Universiteit Leiden
The Netherlands

Opleiding Informatica

Visualization tool for motifs in multilayer temporal networks

Yven Lommen

Supervisors:
F.W. Takes & W.A. Kosters

BACHELOR THESIS

Leiden Institute of Advanced Computer Science (LIACS)
www.liacs.leidenuniv.nl

14/07/2020

Abstract

Studying motifs allows us to better understand the interactions between entities in different kinds of networks. These kinds of networks can range from social networks to economic networks. Motifs are small subgraphs within a network. Motifs can have characteristics such as timestamps and layers, where different layers denote different types of interactions. Counting motifs within networks can be done with fast algorithms. However, these algorithms have the downside of not being user friendly in terms of usage by non-experts and visualization of resulting output. The tool presented in this thesis allows researchers without a background in computer science or programming to gain insights in large multilayer temporal networks by studying their motifs using interactive visualizations. The tool also allows for comparison of motif counts between networks. In experiments we compare real-world networks with random networks of approximately the same size to determine whether or not a motif in a real-world network is significant. The tool helped to discover these insights in the network by denoting dominant motifs which describe the behaviour of entities in the networks.

Contents

1 Introduction

The field of network science [Bar16] has emerged as a separate discipline in the twenty-first century. It aims to understand complex systems by studying the interactions between entities within a system as a network. These networks range from economic networks to social networks. It is very common to represent these networks as a static graph, presenting entities as nodes and interactions or links as edges. However, many of these systems are dynamic and the links between entities change over time [HS12]. These dynamic networks are called temporal networks and can be represented by a series of timestamped edges (temporal edges). Networks can also be extended with the help of layers which indicate different kinds of links [KAB+14].

When the interactions between the entities (nodes in the graph) are studied, motifs (Figure 1; see Section 2.2 for details) can be considered. Motifs are small subgraphs within a bigger graph. These motifs also carry the characteristics of the larger graph from which they are derived, such as timestamps or layers. We want to study motifs because they give insight in how the entities interact with each other on the meso-scale of the graph. In particular, motifs can be counted and these counting results can provide insights in the entities. Take an email service and a text message service, for example. The nodes (in this case persons) and the links (messages) could behave differently in the email service than in the text message service. If so, we want to know how they differ. Network motifs can answer questions like these. In addition, they can describe the most common behaviour of entities in the individual networks.

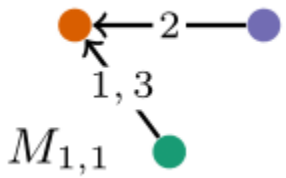

Figure 1: Example of a motif. The numbers on the edges denote the order in which they occur.

Multiple algorithms were introduced over the years [MSoI+02, PBL17] to count the occurrences of particular motifs. The motifs that are counted with these algorithms are currently no larger than 2 or 3 nodes and have 3 edges. This is because otherwise the computation time required for counting larger motifs is too long. The current algorithms and tools are not very user friendly for novice users. Output is typically raw data with very little metadata. The usage of such motif counting tools and algorithms is limited, while potential insights from motifs are possibly very meaningful.

In 2019, a counting algorithm was introduced by Boekhout et al [BKT19]. The algorithm is capable of counting multilayer temporal motifs. However, there are many different motif types and layer permutations for which the number of motif instances is counted. This causes for the results to be very overwhelming, which makes it almost impossible to extract valuable information or gain insights for domain experts. The problem discussed above sparked the first research question of this thesis:

Can a user friendly tool be built that allows for thorough analysis of the results from multilayer temporal motif counting algorithms?

In this context, the user in user friendly refers to users without computer science or programming experience. The words "thorough analysis" in this context mean clear visualizations of the results of the algorithm.

When we further analyze multilayer temporal graphs and their motifs we would also like to know how the motif count of different networks varies. With different networks we may refer to different types of real-world networks and random networks. Comparing a real-world network with a random network could show the significance of the motif count in the real-world network. This leads to the second research question:

What insights can be obtained from multilayer temporal motif count results when we compare real-world data and random data?

To answer this question the tool must allow for multiple results to be shown at once.

In the remainder of this thesis, first motifs are discussed in Section 2 to provide relevant background information. In Section 3 other research about network science and motifs is discussed. Then in Section 4 the approaches to addressing the two research questions are explained. In Section 5 the data used for the experiments and the results of the experiments with these different datasets are discussed and analyzed. Finally a conclusion as well as ideas for future research are given in Section 6.

2 Definitions

In this section the motifs that the algorithm from [BKT19] counts will be discussed. The tool described in this thesis is built upon this algorithm. The notations and definitions follow the ones given in [PBL17] and [BKT19].

2.1 Edges and graphs

The basic building block for a network structure is considered to be an edge: a link between an ordered pair of nodes, in this case a directed link. It is usually defined as a tuple (u, v) where u denotes the source node and v the destination node. Given a node set V with size $n = |V|$, a static graph $G = (V, E)$ is defined by a set E containing edges (u_i, v_i), for $i = 1, 2, \ldots, m$, with $u_i, v_i \in V$. Temporal edges have a timestamp t and such an edge is denoted as (u_i, v_i, t_i) where $t_i \in \{-1\} \cup \mathbb{R}^+$. A collection of temporal edges is called a *temporal graph*. A timestamp of -1 indicates that there is no known timestamp for that edge (in case of partial timing[1]). For layered edges we add a layer number l and we can define a multilayer temporal edge as (u_i, v_i, t_i, l_i) where $l_i \in \{1, 2, \ldots, \Lambda\}$ with Λ denoting the number of layers. A collection of these edges is a *multilayer temporal graph*. The underlying static graph of a multilayer temporal graph is the graph when we ignore all the timestamps, layers and duplicate edges. We henceforth assume that edges are always directed.

2.2 Motifs

We now provide a formal definition of multilayer δ-temporal motifs.

An r-node, s-edge, δ-temporal, λ-layer motif is a sequence of s edges,

$$M = ((u_1, v_1, t_1, l_1), (u_2, v_2, t_2, l_2), \ldots, (u_s, v_s, t_s, l_s))$$

that are time-ordered within a δ duration, $t_1 < t_2 < \ldots < t_s$ and $t_s - t_1 \leq \delta$ and where the edges range over λ different layers such that the underlying static graph is connected and has r nodes.

