Computer Science and Information Systems 19(2):1023–1046 https://doi.org/10.2298/CSIS211215012O

Dynamic Network Modelling with Similarity based Aggregation Algorithm ⋆

Günce Keziban Orman

Computer Engineering Department, Galatasaray University,
İstanbul, TURKEY
korman@gsu.edu.tr

Abstract. Proper modelling of complex systems allows hidden knowledge discovery that cannot be explored using traditional methods. One of the techniques for such modelling is dynamic networks. In this work, we aim to develop a methodology for extracting proper dynamic networks. We concentrate on two fundamentally interconnected problems: first, determining the appropriate window size for dynamic network snapshots; and second, obtaining a proper dynamic network model. For the former problem, we propose Jaccard similarity and its statistical significance based compression ratio, and for the latter, we propose an aggregation approach that extracts dynamic networks with snapshots of varying duration. The aggregation algorithm compresses the system information when there is repetition and takes snapshots when there is a significant structural change. The experiments are realised on four simple or complex data sets by comparing our proposal with baseline approaches. We used well-known Enron emails as simple set and Haggle Infocomm, MIT Reality Mining, and Sabanci Wi-Fi logs as complex data sets. These complex sets like Wi-Fi or Bluetooth connections which are known to be noisy, making analysis difficult show the proximity of system objects. The experimental results show that the proposed methodology can be used to find not only significant time points in simple Enron emails, but also circadian rhythms with their time intervals that reveal the life-cycle of connected areas from complex Wi-Fi logs or bluetooth connections. According to testing on four real-world data sets, both compression ratios and the aggregation process enable the extraction of dynamic networks with reduced noise, are easy to comprehend, and appropriately reflect the characteristics of the system.

Keywords: dynamic networks, data compression, proper time intervals, algorithm.

1. Introduction

Dynamic network modelling of complex systems enables us to uncover previously unknown and emerging properties of the studied field [19], [27], [3], [14], [1],[24]. This model can be defined in different ways, such as link streams, event-based sequential graphs, or time flow-based sequential graphs. In [5], the authors evaluated the effectiveness of the different models by a data compression technique they proposed. The results demonstrated that different model types can be appropriate for different data sets. In this work, we concentrate on dynamic networks under the form of historically ordered

⋆ This is an extended version of INISTA 2021 conference paper ”Aggregating Time Windows for Dynamic Network Extraction”.

graph sequences. Each member of this sequence is called a snapshot, and it represents the interaction of the system objects for a given time interval.

Usually, a fixed duration, window size w, is used for determining the time interval. The topology of each snapshot depends on w. Since the structure of the model will affect all further analysis and temporal dynamics, i.e. community detection [21,8], link prediction, attribute prediction, change point detection [13], epidemic spreads, possibilities of communication and cascade of influence [20], it is crucial to find an optimal w for a proper dynamic network extraction. Yannick et al. underline that the larger the w, the higher the information loss related to temporal dynamics [20]. Previous works define an optimal w as being small enough to cover the temporal dynamics of the system but large enough to eliminate noise in the final model [12,26,29]. In this work, we concentrate on two inter-related problems; first, finding optimal w and second, extracting the proper dynamic network.

In most of the studies done so far, these two problems have been addressed together. Traditional methods consist of iterative and experimental procedures [29,30,18,12,16,13,7]. First, a set of candidates w is determined according to the domain knowledge and expertise of the authors. Second, candidate dynamic networks are extracted for each value of w. Third, the quality of candidate networks is measured for choosing the most appropriate one. Mostly, a dynamic network is represented in the form of its features' time series for measuring its quality. These methods differentiate from each other by first *choosing representative features* and, second, *measuring the quality of the time series*.

The most commonly used features for time series generation are network topological properties such as diameter, average distance, etc. [29]. We come across different time series analysis techniques for their quality measurement, such as discrete Fourier transformation [11], ARIMA [30] or using statistical metrics as variance or standard deviation [29]. Among them, the TWIN approach can be accepted as the baseline of traditional approaches [29]. It relies on basic optimisation of both the noise and variation level of the feature time series generated for different w. However, TWIN suffers from using non-scale invariant topological properties, i.e. diameter, in time series generation. This can be misleading when comparing different dynamic networks extracted for different candidates. A detailed discussion of this issue is done in our previous work, [22].

Recent studies have focused on using a reliable similarity metric to assess the stability of the snapshot sequence [6,22,16]. Darst et al. extract the snapshots for the different time intervals [16]. Time intervals between snapshots are determined by Jaccard similarity optimisation of consecutive events. The similarity of events is measured using an increasing time interval. By linear optimisation of similarity, it is decided which w is the most proper. Snapshots in the final network can have different w. There are more snapshots when the system is very active and a few snapshots when the system events are calm. Chiappori et al. propose a Jaccard similarity-based stability metric and a fidelity score [6]. They show that the extracted dynamic networks are unstable if filtering is not applied independently of w. They propose a parametric filtering procedure for removing less prominent links that are assigned as noise. Although the Jaccard similarity, as a scale-invariant metric, is a more objective measure than topological properties in comparing different w, the methods used in determining the stability of systems or change point detection in the previous two studies are open to criticism because they do not use a reference value for comparing Jaccard values.

We have previously proposed using Jaccard similarity based scale invariant similarity metrics instead of topological properties in the TWIN for finding proper time intervals and demonstrated its effectiveness in [22,31]. In these works, the statistical significance limit of Jaccard is used as a reference point when a comparison is needed. Hence, a more objective selection is made at the best w determination. We have also proposed a procedure for determining a different duration for each snapshot extraction [31]. This procedure is similar to the one proposed in [16]. However, instead of optimisation, it determines the duration of the time interval by comparing the obtained Jaccard values with its statistical significance limit. Most of the traditional approaches use fixed durations for all snapshots [21,8,7]. However, sometimes there can be hectic periods where system members have a lot of interactions among themselves, while other times they stay calm. By considering this fact, our previously proposed procedure extracts snapshots with different durations.

This work can be seen as an extension of our previous work [31]. In the previous work, we first introduced a dynamic network extraction procedure that results in aggregated networks, second, evaluated its performance on two well-known data sets, Enron and Haggle Infocomm, and third, compared the aggregated networks with fixed-duration networks by using the TWIN. In this paper, we divide the problem into two parts: first, determining the optimal w, and second, extracting the appropriate dynamic network.

We propose three new contributions to this current work. Our first contribution is to define a new dynamic network compression ratio. It is based on Jaccard similarity and its statistical significance. The proposed compression ratio is a dynamic network global level metric. It is capable of comparing any two dynamic networks. We use it for determining the best w. Our second contribution is evaluating the effectiveness of both the new compression ratio and the previously proposed aggregation procedure on four data sets, including two additional sets; (1) MIT Reality Mining and (2) Sabanci University Wi-Fi logs. Our final contribution is comparing our proposal compression ratio with not only the TWIN but also with a new additional baseline, which is the stability metric proposed in [6] and also with the average link similarity of the consecutive snapshots of the dynamic network. These metrics are also Jaccard based. That is why it allows us to evaluate the performance of our compression ratio more clearly. Moreover, in this work, we clearly explain the behaviour of previously proposed similarity metrics by using toy examples and also present the algorithm of a previously proposed dynamic network extraction procedure.

In the following sections, the readers will find first, preliminaries with the details of baselines, i.e. TWIN [29], stability[6] and average link similarity, and second, an explication of our new compression ratio proposal with its algorithmic procedure of dynamic network extractor, respectively. Then, in section 3, we explain the experimental set up by giving the details of the studied data sets. In section 4, we first show and interpret the results of experiments for finding the best w by comparing the proposed compression ratio with baselines and, secondly, interpreting the result of the aggregation algorithm by comparing it with constant window size. Finally, in section 5, we explain the essential conclusions of this work and its future aspects.

