instants ← [];
for r = 1 : R do
    [iscandidate, τ] ← ispeak(T(:, r));
    if iscandidate then
        candidates ← candidates ∪ {r};
        instants(r) ← τ;

//Verify candidates
for c ∈ candidates do
    //Extract the nodes associated with the anomalous subgraph G' of candidate c
    nodes1 ← {i : A(i, c) > 0} (according to (4));
    nodes2 ← {i : B(i, c) > 0} (according to (4));
    τ ← instants(c);
    //Check if the density of the anomalous subgraph G' is substantially higher at
    the anomalous instant
    isevent ← isanomaloussubgraph(nodes1, nodes2, τ);
    if isevent then
        event_log ← event_log ∪ {τ};
        event_activity ← event_activity ∪ { $\frac{|nodes_1 \cup nodes_2|}{N}$ };
        event_patterns ← event_patterns ∪ {[A(:, c), B(:, c), C(:, c)]};

```

6 Experiments

The empirical evaluation was carried out on MATLAB using Tensor Toolbox (Bader and Kolda, 2015, 2007). The code is publicly available at <https://github.com/ssfernandes/Event-Detector>.

6.1 Datasets

In this work we considered two synthetic networks (*synth* and *synthx*) and seven real-world time-evolving networks: *stockmarket* (Costa, 2018), *challengenet* (Rayana and Akoglu, 2014), *enron* (Priebe et al., 2005), *manufacturing* (Michalski et al. 2011), *reality* (Eagle and Pentland, 2006), *biketrips*¹ and *zagrebtraffic* (Carić and Fosin, 2020). In particular, *stockmarket* maps the correlation between assets in the stock market at each semester. *challengenet* is a computer communication network. *enron* and *manufacturing* are email exchange networks whose edges are weighted

¹ <https://www.capitalbikeshare.com/system-data>

by the number of emails between two people. `reality` is a proximity network derived from bluetooth devices logs. `biketrips` is network of bike trips between stations in Washington so that an edge between two stations has a weight equivalent to the number of bike trips between those stations. Finally, `zagrebtraffic` is part of the road network of Zagreb whose edges are weighted by the relative average speed between the corresponding road intersections at a given time of working days, with summer months (July and August) excluded from the data (for a more detailed description of this network, the reader may refer to Carić and Fosin (2020)). All these networks are summarized in Table 2. As a pre-processing step, networks with large weights (all but `stockmarket`, `biketrips` and `zagrebtraffic`), were subjected to a logarithmic scaling on the edge weights, similarly to Berry and Browne (2005).

The first four networks (`synth`, `synthx`, `stockmarket` and `challengenet`) were used as validation networks, while the remaining were analysed as case studies.

With respect to the validation networks, the synthetic datasets were generated so that they simulated a network. To achieve our aim, we decomposed a sampled subnetwork of `dblp` (Desmier et al., 2012) using CP with 3 components and combined the node factor matrices with an artificial temporal factor matrix (modelling a periodical cosine, a linear trend and a white noise vector with 60 elements each). Then three local anomalies were injected into the network by replacing three subgraphs with less than 10 nodes with a dense subgraph extracted from a different network (`InfectiousPatterns` Isella et al., 2011). `synth` and `synthx` differ on the nodes and size of the sampled subnetwork.

In `stockmarket`, according to the analysis provided in Costa (2018), which is also supported by the known economic situation, there are two major events (at timestamps 24 and 30).

`challengenet` was characterized by three events: abrupt node degree increase at instants 376, 377 and 1126. Nonetheless, in order to enrich this network, we injected anomalies by increasing at least 10× the degree of one node at timestamps 500 and 612 and by injecting a clique subgraph at timestamp 1053.

It should be noted that the local events injected into the networks were of three different types, described in Table 3. In `synth` and `synthx` the events consisted of nodes interacting with a subset of nodes with which they did not interact in the remaining instants. In `challengenet`, we artificially increased the degree of a given node, not necessarily very active, and we injected a clique subgraph within a subset of interacting nodes. Moreover, the scale of the events in each of the validation networks are listed in Table 4.

6.2 Design of experiments

We processed each network using a sliding window with no overlap and applied the detection procedure to each time window. In our ensemble, we considered window lengths according to the time granularity of the datasets (presented in Table 2). In particular, we used: 5, 10 and 15 timestamps for `synth`, `synthx` and `reality`; 8, 10 and 12 timestamps for `stockmarket`; 9, 18 and 36 timestamps for `challengenet`; 8, 12 and 16 weeks in `enron`; 7, 14 and 21 timestamps for `manufacturing`; 12, 24 and 168 timestamps for `bike` and 4, 6, 8 timestamps for `zagrebtraffic`. Regarding the number of components, we used 15, 25, 35, 50 and 75 in all datasets. Finally, according to our preliminary experiments, the method was able to spot the events, even when we set a small value for γ (in (6)) and, consequently we considered $\gamma = 0.05$ in all of the experiments.