This means that the timestamps are responsible for making an ordering in which the edges of the motif occur. There is also the possibility for multiple edges between nodes (because of the timestamps and layers) and these are counted individually. This definition also allows for λ different layers, but fewer than λ layers are also possible. A motif $M = ((u_1, v_1, t_1, l_1), \ldots, (u_s, v_s, t_s, l_s))$ occurs in a multilayer temporal graph H when there is a time-ordered sequence $S = ((w_1, x_1, t'_1, l'_1), \ldots, (w_s, x_s, t'_s, l'_s))$ of s unique edges in H, such that

1. there exists a bijection f, such that $f(w_i) = u_i$ and $f(x_i) = v_i$ $(i = 1, \ldots, s)$,
2. the edges all occur within δ time, i.e., $t'_s - t'_1 \leq \delta$, and
3. there exists a bijection g on the layers, such that $g(l'_i) = l_i$ $(i = 1, \ldots, s)$, which holds for all motifs within a single search.

[1] Partial timing is whenever only a portion of the edges is timestamped.

Each such sequence of edges is called an instance of motif M.

With the formal definition as a basis, an r-node, s-edge, δ-temporal, λ-layer motif can be defined informally as well. A motif has r nodes and s edges, the edges must appear in a specific order making use of the timestamp. The first and last edge of a motif have a timestamp difference that is smaller than or equal to δ. The final characteristic is that an edge is part of one of λ different types of layers where each layer denotes a different kind of link. Take as an example a forum such as Stackoverflow (`https://stackoverflow.com`), where users can post questions and other people can answer. There could be three types of layers modelling user interaction, one which denotes an answer to a question, one which denotes a comment to an answer and one which denotes a comment to a question. The structure of these motifs is predetermined and an instance of such a motif occurs when we see such a structure in a multilayer temporal graph.

The algorithm introduced in [BKT19] counts all the instances of 2,3-node, 3-edge δ-temporal, λ-layer motifs. In Figure 2, all the 2,3-node, 3-edge δ-temporal motifs are shown. Note that the numbers on the edges denote the order in which the edges occur.

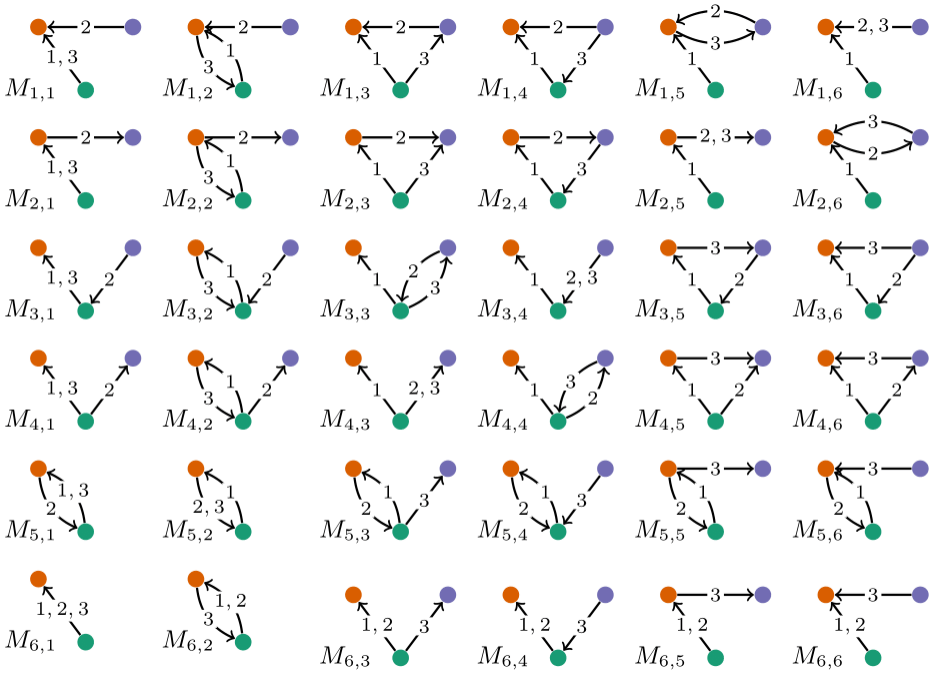

Figure 2: All 2,3-node, 3-edge δ-temporal motifs.

With 3 edges there exist $3^3 = 27$ layer permutations for each motif. These layer permutations are shown in Figure 3. The algorithm counts the instances of all the 27 layer permutations of each of the 36 motifs. This means that there are $27 \times 36 = 972$ motifs for which the number of instances are counted. When we reduce the number of layers to 2 there are $2^3 = 8$ layer permutations and $8 \times 36 = 288$ motifs. These 8 layer permutations can be seen in Figure 4.