2. Method

A complex system that evolves over time $\mathcal{C}$ is a system of interacting objects in which each interaction occurs at a time point in the continuous time interval defined between the beginning and the ending time points $[t_1, t_\theta]$. A dynamic complex network $\mathcal{G} = \langle G_1, \ldots, G_\theta \rangle$ is a sequential graph representation of $\mathcal{C}$ for discovering its emerging and non-linear dynamics. It is a consecutive set of static networks ordered historically. Each G_i $(1 \leq i \leq \theta)$ member of this set is referred to as a snapshot, where i representing a sub-interval in $[t_1, t_\theta]$. $G_i = (V_i, L_i)$ is a static and plain network in which V_i defines the set of nodes and $L_i \subseteq V_i \times V_i$ is the set of links. As in reality $\mathcal{C}$ is defined in continuous interval, we should discretize $[t_1, t_\theta]$ for extracting $\mathcal{G}$ for fitting its definition. Usually, a constant window size, w is determined for such discretization. Thus, finding the best window size w and extracting the proper $\mathcal{G}$ for the best w arise as prominent problems in this task. In the following part, we concentrate on these two problems. First, we explain the baseline methods in preliminaries. Second, we present our proposal for selecting w. Finally, we explain a simple but effective algorithm for automatically deciding time intervals for the snapshots based on previously selected w.

2.1. Preliminaries

In this part, we concentrate on two previously proposed approaches for selecting the optimal w: TWIN by [29] and Stability by [6]. Many previous approaches for determining the optimal w in the literature [29,30,4] employ TWIN as a baseline, whereas Stability is a novel approach.

TWIN is based on the combination of two time series statistics: *noise* and *compression*. For a given w and the corresponding $\mathcal{G}$, $F_w = \langle F_1, \ldots, F_t \rangle$ is a uni-variate time series of a topological property of $\mathcal{G}$. The noise of F_w is measured by its variance. Equation 1 gives the commonly accepted definition of variance where $\mu(.)$ describes the mean value of a studied uni-variate time series. Variance explains how much the F_w changes over time. The larger the variance, the noisier the signal of the studied topological property.

$$\sigma^2 = \frac{1}{t} \sum_{i}^{t} [F_i - \mu(F_w)]^2 \tag{1}$$

The compression ratio of F_w is defined as the string compression by run-length encoding. Hence, in TWIN, F_w is interpreted as a string. Run-length encoding compresses the repeating parts of strings. More clearly, if a character appears several times consecutively, run-length encoding represents it only once with the appearance count. Assume u represents the length of the string representation of F_w. Because each value in u represents a single character, the repetitive values can be compressed as if they were characters in a string. The length of the compressed string representation of F_w is defined as c. Therefore, the compression ratio can be calculated with the formula given in Equation 2. If the consecutive snapshots have the same value for the studied topological property, the compression ratio becomes large. Sulo et al. indicate that σ^2 and compression ratio, s, have opposite behaviours [29]. TWIN analyses σ^2 and s as functions of w. It determines the best w as the one with the smallest difference between σ^2 and s.

$$s = \frac{u}{c} \tag{2}$$

TWIN has three major flaws. For starters, network topological properties are not scale-invariant. As a result, using them to compare different $\mathcal{G}$ retrieved for different w can be misleading. Second, TWIN compresses numerical data using string compression. Run-length encoding checks only if consecutive members of a time series are the same or not. Although it can be used with integers, it has no application with real numbers. And yet, the majority of topological properties, such as density, average degree, and so on, have real values. As a result, only a few topological properties, such as diameter or radius, can be used in TWIN. In addition, variance is not a scale-invariant statistic. It is a popular method for determining the amount of noise in a time series. TWIN, on the other hand, uses it to compare distinct time series. When dealing with tiny changes, using a non-scale invariant metric might be misleading. In this work, we measure the noise level of time series with a normalised standard deviation over the mean when we need to compare them. Another baseline that we consider is the Jaccard Similarity-based Stability Metric proposed by [6]. Jaccard Similarity estimates the amount of shared parts between two sets in its broadest sense. The usage of Jaccard Similarity on link sets of consecutive snapshots is very common for measuring the stability of extracted networks [6,22,16]. In our study, we refer to this special form of Jaccard as *Link Similarity*, J_{link} (see Equation 3). Besides, *Node Similarity*, J_{node} can also be defined (see Equation 4) for consecutive snapshots' node sets. Both of them count the number of recurring links or nodes between two consecutive snapshots as well as the number of links or nodes that are seen at least in one of the consecutive snapshots.

$$J_{link}(t_i, t_{i+1}) = \frac{\mid L_{t_i} \bigcap L_{t_{i+1}} \mid}{\mid L_{t_i} \bigcup L_{t_{i+1}} \mid} \tag{3}$$

$$J_{node}(t_i, t_{i+1}) = \frac{\mid V_{t_i} \bigcap V_{t_{i+1}} \mid}{\mid V_{t_i} \bigcup V_{t_{i+1}} \mid} \tag{4}$$

In particular, *Stability* is defined as the weighted average of the link similarity over all consecutive snapshots. Its formula is given in Equation 5. In fact, Chiappori et al. propose a framework of two metrics; stability and fidelity [6]. However, because their fidelity measures the amount of lost information in the original data due to the snapshot extraction, we do not consider this metric. For instance, if two system actors are connected two times in a given duration, this is represented by one single link in the related snapshot by using plain networks. It can be classified as information loss based on any comparison of raw data and model. Here, using weighted links or shortening the duration of representing two connections in separate snapshots can be proposed as a solution. But it might be possible that the original data is noisy or includes redundant information depending on the applications. This is another crucial aspect to investigate, but it is out of the scope of our work. In this work, we extract snapshots in the form of plain networks. We do not concentrate on the compatibility between the extracted snapshots and raw data, but only on the snapshot's quality. That is why we use stability as another baseline, but we are not interested in fidelity.

$$I = \frac{\sum_{t_i,t_{i+1}} J_{link}(t_i,t_{i+1}).min(|L_{t_i}|,|L_{t_{i+1}}|)}{\sum_{t_i,t_{i+1}} min(|L_{t_i}|,|L_{t_{i+1}}|)} \quad (5)$$

Stability can be seen as a global metric that explains if the extracted $\mathcal{G}$ has overall stability. Because the J_{link} of consecutive snapshots is weighted based on the smallest snapshots' link amount, stability is more regulated to compared snapshots' sizes than the simple average J_{link}. However, it does not consider the effect of the number of common elements in the averaging scores. As a result, it can still be misleading when comparing different w. We consider both stability and average link similarity as baselines in our work. TWIN and stability are both useful in determining the best w which is the one resulting in the highest stability.

2.2. Proposed Method

Link and Node Compression Ratio for Window Size Selection The effectiveness of the usage of network similarity-based graph comparison metrics instead of topological properties is demonstrated in [22,31]. We describe how to choose the right w using novel compression ratios based on similarity. In the literature, there exist various distinct network similarity measures [9]. Nevertheless, because adjacency matrices are used, the majority of these metrics are dedicated to quantifying the similarity of two networks of the same size. When the number of nodes in the compared snapshots differs, which is the most common case, the union network matrices can be quite sparse. Thus, they suffer from the sparsity of the matrix. That is why we concentrate on Jaccard Similarity based metrics. Two different versions of Jaccard Similarity for measuring network stability are explained in the previous part with the equations 3 and 4. We use these two versions for the rest.

The Jaccard Similarity metric is a scale invariant metric with a range of 0.0 to 1.0. Its value is equal to 0.0 if there are no similar elements between the compared sets. In other words, the two sets are completely different. If it is equal to 1.0, it means that the two sets have exactly the same elements. If the network's similarity scores are high and there is a long-term repeat, then the network has redundant snapshots. Non-repeating signals, on the other hand, show variation in the snapshots. However, which value of Jaccard similarity indicates that two sets are sufficiently similar?

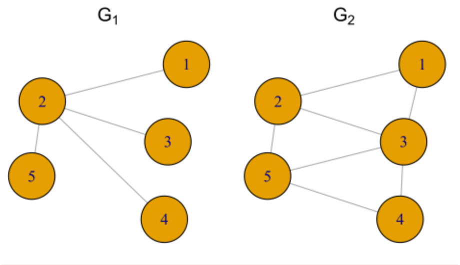

Fig. 1. Two consecutive simple snapshots

For example, the similarity scores for two consecutive snapshots in figure 1 are $J_{node} = 1.0$ and $J_{link} = 0.375$. The changes to the link structure have a direct impact on J_{link}. However, how can we tell whether these two consecutive snapshots are similar or not? We look up the statistically significant limit of Jaccard Similarity, which was previously investigated by [23]. They propose a probabilistic null-model based on the size of the compared sets. This model's score indicates the probability that two sets are similar by chance. In other words, if the Jaccard score of two sets is greater than a significant threshold, they are statistically similar. This null model (See Equation 6) assesses the randomness of two sets by taking into account both their common elements and the size of the smaller set.

$$P = \frac{\sum\limits_{x=0}^{C} \binom{A+B-x}{x}}{\sum\limits_{x=0}^{min(A,B)} \binom{A+B-x}{x}} \tag{6}$$

The numbers of elements in each set are defined as A and B, respectively, while the number of elements in common is defined as C. Because the maximum number of elements in common cannot be more than the minimum of the number of elements in A and B, P represents the probability of two consecutive sets having C elements in common by chance. In this study, A and B represent the number of node sets (or link sets) of the two network snapshots we compared. Similarly, C is the number of common elements in those sets.