6.3 Baselines

Since our aim is not only to improve over the existing tensor decomposition-based approaches but also over the state-of-the-art detectors, we considered two baselines: the tensor decomposition reconstruction error (TDRE) (Koutra et al., 2012) and the recent ensemble approaches proposed by Rayana and Akoglu (SELECTH and SELECTV) (Rayana & Akoglu, 2016).

Regarding TDRE, in order to make a fair comparison, we also considered an ensemble of models with different number of components. The timestamps were ranked based on the reconstruction error and the ranking results of the multiple models were averaged.

In SELECT variants, for each ensemble we considered the time-series of one of three node features: weighted degree (w), unweighted degree (uw) and number of triangles in the node egonet (t). Thus, the method SELECT[X]_[f] refers to the application of a SELECT, with ensemble approach X ($X \in \{H, V\}$), using feature f , for $f \in \{w, uw, t\}$.

6.4 Evaluation metrics

In the evaluation process it should be taken into account that WINTENDED does not assign an abnormality score to all instants, contrary to the baselines. In fact, it only assigns an abnormality score to the instants flagged as events by the method. Therefore, as evaluation metric, we considered the top- k precision. This metric consists of the rate of true events within the top- k anomaly scores, where k is the number of true events.

Nonetheless, it is important to account for how far the method is from making an accurate detection, whenever possible. For example, let us assume that (i) there are 3 true events, τ_1 , τ_2 and τ_3 and (ii) the list of instants sorted by abnormality score (in descending order - from the most anomalous to the least) is $[\tau_1, \tau_2, \square, \tau_3 \dots]$ by method a and $[\tau_1, \tau_2, \square, \square, \square, \tau_3 \dots]$ by method b , with \square representing an instant of time different from τ_1 , τ_2 and τ_3 . Then, both methods have the same top-3 precision ($\approx 66,7\%$), but method a is closer to an accurate detection. Note that in the sorted scores list of method a , τ_3 is the instant with the fourth highest abnormality score while in method b , the same instant was associated with the sixth highest abnormality score. Based on this, we considered an adaptation of the Levenshtein edit distance (Levenshtein, 1966) to quantify the deviation of the abnormality rankings provided by the methods under study. The idea consists of counting the number of operations needed to arrive to the correct ranking (that is, to a top- k precision of 1).

6.5 Results

We carried out our experimental evaluation in order to address the following questions:

- Q1: Is WINTENDED accurate?
- Q2: How irregular are the events flagged by WINTENDED?
- Q3: Does WINTENDED have potential to find other types of events?

The results were as follows.

RQ1: Is WINTENDED accurate?

For the sake of simplicity, we start by analysing the results in each network separately and then present the general observations. The accuracy results on the validation datasets are presented in Table 5.

Synth As depicted in Table 5, our approach was able to spot the three injected local anomalies, while TDRE was able to spot one event and SELECT variants failed at detecting all the known events.

We further analysed the top-3 ranked events flagged by the baselines and verified that two of such instants, 9 and 12, corresponded to the interaction peaks of the network (and therefore, may be regarded as global events, which resulted from the usage of a white noise factor). Our method also flagged instants 9 and 12 as anomalous. However, a lower anomaly score was assigned to them, being the 5th and 6th ranked instants, respectively.

Moreover we measured the Levenshtein edit distance (see Table 6), and observed that TDRE was the state-of-the-art approach whose estimates were closer to a correct ranking. On the other hand, SELECT variants, specially SELECTH variants, were far from an accurate prediction.

Synth X In this network, our method failed at detecting one of the events, but still, it outperformed the baselines. The performances of TDRE and SELECT were similar to the previous. In particular, SELECT variants spotted interaction peaks which resulted from the usage of the white noise factor in the dataset generation.

Moreover, we measured the deviation level using the Levenshtein distance (Table 6) and verified that SELECTV variants were in general, the methods exhibiting the less erroneous rankings, followed by TDRE. SELECTH variants were far from identifying the true events. In particular, we further investigated the abnormality rankings provided by SELECTH and observed that the true event associated with the highest anomaly score was at least 26th in the abnormality ranking. Finally, we note that tensor decomposition was not able to find patterns associated with one of the true events and, consequently, WINTENDED did not assigned an abnormality score to such instant. Because of this, the WINTENDED deviation was not computed in this network.