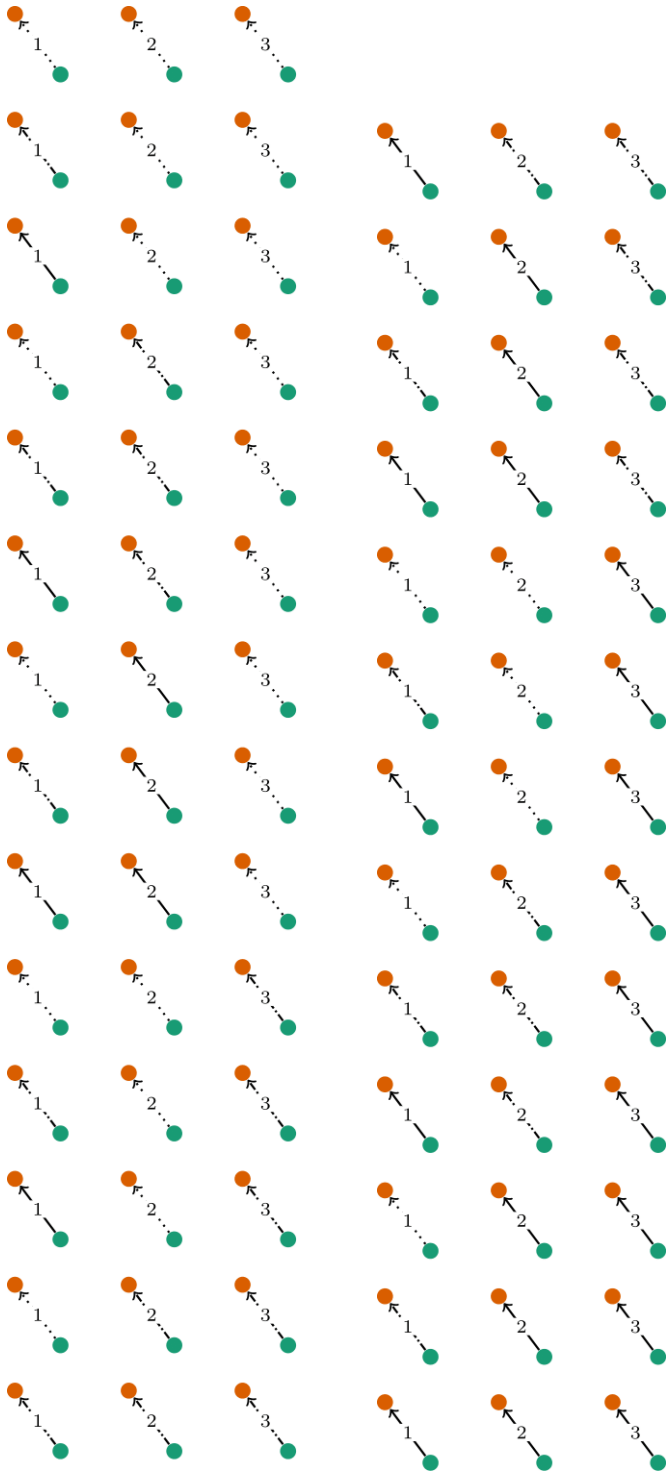

Figure 3: All 3 layer permutations with 3 edges.

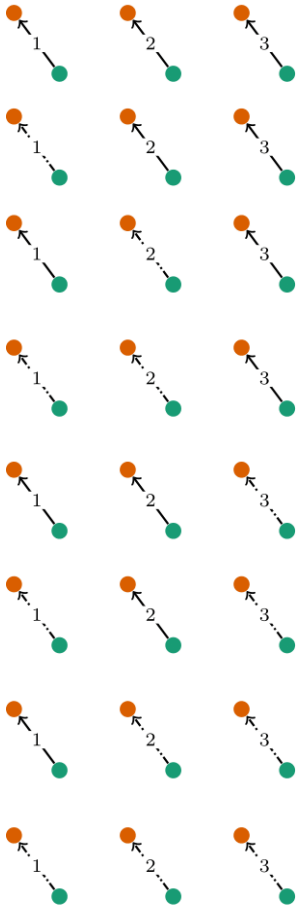

Figure 4: All 2 layer permutations with 3 edges.

3 Related Work

The importance and relevance of network science is discussed in [Bar16]. The book mentions that networks are at the heart of complex systems. Behind every complex system lies a network that encodes the interactions between the components of the system. Such systems can be seen everywhere: cellular networks, neural networks, social networks, communication networks and many more. Although network science exists longer, it just recently emerged as a separate discipline in the 21st century. According to [Bar16], one of the characteristics of network science is that it has an empirical and data driven nature. This is the main difference with graph theory; network science values the tools it uses based on the insights it offers about system properties and behaviour. This characteristic is important for this thesis because it is directly in line with the goal of this work, namely to obtain insights out of the computed motif counts.

Networks are often denoted as static graphs consisting of nodes and edges. In [HS12] it is shown that most networks are dynamic, realizing the definition of a temporal network. The study has two classes of temporal networks, the first, which we also adopt, represents a temporal network as a set of contacts, where links (i,j,t) are contacts with i and j nodes and t denoting the time. The second class of temporal network are called interval graphs, where the edges are not active over a set of timestamps but rather a set of intervals. Another characteristic of modern networks are different types of layers. In [KAB+14] a general formulation for multilayer networks is given. A *multilayer network* consists of a set of nodes, a set of elementary layers, a set of node-layer combinations in which a node is present in the corresponding layer and finally a set of edges where an edge is a pair of possible combinations of nodes and elementary layers.

As mentioned earlier, network science aims to gain information about the behaviour of a real-world network. For this we must zoom in on the network and for example analyze subgraphs. In [KK01] an algorithm called FSG is introduced. It finds all the connected subgraphs that appear frequently in a large graph database. This is a good example of looking at parts of the network. In [GGY+14] an algorithm is given that not only finds matching subgraphs but also ranks how "interesting" these are, and then a top-k of matching subgraphs can be given according to the "interesting" score.

The downside with the subgraph discovery algorithms discussed in the previous paragraph is that the subgraphs that are found were not predetermined and thus may or may not give new insight. The real problem with this is that using these algorithms a researcher may not know what subgraphs he or she is exactly looking for. Network motifs were introduced as recurring, significant patterns of interconnections [MSoI+02]. This study developed an algorithm for detecting network motifs where interactions between nodes were denoted as directed edges and motifs had three or four nodes. It showed that these motifs were crucial for understanding the mechanism of complex systems in several fields (biochemistry, neurobiology, ecology and engineering).

Because many modern networks are dynamic, their motifs may also be dynamic. Kovanen et al. introduced a definition and an algorithm for counting temporal motifs [KKK+11]. Their definition of temporal motifs assumes temporal edges in a motif must be consecutive events for a node. This definition allowed for fast counting algorithms but missed important structures. Many related edges occurring at the same time would not be counted together with this definition, for example. This problem was solved by the definition of [PBL17] which also introduced an algorithm for counting k-node, l-edge, δ-temporal motifs. Built on the techniques introduced in that work, [BKT19] introduced an algorithm for counting multilayer temporal motifs which incorporates the multilayer aspect and partial timing. The tool presented in this thesis is built upon this algorithm.

4 Approach

In this section the approaches to answer the research questions are discussed. In Section 4.1 the problems with the existing command line tool are discussed along with the solutions the tool needed to solve these problems. Then in Section 4.2 the approach for the comparison of networks is discussed. Finally in Section 4.3 the technical implementation of the tool is discussed.

4.1 Motif counting visualization tool

The tool that was made makes use of the algorithm and example program introduced in [BKT19]. The algorithm counts all the instances of 2,3-node, 3-edge δ-temporal, λ-layer motifs (from now on referred to as motifs). The goal of the tool is to give as much visualization-aided insight as possible from the result of this algorithm. To accomplish this the tool must have the following requirements:

- Present every mulitilayer motif count visually while keeping the option to see the exact value.
- For every multilayer motif count we want to know how the motif looks and in which layer each of the edges resides.
- For every multilayer motif count we want to know a number for the motif type and layer permutation so we can record which combinations of these numbers are interesting.

The outputfile of the example program is of importance for this tool and in Figure 5 part of such a file is shown. In the figure the run time of the algorithm and results of the first two layer permutation motif counts are shown (the subsequent layers are formatted in the same way). In this outputfile there are six columns with six rows of numbers for each layer permutation; these numbers and the formatting of them correspond with the motifs in Figure 2.

```
Run time: 0.016350s

Layer permutation 1:
10 17 10 14 10 2
6 14 15 16 4 8
16 14 44 47 0 2
14 16 19 32 16 12
12 7 14 17 4 2
11 16 28 26 11 13


Layer permutation 2:
0 0 0 0 0 0
0 0 0 0 0 0
0 0 0 0 0 0
0 0 0 0 0 0
0 0 0 0 0 0
0 0 0 0 0 0
```

Figure 5: Small part of outputfile of the example program.