If the compared snapshots are similar, their similarity must be greater than P, because being more similar by chance implies that the system's stability is still being maintained. As a result, this network segment can be compressed. The snapshots, on the other hand, are not similar if the similarity is smaller than P. We define D_w as the time series of the difference between the measured similarity and its statistical significance. Let $S_w = \langle S_1, \ldots, S_t \rangle$ denote a time series of a similarity metric, J_{link} or J_{node}, and $P_w = \langle P_1, \ldots, P_t \rangle$ denote a time series of statistical significance limits of each member in S_w. Then, the equation of D_w is given in Equation 7.

$$D_w = \langle S_1 - P_1, \ldots, S_t - P_t \rangle \tag{7}$$

Accordingly, we define a new compression metric s_{sim} as the ratio of t, which is the number of elements in uncompressed similarity time series S_w to t_{pos} the number of positive elements of D_w. Its equation is given in Equation 8.

$$s_{sim} = \frac{t}{t_{pos}} \tag{8}$$

These new statistics are called as *link* and *node compression* for the different similarity scores used in S_w. Because similarity scores are determined for consecutive snapshots, they do not explain the features of a snapshot but the stability of the system. That is why we do not consult the variance of similarity scores. We determine the best w as the one where the compression ratio is the lowest. That window size allows you to extract the most varied and not redundant snapshots. We show the effectiveness of the compression ratio by comparing their results with previously explained baselines in section 2.1.

Window Aggregation Algorithm for Proper Dynamic Network Extraction We previously proposed a window aggregation algorithm for extracting a proper dynamic network with a lower number of snapshots, having all the critical and necessary information for network analysis [31]. Here, we make small modifications to integrate the proposed compression ratios in the algorithm to increase its accuracy. The main idea behind this aggregation algorithm is that if a candidate snapshot is similar enough to already existing snapshots, we do not represent it separately in the final dynamic network but aggregate it into the existing model. The aggregation process is iterative and simple.

First, the best window size, w, is computed for the entire network by using the proposed compression ratio in the previous section, we generate the first snapshot, G_1 from the first time point of aggregation with a duration w. Then, for each following snapshot, G_i, we aggregate it and continue the iteration if G_i is similar to G_1. However, if it is not similar, we end the current aggregation process and start a new aggregation process for the rest of the time slices. The pseudo code is given in the algorithm 1.

Algorithm 1: *Window Aggregation Algorithm for Dynamic Networks*

```
Require: $\mathcal{C}, w, t_1, t_\theta$
Ensure: $\mathcal{G}$
 1: $G_{init} \leftarrow extract(\mathcal{C}, t_1, w)$
 2: $G_{aggr} \leftarrow G_{init}$
 3: $t_i \leftarrow t_1$
 4: while $t_i < t_\theta$ do
 5:     $G_i \leftarrow extract(\mathcal{C}, t_i + w, w)$
 6:     $sim \leftarrow S(G_{init}, G_i)$
 7:     $limit \leftarrow P(G_{init}, G_i)$
 8:     if $sim < limit$ then
 9:         $\mathcal{G} \leftarrow add(\mathcal{G}, G_{aggr})$
10:         $G_{init} \leftarrow G_i$
11:     else
12:         $G_{aggr} \leftarrow aggregate(\mathcal{C}, G_{aggr}, G_i)$
13:     end if
14:     $t_i \leftarrow t_i + w$
15: end while
```

This process requires a complex system $\mathcal{C}$ that evolves over time, a time window size w and the beginning and ending time points t_1 and t_θ respectively. It returns a proper dynamic complex network; $\mathcal{G}$. Lines #1- 3 are devoted to the initialisation of the following objects; the inital snapshot, G_{init}, the base snapshot from which the aggregation will take place, G_{aggr}, and the time flow counter, t_i. At line #5, the snapshots are extracted sequentially in a loop for each coming time duration of length w into the variable G_i. After that, similarity and limit values are calculated at line #6, 7, by $S(.)$ and $P(.)$ respectively. One can use either J_{link} or J_{node} which are introduced in Equations 3 and 4 respectively, or their mean value for similarity. In our experiments, we adopt mainly to J_{link} for quantifying structural changes. For $P(.)$, we use the statistical significance given in Equation 6. The algorithm checks whether these two snapshots are similar enough to be aggregated, i.e. their similarity score being larger than their statistical significance value at line #8. If they are not similar, the snapshots aggregated so far are added to $\mathcal{G}$ at line #9. Then, in order to continue the process, the current snapshot becomes the initial snapshot for the rest of the calculation from this point on (line #10). Otherwise, if the snapshots are similar, then they are aggregated to form a single snapshot that represents both of them at line #12. Finally, the time slice is shifted by the window size at line #14, in order to continue computation the next iteration till the end of the time interval.

Note that if w is too large, some critical time points where the system exhibits important change can be ignored. But, if w is too small, consecutive snapshots can be too similar, especially if the system is changing slowly. Here, we use the link similarity and its statistical significance, which are introduced in Equations (3) and (6) respectively. If the system is changing slowly, comparing consecutive snapshots can be misleading because the changes occur slowly and they cannot be captured in a short period of time. In order to capture these smooth and latent changes, we propose to measure the similarity of the current snapshot with the first snapshot that the aggregation process started with, rather than comparing consecutive ones. One of the most important points for the aggregation process is to decide if the measured similarity is critical. If the J_{link} score of the two snapshots under consideration, G_i and G_1, is greater than P, we aggregate G_i into the networks that have already been aggregated with G_1. An aggregated dynamic network can be seen as a compressed network including informative snapshots and not having redundancies. Although we use link similarity here, it can be applied with applied with any other network similarity metric having a significance threshold.

3. Experiments

We conducted experiments on four data sets: Enron Email, Haggle Infocomm, MIT Reality Mining, and Sabanci University Wi-Fi log data sets. These data sets are used in the studies dedicated to the same problem as us [7,29,22]. The statistics of these sets are given in Table 1.

Table 1. Data sets

Data Set Name	**Size**	**Time Span**
Enron Email [29,13,16]	151 employees	53 months
MIT Reality Mining [29,30,4,7,11,13,16]	90 users	9 months
Haggle Infocomm conference [4,13]	41 participants	4 days
Sabanci University Wi-Fi logs [22]	5378 devices	9 days

The Enron data set is an email set sent to or received by Enron employees between 1997 and 2003. We used the ”From” and ”To” fields of each email as nodes to generate snapshots with the timestamp. When there are multiple emails in the To field, we split them as if they were unique emails. Therefore, the clean data set consists of a date field, a single email address for the From field, and lastly, a single address for the To field. Each of the From and To fields represents a node in the snapshots. Thus, every transaction is a link between these nodes.

The MIT Reality Mining data set consists of Bluetooth discovery data suggesting social interaction between people [10]. The data set was generated at the MIT Media Laboratory over a nine-month period. During the study, 93 participants were given Nokia 6600 cell phones, and a comprehensive data set was composed based on various features such as Bluetooth device discovery scans at five-minute intervals, voice and text messages, active applications, etc. We use only Bluetooth discovery data in which the source mac address and target mac addresses are captured with the timestamp. In our study, we represent each Bluetooth connection as a unique event. The source mac address and each target mac address are represented as nodes, and each transaction is a link between these nodes.

The Haggle Infocomm data set contains bluetooth discovery data like the MIT Reality Mining data set. It contains 98 unique participants, who have scanned 4724 unique devices during the Infocomm conference in 2006 in Barcelona. Special devices called iMotes are deployed throughout the area. Seventeen of them are long-range static iMotes; three of them have been placed in the lift of the hotel; the rest of the 98 are the participants of the workshop. Instead of the mac addresses, we use the IDs as nodes. Each row in the data set represents a discovery event. Hence, they are considered links between nodes. Since the experiment was done between April 24th and April 27th, 2006, and the iMotes were activated on April 23rd at 5:01 pm, we shifted the timestamp to match the correct dates.