Stock Market Both WINTENDED and SELECT variants were able to detect the events in `stockmarket`, which corresponded to global events, involving at least 75% of the nodes. Still, TDRE failed at detecting them: semesters 24 and 30 were the 22th and 27th positions in the abnormality rank. We believe that this is due to the inability of tensor decomposition to model local dynamics when processing a large tensor, which in this case corresponds to the set of all timestamps in the network. Such issue does not hold when applying WINDTENDED because of its sliding window processing that allows the capturing of the local dynamics.

Challenge Network We observed that WINTENDED was the method exhibiting the highest precision, while TDRE was the least accurate method. In particular, by taking a closer look at the results (Table 7), we observed that SELECT variants failed at detecting the local events. Moreover, it is important to note that the event only flagged by our approach (at instant $\tau_3 = 500$) was very local, resulting from increasing the degree of a (not very active) node substantially. In other words, despite increasing the node degree considerably, this increase remained unnoticed to the baseline detectors because the node was originally not very active. Additionally, we also observed that the amount of deviation was similar for all baseline approaches but TDRE, that was extremely far from spotting one of the true events.

How irregular are the events flagged by WINTENDED?

Since there is few availability of real-world networks with known events, we additionally considered four real-world networks as case studies.

It is noteworthy that the number of events found in these networks was large, suggesting that isolated communication patterns between a considerable small number of individuals are regular and should not be considered as abnormal. Therefore, we discarded the events with an activity score (in other words, percentage of nodes in the subgraph) lower than the median score over all events. By carrying out this cleaning step we are guaranteeing that the events flagged are more likely relevant and do not correspond to a regular but very local pattern.

For each of these networks, we analysed the top-12 events flagged by the event detectors considered in this study, with the goal of assessing how the corresponding subgraphs density (at the instant flagged) deviated from the remaining instants.

Case study 1: Enron The top anomalous weeks flagged by WINTENDED were weeks 39, 84, 90, 104, 107, 120, 125, 126, 127, 129, 144 and 145. We further analysed each of such weeks, along with the associated subgraphs and were able to:

- verify the abnormality of the flagged subgraphs in its corresponding week. In particular, we verified that for all of such weeks, the number of edges of the anomalous subgraphs was at least $7\times$ larger at the anomalous week than in the remaining (reaching, in most of the cases, a density greater than $50\times$ the average);
- spot the topic of the conversation, given the emails content availability². For example, we verified that:
 - week 84 was the interaction peak within the subgraph formed by Richard Shapiro (Vice President), James Steffes (Vice President), Jeff Dasovich (Government Executive) and Steven Kean (Vice President) - the individuals flagged in the anomaly - and the discussion was on the preparation of a document on the Gas issues;
 - the event at week 104 was caused by Jonh Lavorato (CEO), that sent an email to almost all employees scheduling a meeting.

These two cases illustrate, respectively, a local and a global event.

From the events detected by WINTENDED, TDRE spotted instants 107 and 145 (which are global events, involving all employees) and SELECT variants flagged, in general, seven instants in common with our approach (also corresponding to events involving the majority of the employees). We analysed the other events flagged by these methods but we were not able to spot considerable irregularities in terms of node activity peaks.

Case Study 2: Manufacturing We analysed the top 12 ranked instants and verified that all of them corresponded to the interaction peak of a given (flagged) node and the other nodes flagged in the event, similarly to example in Fig. 4. We note that, as illustrated in Sect. 1, this type of events is “expected” in an email network and, in this case, results from the sending of an email to a broad set of employees in the manufacturing company.

We further analysed the subgraphs flagged as anomalous by our method and observed that their density was at least $20\times$ larger at the anomalous instants than the average across all instants - a behaviour which attests the abnormality of the events found.

We also applied the baselines to this network and observed that SELECTH and SELECTV variants provided a top-12 set of anomalous instants similar to the one obtained with WINTENDED. We further investigated the few instants of time flagged only by the

² <http://www.enron-mail.com/email/>

SELECT variants and observed that they revealed global anomalies (which were also detectable when tracking the density of the network over time).

Finally, it is noteworthy that the TDRE top-12 ranked events totally differed from the ones provided WINTENDED and the other baselines. Once more, we believe TDRE was not able to capture local (but regular) dynamics, thus flagging such periods as anomalous.