The outputfile provides us with a motif count for each of the 36 motifs for all the 27 layer permutations (the number of layer permutations could be lower depending on the number of layers in the input network). But we cannot derive the following information from the file, which is relevant when interpreting these results:

- Which motif type is represented by which number?
- Which edge has which layer in each layer permutation?
- How big are differences between counts?

These three information gaps stated above must be filled to give proper insight in the network. The third information gap is mainly caused by the high number of motif types and layer permutations. The maximum number of motifs that are counted, with 3 layer graphs, is 972. The following functions describe how the tool fulfills the requirements which were stated earlier:

- The result of the multilayer temporal motif counting algorithm is mapped to an interactive heatmap. A heatmap is a data visualization technique used to give the reader a visual idea on how the cells are relative to each other. In a heatmap the cells with darker (in our case blue) shades have a higher value. In Figure 6 an example heatmap of the output in Figure 5 is shown.
- The illustrations of the motif type and layer permutation belonging to a cell in the heatmap are shown when this cell is clicked.
- The motif type number, layer permutation number and count belonging to a cell are shown whenever this cell is hovered upon or is clicked.

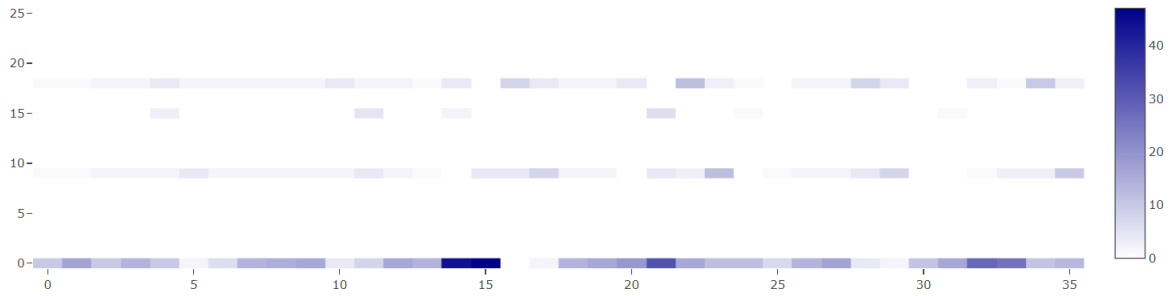

Figure 6: Heatmap of motif counts belonging to output shown in Figure 5.

4.2 Comparison of networks

The insights of one network are not always sufficient, in at least two regards. First, the question is often how a network differs from another network. For example, take two social networks like a WhatsApp and an e-mail network. These two networks are aimed at exchanging messages from one person to another. In this context nodes represent users and links represent messages. There is a large probability that users of WhatsApp send messages in a different way (many messages aimed at one person at a time, for example) than the users of e-mail (a few messages to different people). Comparing the motif count of each of these networks may reveal the difference in the messaging behaviour of the users. The comparison could also reveal that there is not much of a difference in the messaging behaviour between two networks. These kinds of insights are important when studying networks through motifs.

Second, comparing motifs in different networks helps understand the behaviour of entities in these networks. However, we do not know if some behaviour is just a coincidence. To solve this, we compare real-world network graphs with randomly generated graphs of approximately the same size. This will give insight in the significance of the found motifs. For example, a particular group of motifs could stand out (have a high count) in the real-world network but also stand out in the random network, how well do these motifs then describe the behaviour of the entities? The goal of comparing real-world graphs with random ones is trying to show the significance of the motifs found.

4.2.1 Comparison measures

In order to quantify the significance of motifs in real-world networks the *ratio* is used as a measure, as suggested in [TKWH18]. The ratio is an indicator of the significance of the found motifs in the real-world networks and is described below:

$$r(M, G) = |M(G)| \cdot (\sum_{H \in Y} |M(H)|/|Y|)^{-1}$$

Here $M(G)$ denotes the set of the found motifs of a particular type and layer permutation in the real-world graph G. This means $|M(G)|$ is the count of the occurrences of this motif. Furthermore, $M(H)$ is the set of motifs with the same type and layer permutation found in the random graph H and $|M(H)|$ is thus the count in the random graph. Finally, Y denotes a set of generated random graphs. A set of random graphs is used instead of only one random graph because this gives a more representative result of the motif count in random networks. For our experiments we use the average count of 10 random network models. This means we divide the motif count in the real-world graph by the average count of the random models and this is denoted as follows:

$$r(M, G) = |M(G)|/|M(H)|$$

When the ratio for a motif is larger than 1, the probability of this motif M appearing in the real-world graphs is larger than the probability of M appearing in the random graph.

4.2.2 Random multilayer temporal network models

We use three datasets (further discussed in Section 5) for which we must generate ten random networks of approximately the same size. This means the random networks will have approximately the same number of nodes, static edges and parallel edges (unique edges with the same source and target node). The graphs were created with the networkx (`https://networkx.github.io/`) package for Python. The graph was constructed according to the Barabási-Albert preferential attachment model [BA99]. For every real-world network the distribution of the parallel edges was determined. This distributions shows how many static edges have a particular number of parallel edges. Then this distribution was fitted in a model and according to this model the distribution of the parallel edges in the random network was made. In Figure 7 the distribution with the fitted model for one of the used datasets is shown.

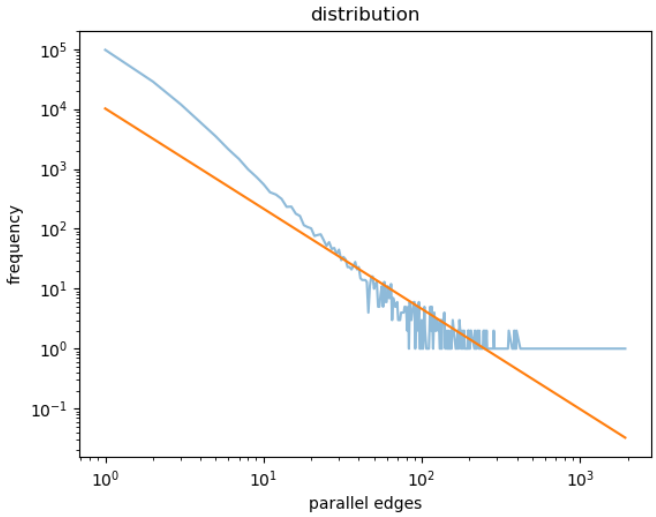

Figure 7: Distribution of number of parallel edges of the MathOverflow network and the fitted model for the number of parallel edges.