Sabanci University Wi-Fi log data consists of system connection metadata on campus from the 2016–2017 fall semester, which was logged in every 10 minutes by Sabanci University's IT department. Every logging procedure consists of all devices' connection activity at the Wi-Fi access points in the campus. Thus, logging 10 minutes allows us to track the changes occurring in less than 10 minutes as well. However, the 10 minute period is critical because even if a connection continues longer than 10 minutes, it is represented as broken and reconnected in the data set. We fix this issue by preprocessing the data. Campus life shows a similar circadian rhythm for all periods. For the sake of comprehensibility, we only interpret nine representative days of the signals in this study, even if the data was collected for 137 total days. But the results and comments presented for the 9-day time period are valid and they can be expanded to the overall period of the data set. The records include the device ID, connection and disconnection timestamps, and the Wi-Fi Access Point (WAP) name. The device IDs were anonymized by assigning unique values to each MAC address connected to any WAPs on the campus. We create a node for each device ID that appears in the system. If two devices connect to the same access point for a given time interval, we put a link between them. A link between two nodes might be a sign that those devices are in the same place.

In [22], we have discovered circadian rhythms in the dynamic network representation of the daily cycle of Sabanci University campus life too. Similar circadian rhythms of human activity from electronic records were discovered by [2,25,15]. In [2], the authors underline two major periods of circadian rhythm; *day time* and *night*. Accordingly, there are cycling ascending and descending interaction activity. In our previous experiments, those periods corresponded to the hours of the day when the campus gets active and passive. Indeed, modelling these kinds of systems with a fixed w for both active and passive periods results in the extraction of many redundant snapshots in the passive period. Also, there is a risk of missing important activities in the active period.

We consult to the previous studies for determining the candidate w values. Accordingly, for Enron, the duration less than 1 day results unrealistic snapshots having several unconnected components with sparse link structure. That is why we consider one day and seven, fifteen, thirty, sixty and hundred eighty days for w. For MIT Reality Mining, we created snapshots with w sizes 3-hour, 6-hour, 12-hour, and lastly 24-hour. For Haggle Infocomm, the w sizes are determined as every minute, every five minutes, every fifteen minutes, and every hour. Finally for Sabancı Wi-Fi logs, candidate w sizes are five, ten, fifteen, thirty and sixty minutes intervals.

4. Results

In the following sections, we first show and interpret the results of baselines and our compression ratios together to find the best w among candidate intervals. Then, we evaluate the results of the proposed aggregation algorithm by interpreting the topological properties of the network, namely diameter, average degree, average distance, or node number.

4.1. Choosing The Best Window Size

The best window sizes for four data sets are determined by TWIN, stability, average link similarity, and link and node compression ratios. For TWIN, we adopt the network diameter as it is done in [29]. TWIN results are displayed in Fig. 2. In the figure, we show the variance and compression ratio of the diameter for different w. The best w is the one whose variance and compression ratio are the closest to each other. Accordingly, for Enron, the best w is 7 days. For Haggle Infocomm, in fact, variance and compression ratio coincide on both 5 and 60 minutes. For reality mining, the best w is 3 hours.

These three data sets were previously studied by Sulo et al. [29] by applying TWIN. We have found similar results with them, except for reality mining. In a previous application, Sulo et al. found 6 hours for the best w [29]. In fact, the reality mining network can be extracted in two different ways. One of them is putting a link between two devices that see each other at a given period of Bluetooth scan. The other way is to put a link between two devices that scan the same devices at a given period. The former network is denser and includes noisy links, while the latter takes into account only the main devices that participate in the experiment. We extract the snapshots according to the former method. This can be the reason why our implementation results are different from the ones presented in [29]. For other data sets, the reliability of our TWIN implementation and experimental set up has been validated. For Sabanci University Wi-Fi logs, which is a new experimental data set for this problem, the best window size that TWIN finds is 15 minutes.

In Figure 3, we show both stability scores and average link similarity that are found for different data sets. Stability is a specific version of average link similarity weighted on the size of the minimum link set (please refer to Section 2.1 for details). These data sets have never been studied before by stability. According to both stability and average link similarity, the most stable window size for Enron is 15 days. For Haggle Infocomm,

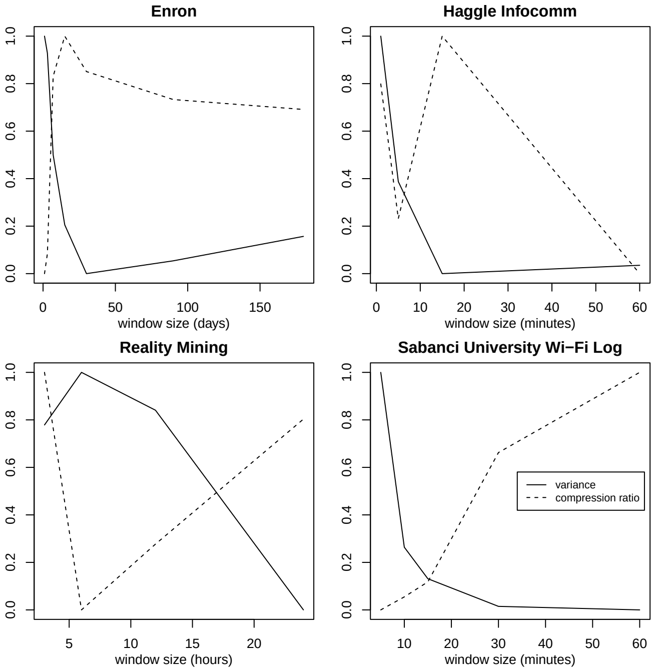

Fig. 2. TWIN results calculated by variance and compression ratio of diameter for Enron, Haggle Infocomm, Reality Mining and Sabanci University Wi-Fi logs. The most proper window sizes are 7 days, 5 minutes, 3 hours and 15 minutes respectively

the two metrics demonstrate opposite behaviour. The higher the duration, the higher the stability, but the lower the average link similarity. This can be due to not only the connections but also the system actors changing dynamically. The best w is 5 minute for average link similarity and 60 minutes for stability. For reality mining, both metrics show a similar trend. The best w is 3 hours. For Sabanci University Wi-Fi logs, 10 minutes seems to be a critical period. Two metrics show similar trends for longer periods than 10 minutes. But for lower periods, stability takes a higher value while average link similarity is lower. We have indicated that the data is collected every 10 minutes in the section 3. That is why this period is already significant for the breaking point of different logging moments. We have found out that the best w is 5 minutes for stability and 10 minutes for average link similarity.

The results of node and link compression are shown in Fig. 4. Overall results are listed in table 2. Accordingly, for Enron, we find 15 days to be the best window size. Both node and link compression results agree on this value. Our result is the same as stability and average link similarity but different from the TWIN result. That's why we examine in detail the diameter signal of Enron when $w = 7$ and 15 days. Those signals are shown in Fig. 5. The compression ratio found by TWIN for diameter is the same at 7-day and 15-day intervals. Because the variance of 7 days in the calculations is greater than that of 15 days, TWIN determined that 7 days is the best w. However, as we mentioned previously, topological properties and variance are not scale invariant.

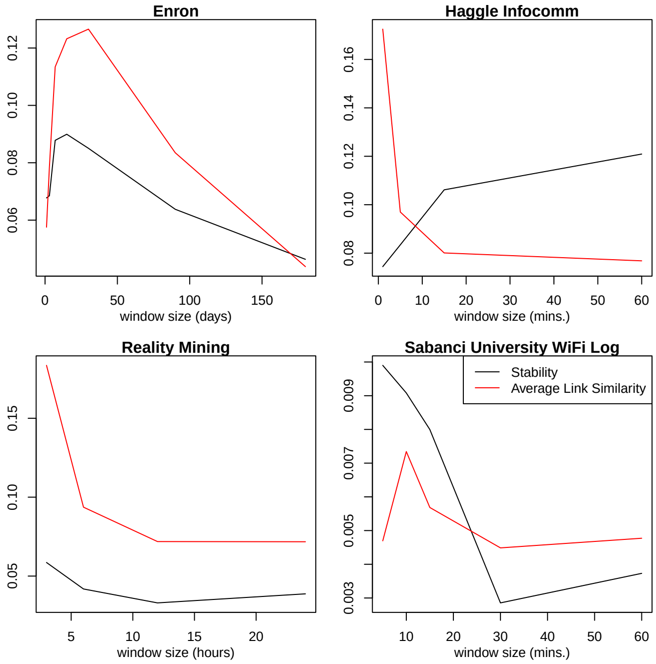

Fig. 3. Stability scores calculated by Equation 5. The most proper window sizes are 15 days, 60 minutes, 3 hours and 5 minutes respectively

Topological properties can have different scores reflecting the same information for two snapshots, or vice versa. In our detailed analysis, we saw that the diameter scores found for these two w were different. When we looked at the normalised standard deviation with the mean over the obtained values instead of simple variance, we found that the variation of the 15-day interval was higher. Hence, variance, which is not a normalised measure, can be misleading for explaining the noise of generated signals.