Case Study 3: Reality mining According to our analysis, all of the instants flagged by WINTENDED corresponded to an interaction level at least 4× stronger in the anomalous instant than the average, between the flagged individuals. Around half of these events were also flagged by SELECT variants and only two were also flagged by TDRE. From the remaining instants flagged by SELECTV, we observed that there was one instant that corresponded to an interaction peak that was not flagged by WINTENDED; however, no anomaly evidence was found for the others.

Besides the 2 instants flagged in common with WINTENDED, the top ranked instants by TDRE were mostly associated with extremely low network activity instants, which were neglected, leading to a high reconstruction error.

Case Study 4: Bike trips From analysing the top-12 abnormal instants provided by WINTENDED, we verified that all of them corresponded to a densification peak of at least 7× the average density (of the anomalous subgraph). However, it should be noted that, on the contrary to the previous networks, the hourly `bike` network is expected to be periodic with density peaks in the rush hours. Therefore, an event detector should only flag a rush hour if it deviates from the remaining rush hours, which was the case of WINTENDED: most of the instants flagged by WINTENDED occurred at 9 a.m in weekdays (corresponding to rush hours) but the traffic level at the instants flagged as abnormal was even higher than the usually observed in the same hour of a different weekday. In Fig. 6 we provide an example of a pattern modelling one of the anomalies found: at 9 a.m. of September, 19th there was more traffic than usual between the stations flagged in the nodes factor vectors. This event is confirmed by the number of trips occurring between the stations flagged (see Fig. 7): the number of trips between the stations is periodic but, still, it is substantially higher at 9 a.m. of September, 19th.

SELECTH and SELECTV variants flagged around 3 and 6 events in common with WINTENDED, respectively. No strong anomaly evidence was found for the remaining instants of top-12 SELECT variants ranking. TDRE top instants differed from all of the other approaches and were mainly few active instants of the network (similarly to the observed in the `reality` case study).

Q3: Does WINTENDED have potential to find other types of events?

While our method was designed to spot densification events, we also wanted to study how we could adapt the method to detect other types of events.