For each static edge and its parallel edges a random timestamp was given. Just as the timestamps the layers were also assigned randomly according to a uniform distribution. This means the timestamps and layers are not generated according to a model as is done with the parallel edges.

4.3 Implementation

In Section 4.1 some functionality of the tool was already discussed. The complete list of functional requirements for the tool is listed below:

1. The tool must be able to call the command line tool.
2. The tool must be able to read the outputfile from the command line tool.
3. The tool must be able to hold information of every input for each user.

4. A user must be able to upload a network in a simple way.
5. The tool must be able to make a heatmap for every result of the multilayer temporal motif counting algorithm.
6. The tool must be able to visualize every motif type and layer permutation whenever requested by the user.
7. The tool must be able to show the motif count, motif type number and layer permutation number whenever requested by the user.

Because the tool needs access to a command line and then be able to create visuals in a GUI, the Flask web application framework (`https://flask.palletsprojects.com/en/1.1.x/`) is chosen for Python. The Flask framework uses a Python backend and an HTML based frontend which can be used for a good and simple GUI. Another reason for picking the Flask framework for Python is that we want the tool to be cross-platform, meaning we want compatibility with as many devices as possible, with different operating systems. Making the tool a web-based application gives it the possibility to be run on a server which is then accessible through the web. The Python backend allows for making calls to the command line which fulfils the first requirement.

To fulfil the second and third requirement a session is created for each user with only the information of this user's input. The session data includes the filename of the file where the output will be stored, which will be used to read the file when the algorithm is done. The session includes the following data for every upload made:

- The original filename of the network when it was uploaded.
- The value of delta (δ).
- The number of layers of the network.
- The original filename of the network extended with the time of upload (this is the name of the saved file in the directory).
- The filename of the corresponding outputfile.
- The time at which the file was uploaded.
- The formatted data from the result of the counting algorithm.

For the fourth requirement we make use of the HTML based frontend. The upload page of the tool allows for a simple upload of the network file. And is shown in Figure 8. The network file must consist of lines in the well-known edge list format: *source target timestamp layer*. These values must be numeric and whitespace-separated. The tool must check if this format is correct for the uploaded file. It must also check if the value of delta and number of layers are filled in when uploaded. When one of these requirements is not satisfied, the upload must be done again, and the user is informed.

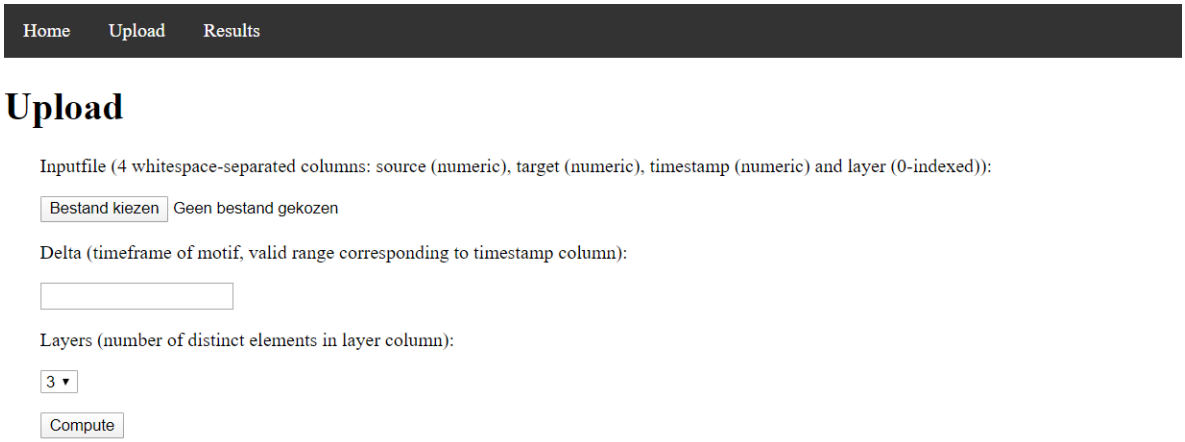

Figure 8: Upload page of the tool.

The last three requirements are realized through the Plotly graphic library for JavaScript (`https://plotly.com/`). The HTML based frontend allows for Javascript which can take data from the backend, in this case the data for every upload. For every upload a heatmap is made using Plotly. If the data is not yet available but an upload is made, the tool will refresh the page automatically every 5 seconds until the data is available. The heatmap shows for every cell in the heatmap the layer permutation and motif type as illustrations when clicked. The layer permutation number, motif type number and motif count are also displayed when a cell is clicked or hovered. The result page with an example upload is shown in Figure 9.

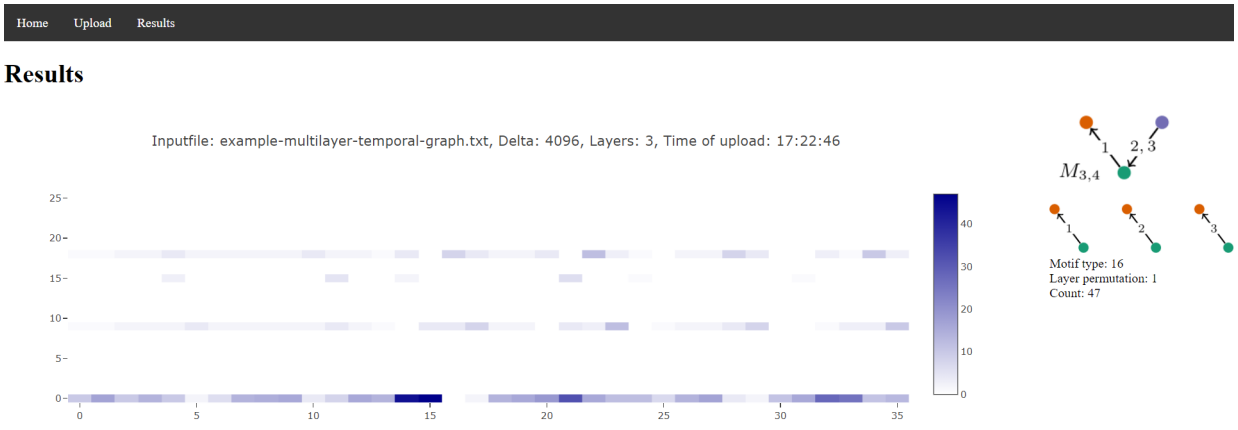

Figure 9: Screenshot of result page with heatmap of motif counts.

5 Experiments

Experiments are performed to answer the research questions introduced in Section 1. We investigate in particular the differences between the motif counting results of real-world and random networks.