In order to get deeper in the analysis for this hypothesis, we examined the diameter signals for the two w in detail. The results are depicted in Fig. 5. Some of the important events related to the Enron Company's collapse are marked with arrows and with different colours in the Fig. 5. Accordingly, the first event, Event1, in the signal part that coincides with the date of December 2001, is the period when Enron announced bankruptcy. Event1 is hidden in the plot of $w = 7$ because it is both noisy and does not reflect the significance of the event with a critical peak. Contrarily, we can see important details with less noise but in sufficient detail in the plot of $w = 15$. The critical diameter peak is also remarkable in the section corresponding to the bankruptcy date.

More information can be retrieved from Enron signals. For example, Event2 is the period when George Bush named Kenneth Lay, who is Enron's chairman and contributed more than $290,000 to George Bush's election campaign, as an adviser to his presidential transitional team in January 2000. We catch this event in the plot of $w = 15$ with a high peak. However, it is hidden among the noise of the signal and cannot be caught in the plot of $w = 7$. Similarly, around February and March 2002, the hearing started before

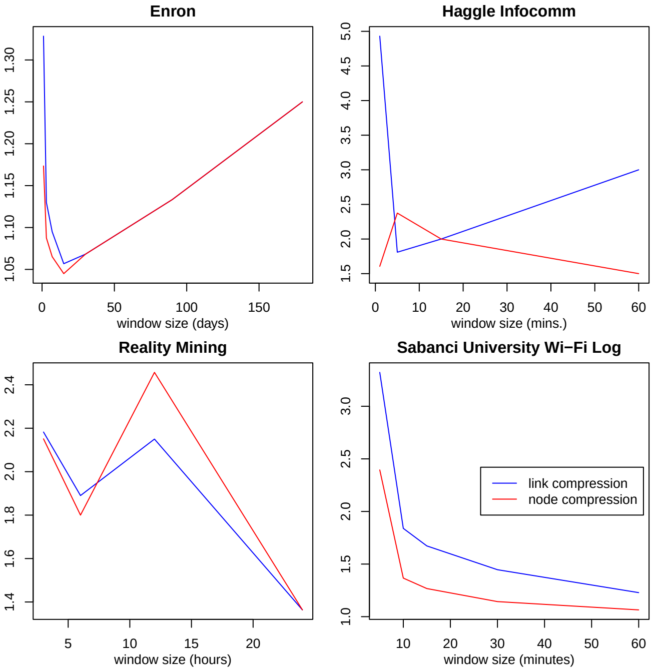

Fig. 4. Link and Node Compression results for Enron, Haggle Infocomm, Reality Mining and Sabanci University Wi-Fi logs. The most proper window sizes are 15 days, 5 minutes, 6 hours and 60 minutes respectively

Table 2. The best window sizes for four data sets according to different methods

	Enron	**Haggle Infocomm**	**Reality Mining**	**Sabanci University Wi-Fi logs**
TWIN	7 days	5 mins./60 mins.	3 hours	15 mins.
Stability	15 days	60 mins.	3 hours	5 mins.
Average Link Similarity	15 days	5 mins.	3 hours	10 mins.
Node Compression Ratio	15 days	60 mins.	6 hours/24 hours	60 mins.
Link Compression Ratio	15 days	5 mins.	6 hours/24 hours	60 mins.

the Senate. This event, Event3, is clearly visible in the plot of $w = 15$. However, it is represented by multiple snapshots in the plot of $w = 7$. Moreover, on June 15, 2002, Arthur Andersen was convicted. This event, Event4, is captured by both $w = 7$ and $w = 15$. We could discern many significant events from the company's routine, such as the resignation of the chairman and chief executive of Enron at the peak points of the signal of $w = 15$. When we look at the overall picture, the window size found by node and link compression seems to be more informative in terms of reflecting real-world phenomena for this system than the one that TWIN found.

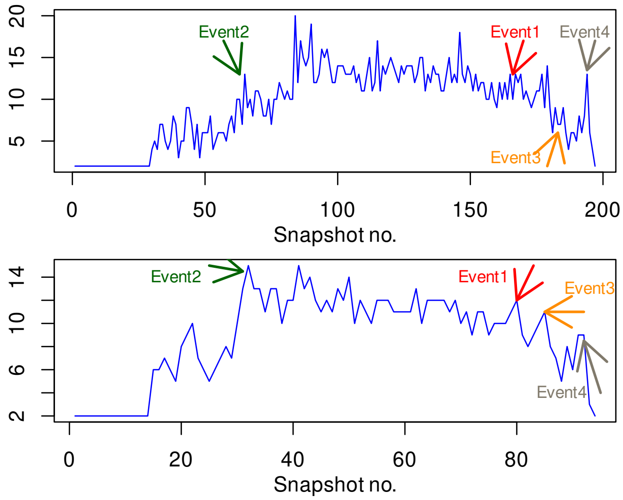

Fig. 5. Top and bottom time series are Enron diameter scores for $w = 7$ and $w = 15$ days respectively. The biggest bankruptcy in US history happened at December 2001. The date is indicated with red arrow in both plots

For Haggle Infocomm, link compression finds the best result for w as 5 minutes, but node compression has the opposite behaviour (see Fig. 4). In fact, node similarity represents the stability of the actors in the system. If the system is changing slowly or the studied w sizes are too small, it can be expected that the nodes of the system do not change considerably. Nevertheless, link similarity represents the stability of the connection between system actors. Although the system actors stay the same, their connections can change. Let us remind you that Haggle Infocomm is a set of controlled experiments including Bluetooth connections in a predefined environment. The participants of this conference did not change considerably, whereas their connections were actively changing during the conference. Thus, in this system, we take into account the results of link compression instead of node compression. As a result, we obtain the same result with TWIN and average link similarity. Our node compression result is the same as stability.

For Reality Mining, the lowest values of compression scores are obtained when w is 6 and 24 hours. Surprisingly, these are two different suggested w values, as explained in two separate articles, [29] and [11]. If we want to capture sudden changes occurring daily, we extract the network with $w = 6$. But if we want to observe the system with its macro changes and evaluate them, we use $w = 24$. The studied baselines, TWIN, stability, and average link similarity, result in different ways from our proposals. We have found 3 hours to be the best w according to these baselines. But the detailed topological analysis reveals that, as in the case of Enron, the noise of the $w = 3$ signals obscures the important events. We show the change of link number in the snapshots of $w = 3$, $w = 6$ and $w = 24$ in the figure 6 for the first two months of the data set. There is a slight difference between $w = 3$ and $w = 6$ signals, while the signal of $w = 24$ shows only weekly changes. We can distinguish the daily circadian rhythm from the weekday activity ups and downs by a cyclic signal in both $w = 3$ and $w = 6$. Moreover, the lower activity of the weekend is also discernible. When comparing $w = 3$ with $w = 6$, there is no significant difference, but the signal is more noisy when $w = 3$. For $w = 3$ and 6, the normalised standard deviation is 1.46 and 1.32, respectively. Accordingly, when w is 3 hours, extracted snapshots become noisy and seem to include the same information with a higher window size, $w = 6$. Among 6 and 24 hours, it seems like 6 hours is at an optimal granularity level, while 24 hours smooths too much of the daily activity. Therefore our proposed compression ratios suggest better window sizes that the baseline methods.