For example, in a traffic network, in which the nodes represent intersections and the edges represent road segments, there are no densification events (because the structure of the network is always the same) but the volume and speed of the traffic on those roads changes over the day and is indicative of rush hours, accidents, ...In such scenario, an event is characterized by an abrupt change in the weights of the edges.

With this type of event in mind, we removed the verification step of WINTENDED (step 3 in Fig. 5) and we additionally incorporated a rule similar to (5) but for low extreme outliers (Dawson 2011) so that the resulting method is not restricted to detecting high peaks. We applied the resulting variant to `zagrebtraffic` network, which is characterized by three anomalous events: significant vehicle speed decrease in two rush hours (due to congestions caused by large number of daily commuters) and increase in vehicles speeds during night time interval.

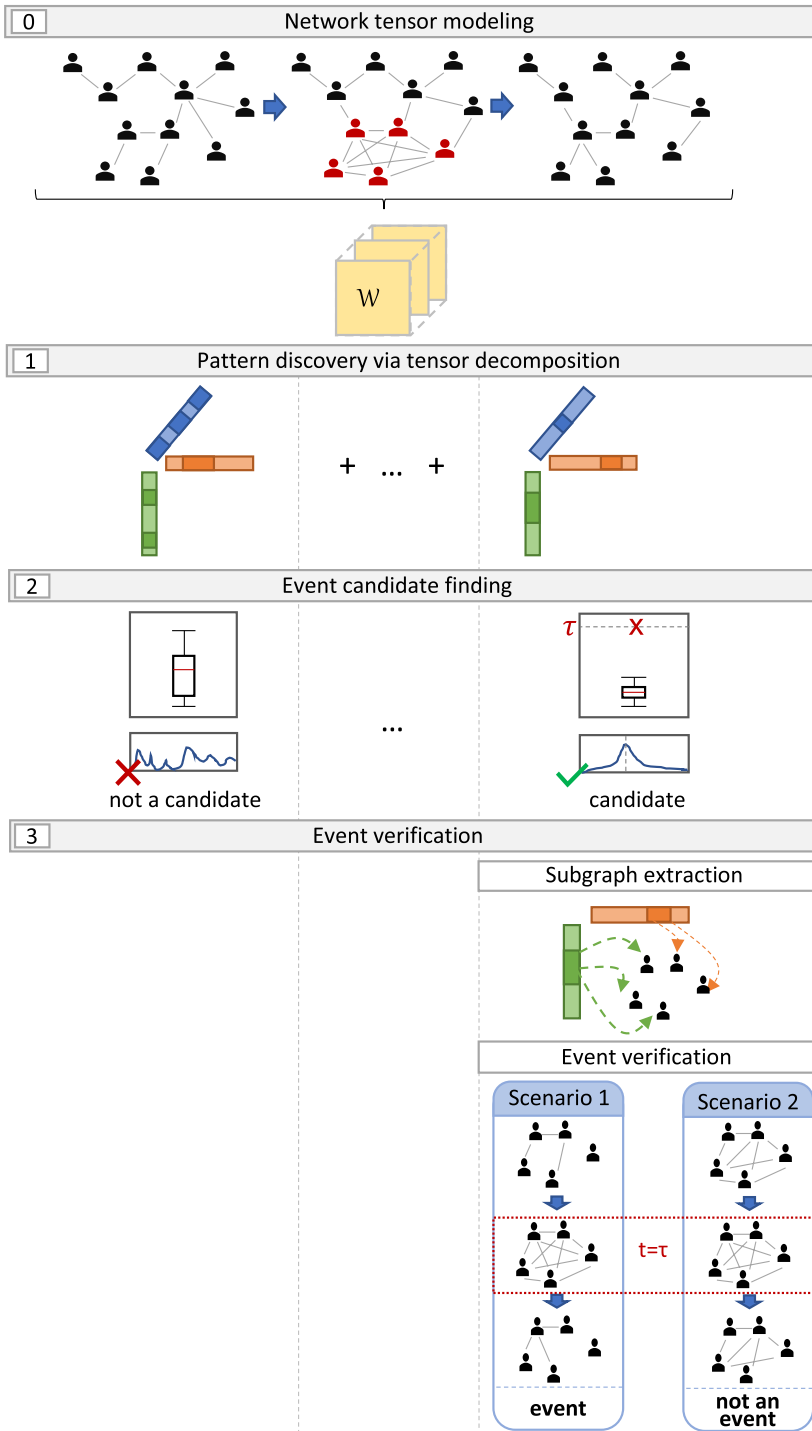


Fig. 5 Illustration of the WINTENDED processing of a network window with 3 timestamps

Table 2 Datasets Summary

Dataset	Type	Nodes	Time granularity	Size
synth	Synthetic network	–	–	500 × 500 × 60
synthx	Synthetic network	–	–	1000 × 1000 × 60
stockmarket	Stock market network	assets	6 months	30 × 30 × 42
challengenet	Computer communication network	hosts	10 minutes	125 × 125 × 1304
enron	E-mail exchange network	employees	week	184 × 184 × 44
manufacturing	E-mail exchange network	employees	day	167 × 167 × 272
reality	Proximity network	mobile phones	working day	94 × 94 × 175
bike	Bike traffic network	stations	hour	573 × 573 × 720
traffic	Traffic road network	road junctions	time intervals of a day	34 × 34 × 8

Table 3 Types of local events injected in the networks

Dataset	Injected local events	Number of events
synth/synthx	Instant substantial interaction between nodes which do not interact often	3
challengenet	Instant substantial node degree increase	2
	Instant substantial subnetwork densification	1

Table 4 Number of event types in the validation networks

Dataset	Global Events	Local events
synth	0	3
synthx	0	3
stockmarket	2	0
challengenet	3	3

Table 5 Top-*k* precision in the validation networks

	synth	synthx	stockmarket	challengenet
TDRE	0,33	0,33	0,00	0,17
SELECTH _w	0,00	0,00	1,00	0,67
SELECTH _{ww}	0,00	0,00	1,00	0,67
SELECTH _l	0,00	0,00	1,00	0,67
SELECTV _w	0,00	0,00	1,00	0,67
SELECTV _{ww}	0,00	0,00	1,00	0,83
SELECTV _l	0,00	0,00	1,00	0,83
WINTENDED	1,00	0,66	1,00	1,00

Best performance in bold

Table 6 Level of deviation from a correct detection according to the Levenshtein edit distance in the validation networks

	synth	synthx	stock-market	challengenet
TDRE	15	19	25	164
SELECTH _w	38	48	0	10
SELECTH _{tw}	51	29	0	6
SELECTH _l	51	29	0	6
SELECTV _w	32	5	0	7
SELECTV _{tw}	33	5	0	4
SELECTV _l	33	5	0	4
WINTENDED	0	\	0	0

Table 7 Top-k precision per event type in challengenet