5.1 Data

In this section the datasets that are used for the experiments are discussed. This data was used in [PBL17] and originally represented a non-layered temporal network. It was transformed to fit the purpose of this research and thus layers were added. In Table 1 the summary statistics of the network datasets are shown. Below every real-world dataset we list the statistics of one of the corresponding random networks to give insight in the approximate sizes of the random networks. Some of the randomly generated graphs have more temporal edges than the corresponding real-world graphs and some have less. This is because of the distribution of the parallel edges in the random graph, which is based on a model of the real-world graph. Because it follows a model the number of temporal edges could be less or more than in the real-world graph.

Dataset	**Nodes**	**Temporal edges**	**Static edges**	**Layers**
Ask Ubuntu	117 101	684 335	354 723	2
Ask Ubuntu Random	120 000	892 531	359 991	2
Math Overflow	21 190	398 969	156 987	2
Math Overflow Random	22 000	320 827	153 951	2
Super User	144 512	1 013 306	548 269	2
Super User Random	145 000	1 345 468	579 984	2

Table 1: Statistics of different datasets.

Ask Ubuntu, Math Overflow and Super User are forums were users can post questions and other users can answer them. Furthermore users can comment on questions or answers. We now define the networks of these forums by deriving edges *(u,v,t,l)* where at time t and $l = 0$ a user u comments on an answer given by user v and at $l = 1$ a user u comments on a question posted by user v. These datasets are previously studied in [PBL17].

5.2 Results

In Figures 10, 11 and 12 the resulting heatmaps of the real-world Ask Ubuntu, Math Overflow and Super User datasets are presented, each with one of the random dataset's heatmap as an example. The networks were given a delta of 604 800 seconds (one week). We have chosen this delta value because we estimate that questions are often answered and discussed within a time frame of no more than a week. Note that the horizontal axis denotes the motif type (the horizontal

axis list the motif types from 1 to 36, this corresponds to the motifs from Figure 2 numbered from left to right and from top to bottom) and the vertical axis denotes the layer permutation. In the heatmaps for the random networks it is visible that every layer permutation for each motif type has approximately the same count. This is likely due to the fact that layers are assigned randomly according to a uniform distribution to each edge in this network.

In Tables 2, 3 and 4 ten particular motifs are chosen from each dataset, and for each of them the ratio is calculated. The ratio is calculated with the average count of the ten randomly generated networks for each motif, the standard deviation of this average is also stated in the tables. These motifs were chosen because they all had a much higher count than the other motifs in the same network. All motifs in the table have a very high ratio well above 1 and thus their probability to appear in the real-world network is higher than appearing in the random graph. This means that these motifs can be labeled as significant compared to random graph. Also because of the dominant counts relative to the other motifs in the heatmap these motifs describe the interactions of this network well. However, the Math Overflow heatmap has a less dominant distribution of the counts than the Ask Ubuntu and Super User heatmaps. The motifs in Table 3 are still the most dominant of their network but are less dominant than the motifs in the tables of the other two datasets.

Motif types 1, 5, 6, 11, 12, 28, 36 ($M_{1,1}$, $M_{1,5}$, $M_{1,6}$, $M_{2,5}$, $M_{2,6}$, $M_{5,4}$,$M_{6,6}$ in Figure 2, respectively) occur in two or all three of the tables with layer permutation 1. This means that all the edges in these motifs are comments to answers. Motifs 1, 6 and 36 denote that multiple users comment on the answer of another which could indicate that these comments are additions to the given answers or critique on the initial answer. Whereas Motifs 5, 11, 12 and 28 would indicate more of a discussion about the answer rather than additions or critique. Furthermore motif types 27 and 29 ($M_{5,3}$ and $M_{5,5}$ respectively in Figure 2) occur in two tables with layer permutation 5. In this layer permutation the first two edges denote a comment to an answer and the third edge denotes a comment to a question. These motifs could indicate that after a discussion of two users a comment was given on the question stating something wrong with the questions or asking for clarification. In general the most frequently appearing edges are comments to answers. In the motif counts that are dominant two or three of the edges are comments to answers. This indicates that there is a lot of discussion about the answers but not so much about the questions themselves.

The Ask Ubuntu and Super User heatmaps share a common distribution of dominant motifs. This is likely due to the fact that they both are question-and-answer platforms. However, as stated earlier the heatmap of the Math Overflow dataset has less of these dominant motifs but shares the characteristic of being a question-and-answer platform. This may be because the dataset is significantly smaller than the other two datasets.

When we compare the heatmaps from the real-world datasets with the ones of the randomly generated ones we get an interesting result. The motif types 25, 26, 31 and 32 ($M_{5,1}$, $M_{5,2}$, $M_{6,1}$, $M_{6,2}$ respectively in Figure 2) are dominant in all of the randomly generated heatmaps. These heatmaps are from one of the ten random datasets for each real-world dataset but the other nine always have the same dominant motifs. This is interesting because in the real-world networks these motifs are some of the least occurring ones. This is likely because the random model assumes a number of parallel edges between users but does not assume a third user to "participate" in this

discussion and thus a third edge in the motif must occur at random. This may cause the dominance of two node motifs in the random networks. In the real-world networks discussions on answers are occurring often with more than two users and thus the two node motifs do not occur as often.