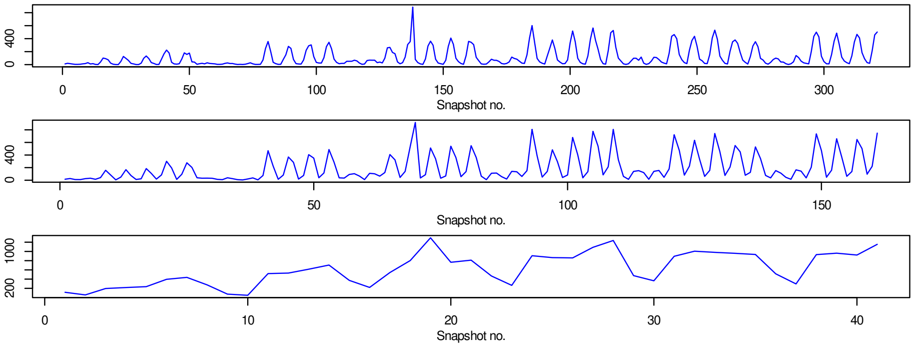

Fig. 6. Top, centre and bottom time series are MIT Reality Mining link number scores for first two months of $w = 3$, $w = 6$ and $w = 24$ hours respectively

When the compression ratio of Sabancı University Wi-Fi logs is analysed logs', an exponential decay with an increase of w is observed. Thus, the lowest values of both compression ratios are obtained for $w = 60$ minutes. Because this is different from baseline results, we study the signals in depth. The daily circadian rhythms are clearly observed in the average degree signals of Sabanci University Wi-Fi Logs for $w = 5$, $w = 10$, $w = 15$ and 60 minutes intervals (see Fig. 7). We can also notice the effect of lower activity at the weekend on both signals. The first cycle and the seventh and eighth cycles in the plots belong to the 4th, 9th, and 10th of December 2016, Sunday, Saturday, and Sunday respectively. The rest of the cycles reflect the daily activity of campus life. Accordingly, the activity increases from the morning till afternoon, afterwards it starts to decrease. Moreover, during the daytime, there are some periods when the activity peaks occur. The signal of 5, 10, and 15 minutes seems to be too noisy to reflect the daily picks. The normalised standard deviation of average degree signals is also given in the figures. As the signal plots show, the noise level of all signals lower than 60 minutes is higher than the one of 60 minutes. Similar to the results obtained for the previous data sets, our proposed method reflects more realistic, less noisy and easily interpretable results than the basic approaches for the Sabancı University Wi-Fi logs data set.

4.2. Qualitative Performance of Window Aggregation Algorithm

Once we determined the best w as explained in details in the previous section, we applied the window aggregation algorithm in order to obtain a proper dynamic network

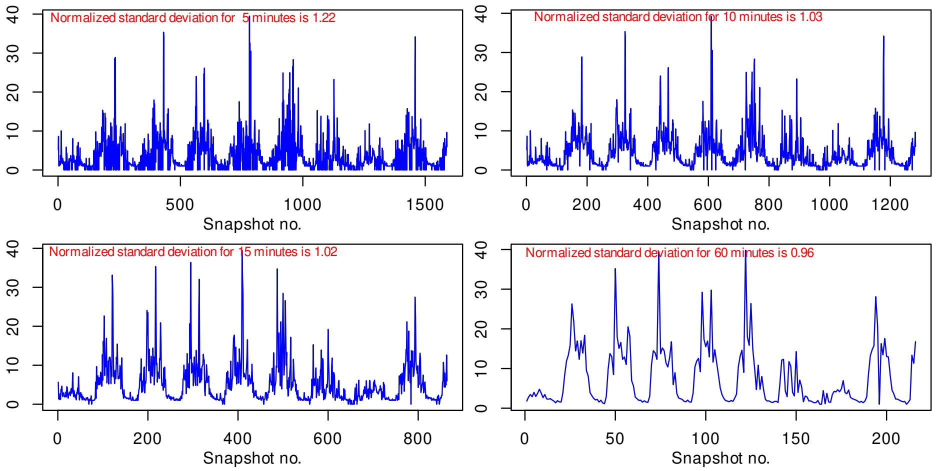

Fig. 7. Top-left, top-right, bottom-left and bottom-right time series are Sabanci University Wi-Fi logs average degree scores for $w = 5$, $w = 10$, $w = 15$ and $w = 60$ minutes respectively. The related normalised standard deviation is written with red text on each plot

representation for the data sets. We evaluate the resulting networks based on the similarity of consecutive snapshots as well as their topology. On one hand, for Enron and Reality Mining data sets, node and link similarity signals of snapshots with the constant w are difficult to analyse since there are no characteristic changes that give meaningful information about the system. as explained in details in the previous section.

On the other hand, we detect meaningful changes in node similarity signals of aggregated snapshots. These peak values were latent in the signal of constant snapshots. They were undiscernable in the flat stationary signals. Therefore, the aggregation strategy allows us to detect instantaneous changes in the system. Moreover, we could detect peak values, which are the points where the system shows a considerable change. These remarks are also validated with the Haggle data set.

Node and link similarity signals for both constant snapshots and aggregated snapshots of Haggle data sets are shown in Fig. 8. The selected window size for this analysis is 5 minutes. If we take a look at node similarity plots for constant snapshots, the signals show a circadian rhythm, but they are too noisy. This data set was collected for four days. We clearly observe daily cycles between day and night in constant window usage with the periodic increase and decrease in the signals. These types of circadian rhythms are common and explained in detail in [2]. Obtaining circadian signals does not add/contribute much to understanding of the studied system. However, although the signals generated by aggregated snapshots have no distinct trend, one can detect peak values and important changes in specific time points. Because circadian periods are aggregated and represented with fewer snapshots, we observe only important changes in those signals. The topology of these snapshots supports our interpretation.

We obtain similar circadian rhythms for Haggle Infocomm as well. In Fig. 9, diameter and average distance signals for constant (top layout) and aggregated (bottom layout)

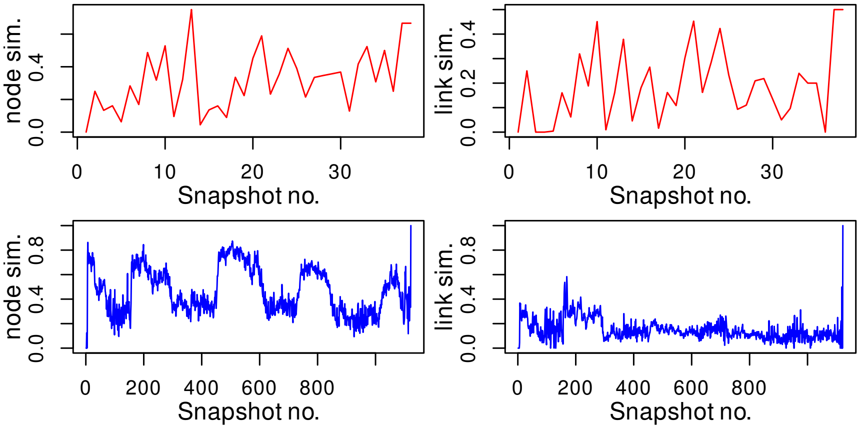

Fig. 8. Node and Link Similarity signals of Haggle snapshots extracted for $w = 5$ minutes from left to right, respectively. The plots of the snapshots extracted with aggregated windows and constant window are given in top and bottom layout, respectively

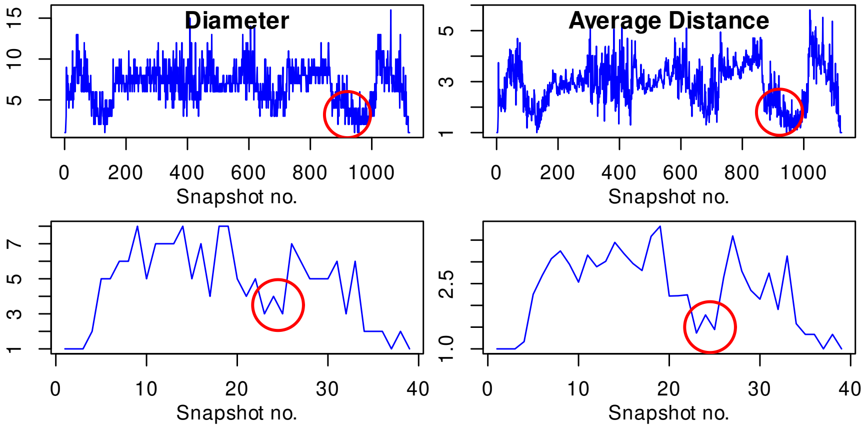

Fig. 9. Diameter and Average Distance Signals for $w = 5$ minutes for Haggle Infocomm. The plots of the snapshots extracted with constant window and aggregated windows are given in top and bottom layout, respectively. The red circles correspond to the same period of critical decrease for all plots

windows for the Haggle Infocomm data set are shown. The red circled zones of both plots correspond to the same period, the last night of the conference. Some of the participants have already left. That's why the number of nodes and links decreases, and the diameter and average distance get shorter. The aggregation process merges many snapshots before that period because the system repeats similar activity. However, the decline in this period is greater than in the previous one. As a result, they are represented as separate snapshots. The final night of the conference appears to be similar to other nights when we extract by constant window size. We cannot understand its difference because of having too many repetitive snapshots, i.e., a noisy signal.