Method	Global Events	Local Events
TDRE	1/3	0/3
SELECTH _w	3/3	0/3
SELECTH _{tw}	3/3	1/3
SELECTH _l	3/3	1/3
SELECTV _w	3/3	1/3
SELECTV _{tw}	3/3	2/3
SELECTV _l	3/3	2/3
WINTENDED	3/3	3/3

Best performance in bold

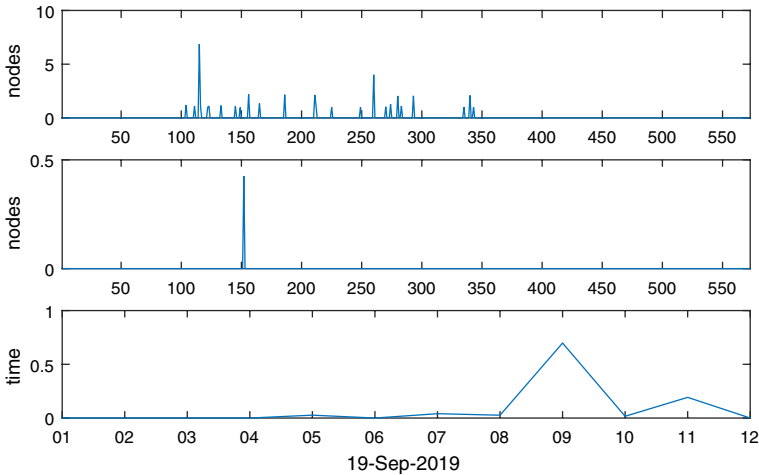
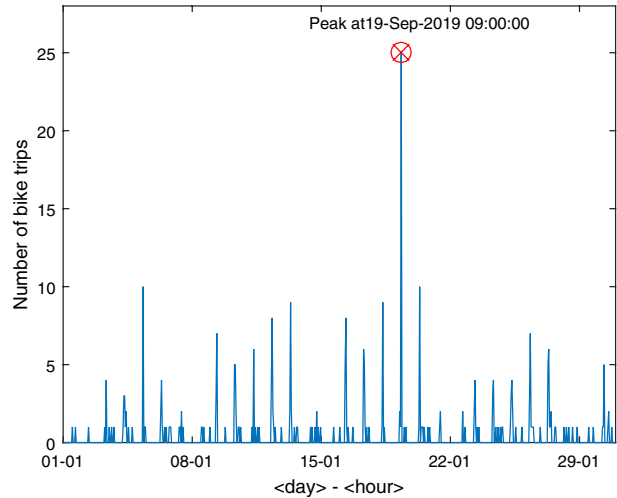


Fig. 6 Event (9 a.m. 19th September) flagged by WINTENDED in bike traffic network

We observed that our method flagged the periods mentioned above as abnormal. In particular, it flagged the morning and evening rush periods as low speed peaks and the night time as high speed peak.

Fig. 7 Number of bike trips between the stations flagged in the event of Fig. 6



Once more, we also applied the baselines to this data and verified that all of them flagged the evening rush hour as a top-3 event. The other instants in top-3 anomalous rank were periods of time immediately prior or posterior to the rush hours, in which the traffic flowed normally, and therefore did not represent an anomaly.

Based on this, the output of the modified version of WINTENDED was more meaningful than the others.

General observations In this study, we observed that:

- TDRE performance was generally poor: not only it failed at spotting known events (as in *stockmarket*), but it also provided substantially different abnormality rankings (as in *manufacturing* case study). We believe that if the networks under study exhibited a more regular evolution over time (with no local dynamics), then TDRE would be able to spot the events. However, in such scenario most of the existing detectors would also perform well.
- The estimates made by the variants of SELECT considerably differed on whether using SELECTH and SELECTV ensembles, with SELECTV exhibiting more accurate results. In this context, SELECTV variants were mostly successful in spotting global events (as in *stockmarket*), but failed at detecting some of the local events (as occurred in the synthetic networks). We believe that its inability to detect some local events arises from the fact that this type of events may not affect the node features considered by the method. In the specific case of the synthetic networks, an event caused by a subset of nodes interacting with nodes they usually do not interact may not affect the node degree nor number of triangles, thus remaining undetectable by these approaches.
- WINTENDED was able to detect not only global events (exhibiting as good or better performance than the baselines in *stockmarket*), but also local events, which are more difficult to spot (exhibiting the best performance in the synthetic networks).

Moreover, we observed that WINTENDED has the potential to find other types of events in different networks.