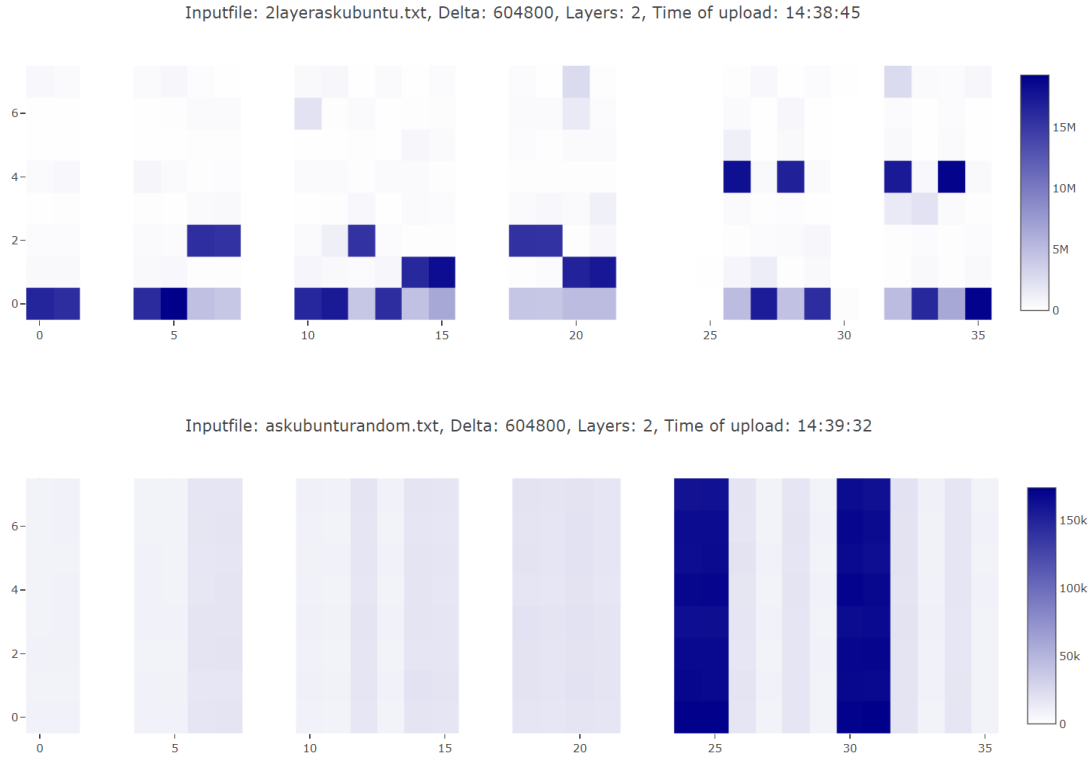

Figure 10: Heatmaps of motif counts of the real-world Ask Ubuntu dataset (top) and a randomly generated network (bottom).

Motif type	**Layer permutation**	**Real-world count**	**Count avg. in random**	**Std. dev. in random**	**Ratio**
1	1	16 528 089	7340	1772	2251.78
6	1	19 267 902	7519	1801	2562.56
12	1	17 325 774	7455	1590	2324.05
28	1	17 115 937	7439	1726	2300.84
36	1	19 148 382	7357	1821	2602.74
15	2	16 226 377	14 783	4090	1097.64
8	3	15 500 597	14 842	4129	1044.37
20	3	15 510 620	14 548	3965	1066.17
27	5	18 141 234	14 627	3989	1240.26
29	5	16 877 724	14 941	4172	1129.62

Table 2: Motifs from the Ask Ubuntu network with their ratio.

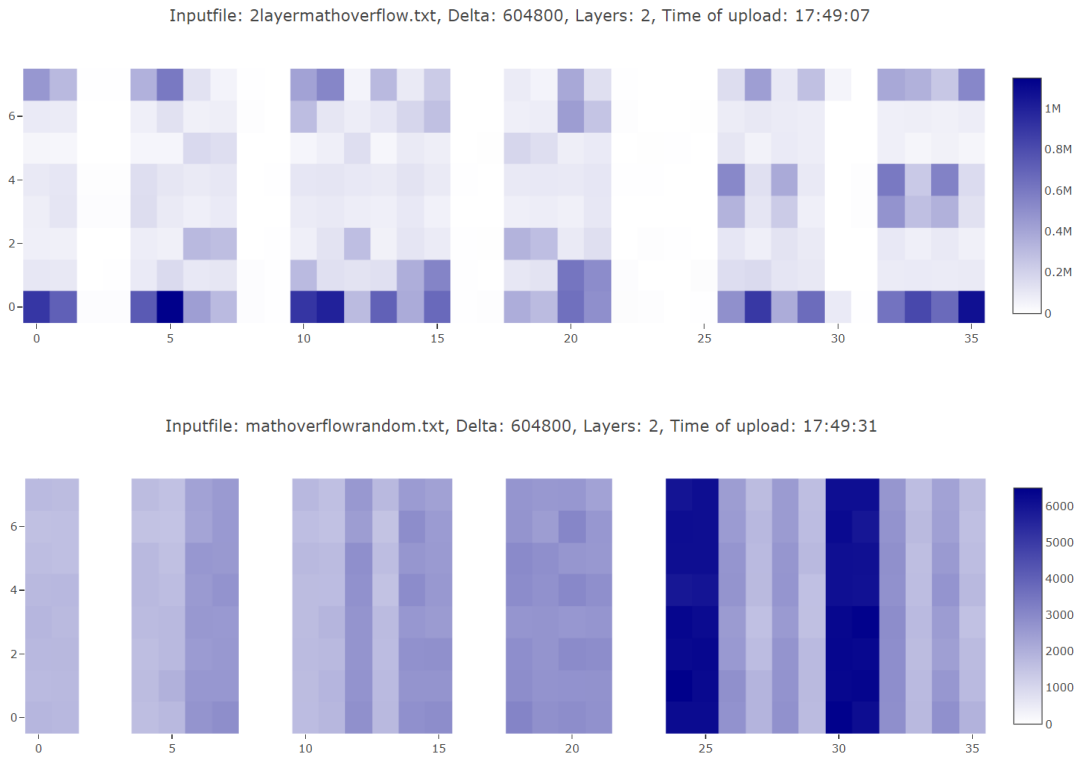

Figure 11: Heatmaps of motif counts of the real-world Math Overflow dataset (top) and a randomly generated network (bottom).

Motif type	**Layer permutation**	**Real-world count**	**Count avg. in random**	**Std. dev. in random**	**Ratio**
1	1	903 138	2124	700	425.21
2	1	713 491	2207	720	323.29
5	1	742 197	2174	711	341.40
6	1	1 147 369	2065	632	555.63
11	1	907 504	2248	727	403.69
12	1	1 000 156	2246	705	445.31
28	1	896 424	2222	772	403.43
36	1	1 076 648	2126	764	506.42
33	5	601 496	4382	2025	137.27
35	5	560 655	3976	2093	141.01

Table 3: Motifs from the Math Overflow network with their ratio.