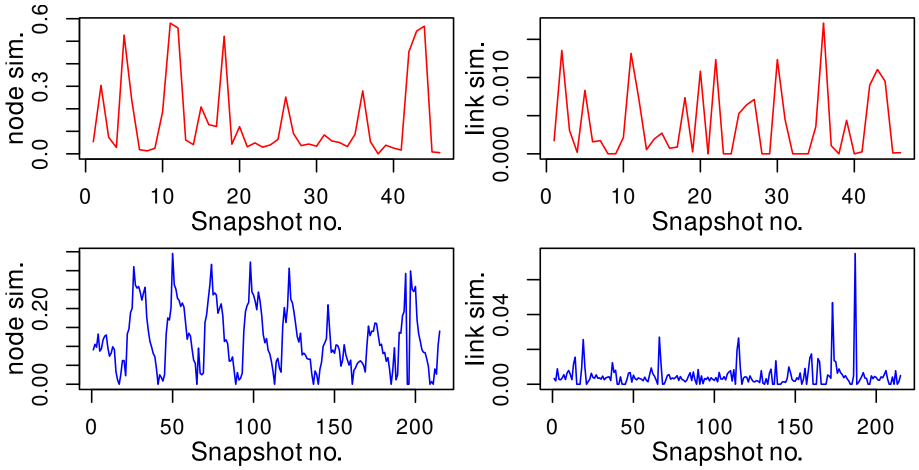

Fig. 10. Node and Link Similarity signals of Sabanci University Wi-Fi logs snapshots extracted for $w = 60$ minutes from left to right, respectively. The plots of the snapshots extracted with aggregated windows and constant window are given in top and bottom layout, respectively

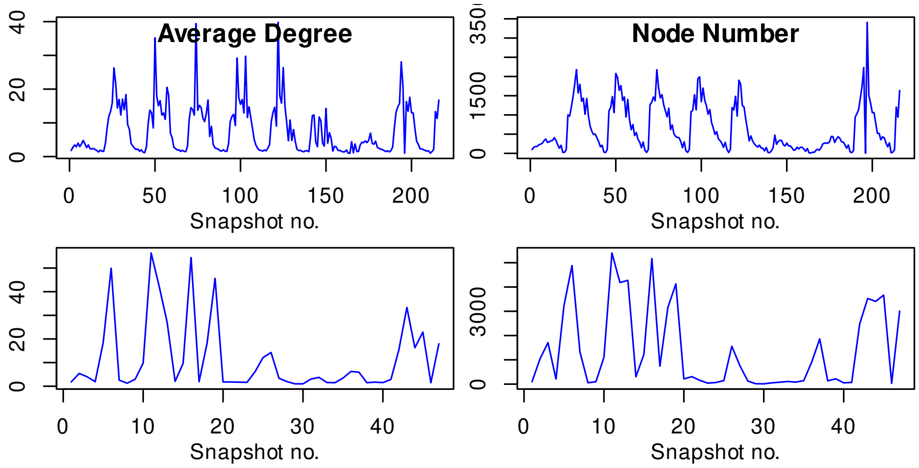

Fig. 11. Average Degree and Node Number Signals for $w = 60$ minutes for Sabanci University Wi-Fi logs. The plots of the snapshots extracted with constant window and aggregated windows are given in top and bottom layout, respectively

Another data set that reflects the circadian rhythms of the studied system is the Sabanci University Wi-Fi logs. Differently from Haggle experiments, Sabanci University Campus is an active area in which the members of the system change dynamically. Therefore, not only the link but also the node structure changes during the day and also the week. We observe the effect of the aggregation algorithm in Fig. 10 and 11. The similarities of consecutive aggregated snapshots are shown in the top plot of Fig. 10. Accordingly, especially from node similarity, we can clearly see the circadian rhythm, the consecutive increase and decrease of activity from day to night and, moreover, the lower activity of weekends. This fact is also observed in network topology (see Fig. 11). As in our previous comment about Sabanci University Wi-Fi logs, the campus population declines on weekends, only some students staying in the university dormitory connect to the Wi-Fi access points. However, during the week days, both students and university staff go on and off campus. They move around the campus by connecting to the different Wi-Fi access points at different locations, which in turn creates more activity on weekdays.

Comparing two dynamic networks through their signals; the one extracted by an aggregation algorithm and the one with constant window size, we notice clearly that the aggregation algorithm allows us to model the system by keeping its essential properties, such as the life cycle periods of the campus, with clearer signals. In Section 4.1, we saw that the node and link compression ratio let the modeller extract a dynamic network model with less noise. The aggregation algorithm seems to refine the modelling. This was a regular period when the campus exhibited its usual activities. That's why; we do not observe any peak or dramatic change except for circadian rhythms. The most remarkable result of the aggregation algorithm for Sabanci University Wi-Fi logs, however, is that it allows the extraction of the final dynamic network with features reflecting almost no noise, while still retaining the dynamics and properties of the studied system.

4.3. Discussion

The experimental results reveals that both novel compression ratios and aggregation algorithm reduce the noise in the data. They ensure the extraction of clearer, dynamic network models whose analysis is easier for knowledge discovery. Data sets collected from telecommunications networks such as e-mail exchange, Facebook and Twitter are already convenient for network analysis. For them, direct relationships between person A and person B can be represented with network links. Therefore these data sets cause relatively less noise or less challenge for dynamic network modelling. On the other hand, proximity data sets, such as Haggle Infocomm or Sabanci University Wi-Fi Logs, do not directly reflect social relationships, but they may help us discover common behaviours among people in the same environment. Stopczynski demonstrates that the physical proximity of people can be inferred by their connection to the same Wi-Fi access points within a sufficiently small time window [28]. However, there are many problems with limiting the handling of raw logs. Some of them are listed in [17] as first because it is noisy data compared to other location data sources such as GPS. Second, the data might contain some misleading information. However, our proposed aggregation algorithm helps us to eliminate the noise when using optimal w. Consequently, it might be useful for discovering the behaviour analysis of system actors in further analysis.

Although similarity-based compression ratios and aggregation algorithm result in less noisy and more informative dynamic network models, they can be criticised because they consider only local information. The methodology proposed here is completely based on the Jaccard similarity of node-or link-sets of consecutive snapshots. Accordingly, the topological role or position of the nodes or links is not privileged. For example, the disappearance of one critical link in the role of a bridge and the disappearance of one simple link return the same link similarity result, although their effect on the system is completely different. We measure the role of the nodes, or links, via their topological properties. However, it can be possible to neglect the system's emerging properties by using only local comparison of nodes or link sets. Because such an analysis is outside the scope of this work, we do not consider this problem. For further information please refer to our previous study [22], where the authors showed that the use of similarity instead of topological properties allows one to extract more informative models.

5. Conclusion

This research focuses on the accurate modelling of complex systems, such as interacting objects in the form of dynamic networks that evolve in real time. It solves both of the model's key problems: first, selecting the ideal temporal window size, w, and second, extracting the appropriate dynamic network using w. It provides two novel dynamic network compression metrics based on the similarity ratios of sequential snapshots to find w, as well as a window aggregation approach using the previously determined w. According to experiments on four data sets, compression ratios can extract more noise-free and informative networks than baseline approaches. Furthermore, the aggregation method has managed to lower noise levels even further without compromising the system's overall and crucial features.

Three of the studied data sets consist of Bluetooth or Wi-Fi connection data, which may contain the proximity information of the people in the system. The analysis of these data sets with basic methods is quite challenging, as they are noisy. However, people's behavioural patterns are hidden behind them. The most important principal for these data sets is that a noise-reduced analysis model could be built by using proposed compression ratios and aggregation algorithms. In Sabanci University Wi-Fi log data, for example, we were able to observe system-specific circadian results such as campus life cycle and weekday-to-weekend activity differences with noise-reduced signals. Or, in Haggle Infocomm data, we were able to observe a decrease in activity on the last day of the conference, which is a detail that was not observed before the noise was eliminated by our proposals. Apart from all these three sets, we were able to capture the important dates of the collapse of the company in the Enron data set in the networks obtained by our approach. Those event details were not observed in the noisy modelling.