7 Conclusions

In this work, we propose a new perspective regarding the application of tensor decomposition to event detection in time-evolving networks - the WINDowed TENSOR decomposition for Densification Event Detection (WINTEDD). In particular, we resort to a sliding window processing in order to capture local dynamics of the networks and use statistical tools to automatically identify network global and local density peaks from the window tensor decomposition result. Moreover, our method provides the pattern associated to each flagged instant, thus allowing for the characterization of the events.

According to our empirical study, our method spots irregular density peaks even if they occur between a small subset of nodes, on the contrary to its competitors that fail at identifying local abnormalities.

In order to make our method a more versatile and powerful tool, as future direction, it would be interesting to make WINTEDD robust to detect different types of events simultaneously, independently of the network type.

Acknowledgements This work was financed by National Funds through the Portuguese funding agency, FCT - Fundação para a Ciência e a Tecnologia within project: UIDB/50014/2020 and by the European Regional Development Fund under the grant KK.01.1.1.01.0009 (DATACROSS). Sofia Fernandes also acknowledges the support of FCT via the PhD grant PD/BD/114189/2016. The traffic data used for this research was collected during project SORDITO, European Regional Development Fund under contract RC.2.2.08-0022. Finally, the authors would like to acknowledge the SocioPatterns collaboration for making the dataset (in Isella et al. (2011)) available.

References

- Akoglu, L., & Faloutsos, C., (2010). Event detection in time series of mobile communication graphs. In: Army Science Conference, pp. 77–79
- Akoglu, L., Tong, H., & Koutra, D. (2015). Graph based anomaly detection and description: A survey. *Data mining and knowledge discovery*, 29(3), 626–688.
- Bader, B. W., & Kolda, T. G. (2007). Efficient MATLAB computations with sparse and factored tensors. *SIAM Journal on Scientific Computing*, 30(1), 205–231. <https://doi.org/10.1137/060676489>.
- Bader, B.W., & Kolda, T.G., et al. (2015). Matlab tensor toolbox version 2.6. Available online . <http://www.sandia.gov/~tgkolda/TensorToolbox/>
- Berlingerio, M., Koutra, D., Eliassi-Rad, T., & Faloutsos, C., (2013). Network similarity via multiple social theories. In: Advances in Social Networks Analysis and Mining (ASONAM), 2013 IEEE/ACM International Conference on, pp. 1439–1440. IEEE
- Berry, M. W., & Browne, M. (2005). Understanding search engines: Mahemathical modeling and text retrieval. Siam
- Carić, T., & Fosin, J. (2020). Using congestion zones for solving the time dependent vehicle routing problem. *Promet - Traffic&Transportation*, 32(1), 25–38. <https://doi.org/10.7307/ptt.v32i1.3296>.
- Chi, E. C., & Kolda, T. G. (2012). On tensors, sparsity, and nonnegative factorizations. *SIAM Journal on Matrix Analysis and Applications*, 33(4), 1272–1299.
- Ching, A., Edunov, S., Kabiljo, M., Logothetis, D., & Muthukrishnan, S. (2015). One trillion edges: Graph processing at facebook-scale. *Proceedings of the VLDB Endowment*, 8(12), 1804–1815.
- Costa, P., (2018). Online network analysis of stock markets. Master's thesis, University of Porto
- Dawson, R. (2011). How significant is a boxplot outlier? *Journal of Statistics Education*, 19(2). <https://doi.org/10.1080/10691898.2011.11889610>
- Desmier, E., Plantevit, M., Robardet, C., & Boulicaut, J.F., (2012). Cohesive co-evolution patterns in dynamic attributed graphs. In: International Conference on Discovery Science, pp. 110–124. Springer
- Dunlavy, D. M., Kolda, T. G., & Acar, E. (2011). Temporal link prediction using matrix and tensor factorizations. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 5(2), 1–27.

- Eagle, N., & Pentland, A. S. (2006). Reality mining: Sensing complex social systems. *Personal and ubiquitous computing*, 10(4), 255–268.
- Fanaee-T, H., & Gama, J. (2016). Event detection from traffic tensors: A hybrid model. *Neurocomputing*, 203, 22–33.
- Fernandes, S., Fanaee-T, H., & Gama, J. (2019). Evolving social networks analysis via tensor decompositions: From global event detection towards local pattern discovery and specification. In: *International Conference on Discovery Science*, pp. 385–395. Springer
- Hansen, S., Plantenga, T., & Kolda, T. G. (2015). Newton-based optimization for kullback-leibler non-negative tensor factorizations. *Optimization Methods and Software*, 30(5), 1002–1029.