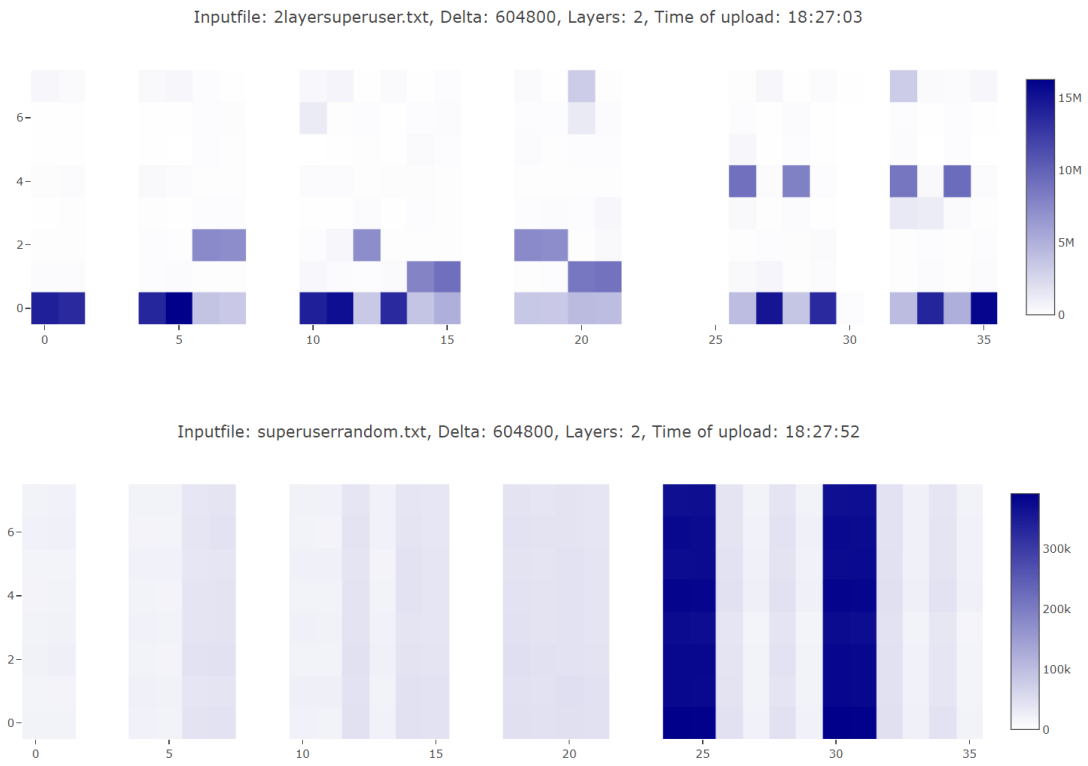

Figure 12: Heatmaps of motif counts of the real-world Super User dataset (top) and a randomly generated network (bottom).

Motif type	**Layer permutation**	**Real-world count**	**Count avg. in random**	**Std. dev. in random**	**Ratio**
1	1	14 266 649	14 316	5575	996.55
5	1	13 904 289	14 473	5220	960.71
6	1	16 242 391	14 392	5630	1128.57
11	1	14 289 682	14 849	5184	962.33
12	1	15 329 117	14 492	5234	1057.76
14	1	13 596 562	14 565	5282	933.51
28	1	14 929 865	14 614	5458	1021.61
30	1	13 565 996	14 552	5106	932.24
27	5	9 134 186	30 700	13 237	297.53
29	5	8 064 613	30 479	12 868	264.60

Table 4: Motifs from the Super User network with their ratio.

6 Conclusions and future research

In order to understand the complex interactions between entities in networks we study motifs. These motifs are small subgraphs and their significance in networks is often denoted by the number of times they occur in a network. Counting these motifs is done by high level algorithms but have the downside of not being user friendly and the usage is limited. This thesis aimed to provide a solution for this by answering two research questions, namely:

1.Can a user friendly tool be built that allows for thorough analysis of the results from multilayer temporal motif counting algorithms?

2.What insights can be obtained from multilayer temporal motif count results when we compare real-world data and random data?

The first research question was answered by the tool that was constructed. The tool maps the result of the multilayer temporal motif algortihm to an interactive heatmap. This heatmap displays the motif type, layer permutation and the resulting count. Additionally, whenever a cell is clicked, it displays the motif type of that cell as an image, including the layer permutation for each of the three edges. This tool can help to give insight on how the nodes of networks interact with each other and allows for thorough investigation of multilayer temporal networks.

The second question was answered by analyzing real-world network datasets, calculating the ratio of motifs. The ratio indicates the probability of a motif occurring in a real-world network as opposed to a random one. Some of these motif ratios were well above 1 and thus can be labeled as significant. These significant motifs can describe the interactions between entities in the studied networks. We tried to explain some of the significant motifs found in the network by looking at the type of each edge and the corresponding motif type. We found that answers are more discussed than the corresponding questions in the question-and-answer forums. We also found that the motifs with two nodes are dominant in the randomly generated network while this is not the case in the real-world networks.

The proposed tool helps to visualize multilayer temporal motif counting results. Not only for one network, but also to compare between networks. Future research could go deeper into how these differences in interactions between nodes come to exist. What was seen in our experiments is that two of the real-world datasets had similar dominant motifs and were both question-and-answer forums. Future research could find other networks of the same type (messaging or idea-sharing type of platforms, for example) and study whether or not the networks share the same dominant motifs. By comparing different networks with the same type of communication it could shed new light on user behaviour.

References

[BA99] Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *Science*, 286(5439):509–512, 1999.

[Bar16] Albert-László Barabási. *Network Science*. Cambridge University Press, 2016.

[BKT19] Hanjo D. Boekhout, Walter A. Kosters, and Frank W. Takes. Efficiently counting complex multilayer temporal motifs in large-scale networks. *Computational Social Networks*, 6(8), 2019.

[GGY+14] M. Gupta, J. Gao, X. Yan, H. Cam, and J. Han. Top-k interesting subgraph discovery in information networks. In *Proceedings of the IEEE International Conference on Data Engineering*, pages 820–831, 2014.

[HS12] Petter Holme and Jari Saramäki. Temporal networks. *Physics Reports*, 519:97–125, 2012.

[KAB+14] Mikko Kivelä, Alex Arenas, Marc Barthelemy, James P. Gleeson, Yamir Moreno, and Mason A. Porter. Multilayer networks. *Journal of Complex Networks*, 2(3):203–271, 2014.

[KK01] M. Kuramochi and G. Karypis. Frequent subgraph discovery. In *Proceedings of the IEEE International Conference on Data Mining*, pages 313–320, 2001.

[KKK+11] Lauri Kovanen, Márton Karsai, Kimmo Kaski, János Kertész, and Jari Saramäki. Temporal motifs in time-dependent networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2011(11):P11005, 2011.

[MSoI+02] R. Milo, S. Shen-orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon. Network motifs: Simple building blocks of complex networks. *Science*, 298(5594):824–827, 2002.

[PBL17] Ashwin Paranjape, Austin R. Benson, and Jure Leskovec. Motifs in temporal networks. In *Proceedings of the ACM International Conference on Web Search and Data Mining*, pages 601–610, 2017.

[TKWH18] Frank W. Takes, Walter A. Kosters, Boyd Witte, and Eelke M. Heemskerk. Multiplex network motifs as building blocks of corporate networks. *Applied Network Science*, 3(39), 2018.