All these results show us that we can make a correct dynamic network model, especially in data sets that contain complex structures. The accuracy of the extracted models should be supported by further analysis like link prediction, community detection, knit group finding, or behaviour prediction in the next steps. This work can be extended in several ways. For example, the use of candidate w when determining the best w still requires system expertise. This can be improved over the proposed compression ratios. These ratios, in their current form, compress the snapshots against an analytically calculated upper limit of similarity. This limit gives us the lowest possible similarity. However, we have not yet established the highest possible similarity analytically. More precisely, we can decide that two consecutive snapshots are similar but cannot decide that they are dissimilar. This way, we can further refine the compression ratio by fixing a second limit. This allows us to determine more accurate time windows. Thus, as a result, we can achieve the best w by optimising w value according to lower and upper limits without using candidate w. The work proposed here is based on the local measurement of Jaccard similarity and does not take into account the topological position of links or nodes. This can cause neglect of the emerging and non-linear properties of the studied system. Another perspective of this work could be to measure the weighted Jaccard in a way to quantify the topological importance of the nodes or links for considering emerging properties. Furthermore, this work can be adapted and tested on the systems from different domains that are evolving at different speeds. Thus, the effect of the system's being fast or slow on the proposed methods can be examined.

Acknowledgments. We thank Prof. Dr. Selim Balcisoy from Computer Engineering Department of Sabanci University for providing Sabanci University Wi-Fi Log dataset. This article is partially supported by Galatasaray University Research Fund (BAP) within the scope of project number fba-2021-1063, and titled "Niteliklendirilmiş çift yönlü ağlarda bağlantı tahmini ile öneri sistemleri geliştirilmesi".

References

1. Aggarwal, C., Subbian, K.: Evolutionary network analysis: A survey. ACM Computing Surveys 47(1), 10:1–10:36 (2014)
2. Aledavood, T., López, E., Roberts, S.G.B., Reed-Tsochas, F., Moro, E., Dunbar, R.I.M., Saramäki, J.: Daily rhythms in mobile telephone communication. PLOS ONE 10(9), 1–14 (09 2015), https://doi.org/10.1371/journal.pone.0138098
3. Blonder, B., Wey, T.W., Dornhaus, A., James, R., Sih, A.: Temporal dynamics and network analysis. Methods in Ecology and Evolution 3(6), 958–972 (2012)
4. Caceres, R.S., Berger-Wolf, T., Grossman, R.: Temporal scale of processes in dynamic networks. In: 2011 IEEE 11th International Conference on Data Mining Workshops. pp. 925–932 (Dec 2011)
5. Cazabet, R.: Data compression to choose a proper dynamic network representation (2020)
6. Chiappori, A., Cazabet, R.: Quantitative evaluation of snapshot graphs for the analysis of temporal networks (2021)
7. Clauset, A., Eagle, N.: Persistence and periodicity in a dynamic proximity network. CoRR abs/1211.7343 (2012)
8. Dakiche, N., Tayeb, F.B., Slimani, Y., Benatchba, K.: Sensitive analysis of timeframe type and size impact on community evolution prediction. In: 2018 IEEE International Conference on Fuzzy Systems. pp. 1–8 (2018)
9. Donnat, C., Holmes, S.: Tracking network dynamics: A survey using graph distances. The Annals of Applied Statistics 12(2), 971 – 1012 (2018), https://doi.org/10.1214/18-AOAS1176
10. Eagle, N., Pentland, A.S., Lazer, D.: Inferring social network structure using mobile phone data. In: PROCEEDINGS OF NATIONAL ACADEMY OF SCIENCES. pp. 127–136 (2009)
11. Eagle, N., (Sandy) Pentland, A.: Reality mining: Sensing complex social systems. Personal Ubiquitous Comput. 10(4), 255–268 (Mar 2006)
12. Fish, B., Caceres, R.S.: Handling oversampling in dynamic networks using link prediction. In: Machine Learning and Knowledge Discovery in Databases. pp. 671–686. Springer International Publishing (2015)

13. Fish, B., Caceres, R.S.: A supervised approach to time scale detection in dynamic networks. CoRR abs/1702.07752 (2017)
14. Holme, P., Saramäki, J.: Temporal networks. Physics Reports 519(3), 97–125 (2012)
15. Jo, H.H., Karsai, M., Kertész, J., Kaski, K.: Circadian pattern and burstiness in mobile phone communication. New Journal of Physics 14(1), 013055 (jan 2012), https://doi.org/10.1088%2F1367-2630%2F14%2F1%2F013055
16. K. Darst, R., Granell, C., Arenas, A., Gomez, S., Saramäki, J., Fortunato, S.: Detection of timescales in evolving complex systems. Scientific Reports 6 (04 2016)
17. Kjærgaard, M.B., Nurmi, P.: Challenges for social sensing using wifi signals. In: Proceedings of the 1st ACM Workshop on Mobile Systems for Computational Social Science. pp. 17–21. MCSS '12, ACM, New York, NY, USA (2012)
18. Krings, G., Karsai, M., Bernhardsson, S., Blondel, V.D., Saramäki, J.: Effects of time window size and placement on the structure of an aggregated communication network. EPJ Data Science 1(1), 4 (May 2012)
19. Kuhn, F., Oshman, R.: Dynamic networks: Models and algorithms. ACM SIGACT News 42(1), 82–96 (2011)
20. Léo, Y., Crespelle, C., Fleury, E.: Non-altering time scales for aggregation of dynamic networks into series of graphs. In: Proceedings of the 11th ACM Conference on Emerging Networking Experiments and Technologies. CoNEXT '15, Association for Computing Machinery, New York, NY, USA (2015), https://doi.org/10.1145/2716281.2836114
21. Medo, M., Zeng, A., Zhang, Y., Mariani, M.S.: Optimal timescale of community detection in growing networks. CoRR abs/1809.04943 (2018)
22. Orman, G.K., Türe, N., Balcisoy, S., Boz, H.A.: Finding proper time intervals for dynamic network extraction. Journal of Statistical Mechanics: Theory and Experiment 2021(3), 033414 (mar 2021), https://doi.org/10.1088/1742-5468/abed45
23. Real, R., Vargas, J.M.: The probabilistic basis of jaccard's index of similarity. Systematic Biology, JSTOR 45(3), 380–385 (1996), www.jstor.org/stable/2413572
24. Rossi, R.A., Gallagher, B., Neville, J., Henderson, K.: Modeling dynamic behavior in large evolving graphs. In: Proceedings of the Sixth ACM International Conference on WSDM. pp. 667–676 (2013)
25. Song, C., Qu, Z., Blumm, N., Barabási, A.L.: Science 327(5968), 1018–1021 (2010)
26. Soundarajan, S., Tamersoy, A., Khalil, E.B., Eliassi-Rad, T., Chau, D.H., Gallagher, B., Roundy, K.: Generating graph snapshots from streaming edge data. In: Proceedings of the 25th International Conference Companion on WWW. pp. 109–110 (2016), https://doi.org/10.1145/2872518.2889398
27. Spiliopoulou, M.: Evolution in social networks: A survey. In: Social Network Data Analytics, chap. 6, pp. 149–175. Springer (2011)
28. Stopczynski, A., Sekara, V., Sapiezynski, P., Cuttone, A., Madsen, M.M., Larsen, J.E., Lehmann, S.: Measuring large-scale social networks with high resolution. PLOS ONE 9(4), 1–24 (04 2014)
29. Sulo, R., Berger-Wolf, T., Grossman, R.: Meaningful selection of temporal resolution for dynamic networks. In: Proceedings of the Eighth Workshop on Mining and Learning with Graphs. pp. 127–136. MLG '10, ACM, New York, NY, USA (2010)
30. Uddin, S., Choudhury, N., M. Farhad, S., Rahman, M.: The optimal window size for analyzing longitudinal networks. Scientific Reports 7 (12 2017)
31. Çolak, S., Orman, G.: Aggregating Time Windows for Dynamic Network Extraction (2021)

Günce Keziban Orman has a PhD in Computer Science from INSA de Lyon. She is working as a full-time assistant professor at Galatasaray University, Istanbul, Turkey.

She works on complex network analysis in general and accurate network modelling, community detection, and link prediction in particular.

Received: September 29, 2021; Accepted: December 26, 2021.