- Hosseinimotlagh, S., & Papalexakis, E.E. (2018). Unsupervised content-based identification of fake news articles with tensor decomposition ensembles. In: *Proceedings of the Workshop on Misinformation and Misbehavior Mining on the Web (MIS2)*
- Isella, L., Stehlé, J., Barrat, A., Cattuto, C., Pinton, J.F., & Van den Broeck, W., (2011). What's in a crowd? analysis of face-to-face behavioral networks. *Journal of theoretical biology* 271(1), 166–180 <http://www.sociopatterns.org/datasets/infectious-sociopatterns-dynamic-contact-networks/>
- Jeon, B., Jeon, I., Sael, L., & Kang, U., (2016). Scout: Scalable coupled matrix-tensor factorization-algorithm and discoveries. In: *2016 IEEE 32nd International Conference on Data Engineering (ICDE)*, pp. 811–822. IEEE
- Kapsabelis, K. M., Dickinson, P. J., & Dogancay, K. (2007). Investigation of graph edit distance cost functions for detection of network anomalies. *ANZIAM Journal*, 48, 436–449.
- Kolda, T. G., & Bader, B. W. (2009). Tensor decompositions and applications. *SIAM review*, 51(3), 455–500.
- Kolda, T.G., & Sun, J., (2008). Scalable tensor decompositions for multi-aspect data mining. In: *2008 Eighth IEEE International Conference on Data Mining*, pp. 363–372
- Koutra, D., Papalexakis, E.E., & Faloutsos, C. (2012). Tensorsplat: Spotting latent anomalies in time. In: *Proceedings of the 2012 16th Panhellenic Conference on Informatics, PCI '12*, pp. 144–149. IEEE Computer Society
- Levenshtein, V. I. (1966). Binary codes capable of correcting deletions, insertions, and reversals. *Soviet physics doklady*, 10, 707–710.
- Michalski, R., Palus, S., & Kazienko, P. (2011). Matching organizational structure and social network extracted from email communication. In: *Lecture Notes in Business Information Processing*, vol. 87, pp. 197–206. Springer Berlin Heidelberg
- Papadimitriou, S., Sun, J., & Faloutsos, C. (2005). Streaming pattern discovery in multiple time-series
- Papalexakis, E., Pelechrinis, K., & Faloutsos, C. (2014). Spotting misbehaviors in location-based social networks using tensors. In: *Proceedings of the companion publication of the 23rd international conference on World wide web companion*, pp. 551–552. International World Wide Web Conferences Steering Committee
- Papalexakis, E.E., (2016). Automatic unsupervised tensor mining with quality assessment. In: *Proceedings of the 2016 SIAM International Conference on Data Mining*, pp. 711–719. SIAM
- Papalexakis, E.E., Faloutsos, C., & Sidiropoulos, N.D., (2012). Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2012, Bristol, UK, September 24–28, 2012. Proceedings, Part I, chap. ParCube: Sparse Parallelizable Tensor Decompositions, pp. 521–536. Springer Berlin Heidelberg, Berlin, Heidelberg
- Priebe, C. E., Conroy, J. M., Marchette, D. J., & Park, Y. (2005). Scan statistics on enron graphs. *Computational & Mathematical Organization Theory*, 11(3), 229–247.
- Ranshous, S., Shen, S., Koutra, D., Harenberg, S., Faloutsos, C., & Samatova, N. F. (2015). Anomaly detection in dynamic networks: a survey. *Wiley Interdisciplinary Reviews: Computational Statistics*, 7(3), 223–247.
- Rayana, S., & Akoglu, L., (2014). An ensemble approach for event detection and characterization in dynamic graphs. In: *ACM SIGKDD ODD Workshop*
- Rayana, S., & Akoglu, L. (2016). Less is more: Building selective anomaly ensembles. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 10(4), 42.
- Shin, K., Hooi, B., & Faloutsos, C. (2016). M-zoom: Fast dense-block detection in tensors with quality guarantees. In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 264–280. Springer
- Shin, K., Hooi, B., Kim, J., & Faloutsos, C., (2017). D-cube: Dense-block detection in terabyte-scale tensors. In: *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, pp. 681–689. ACM
- Shoubridge, P., Kraetzl, M., WALLIS, W., & Bunke, H. (2002). Detection of abnormal change in a time series of graphs. *Journal of Interconnection Networks*, 3(01n02), 85–101.

da Silva Fernandes, S., Tork, H.F., & da Gama, J.M.P., (2017). The initialization and parameter setting problem in tensor decomposition-based link prediction. In: 2017 IEEE International Conference on Data Science and Advanced Analytics (DSAA), pp. 99–108. IEEE

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.