IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 26 September 2022, accepted 9 October 2022, date of publication 13 October 2022, date of current version 20 October 2022.

*Digital Object Identifier 10.1109/ACCESS.2022.3214543*

RESEARCH ARTICLE

# Event Detection via Tracking the Change in Community Structure and Communication Trends

**RIZA AKTUNC, PINAR KARAGOZ, (Member, IEEE), AND ISMAIL HAKKI TOROSLU**
Department of Computer Engineering, Middle East Technical University, 06800 Ankara, Turkey

Corresponding author: Riza Aktunc (riza.aktunc@ceng.metu.edu.tr)

**ABSTRACT** Event detection is a popular research problem aiming to detect events from various data sources, such as news texts, social media postings or social interaction patterns. In this work, event detection is studied on social interaction and communication data via tracking changes in *community structure* and *communication trends*. With this aim, various community structure and communication trend based event detection methods are proposed. Additionally, a new strategy called *community size range based change tracking* is presented such that the proposed algorithms can focus on communities with different size ranges, and considerable time efficiency can be obtained. The event detection performance of the proposed methods is analyzed using a set of real world and benchmark data sets in comparison to previous solutions in the literature. The experiments show that the proposed methods have higher event detection accuracy than the baseline methods. Additionally, their scalability is presented through analysis by using high volume of communication data. Among the proposed methods, CN-NEW, which is a community structure based method, performs the best on the overall. The proposed communication trend based methods perform better mostly on communication data sets (such as CDR), whereas community structure based methods tend to perform better on social media-based data sets.

**INDEX TERMS** Change tracking, communication trends, community detection, community structure, ensemble model, event detection, network features, temporal network.

## I. INTRODUCTION

Event detection has been a trending research topic [6], [24], [27], [38] since it can be useful for a variety of domains such as planning emergency actions for disasters, managing urban transportation effectively [29], and getting the trending and up-to-date news such as financial changes [30]. *Event* refers to a happening that takes place at a certain time and at a certain location, and attracts people's attention. One can find variations of this definition depending on the domain, such as event can happen at a time period instead of a time instance, or place can be multiple or even virtual. But in all definitions, event is considered as an interesting happening and hence affects people's situation and also behavior.

Events can be detected from a variety of resources. One of the most popular data resources is textual contents in the social media and the web [5], [16], [24], [25], [38], such that events trigger social posts and provide a rich resource to detect the events in a much more rapid way than traditional news media. Another valuable resource for event detection is the social interaction and communication structures and their evolution through time. Changes in social communications can indicate *events* since such happenings can trigger interaction among individuals out of daily and ordinary routines. One of the approaches for event detection from communications is based on modeling communication traces as temporal graphs and analyzing the changes through graph features [13], [17], [27], [35].

Recently, event detection studies analyzing the change in the community structure [2], [3] and communication trend [20] on the communication graphs have produced effective solutions. They offer an alternative to feature analysis on communication graphs, and focus on the structural changes on the graph. In [2] and [3], event detection problem is modeled as change detection in the community structures extracted from a given communication network. The

The associate editor coordinating the review of this manuscript and approving it for publication was Anandakumar Haldorai.

communities are determined for predefined time intervals, such as daily or weekly communication traces, and the striking changes within the detected communities are marked as event. Community detection is also a popular research problem, and there is already variety of successful methods in the literature, such as SLM [36], dSLM [4], Louvain [7], InfoMap [28], and more advanced models [31], [32], [33], which can be used for community detection based event detection solutions. On the other hand, in [20], an event is modeled as a strong change of communication patterns within the communication network. The communities are detected once at the beginning of the time span and the changes in inter-intra community communications are tracked at each time interval. Although both approaches provide promising results, there are still plenty of possibilities for further improvement and there is a need for a thorough and comparative analysis on a variety of data sets.

In order to fill in this gap, this work proposes new concepts and approaches, as well as improvements over previous community structure and communication trend based event detection approaches. For all the proposed solutions, a communication graph, which is constructed from communication or social interaction data, and evolution of such graph along the timeline constitute the basis. For each time interval, a communication graph is generated such that nodes represent the users and edges represent the communications or interactions between the users. In the community structure based solutions, at each time interval, communities within the graph are extracted and tracked, whereas in communication trend based solutions, graph of each time interval is considered as a whole and compared against the initial graph constructed at the beginning of the timeline. In this setting, the novelty and contributions provided by this study are as follows:

- For community structure based event detection, determining the central nodes of the communities and tracking the change in terms of central nodes is shown to be a promising approach in the previous studies [2], [3], however in such earlier studies, node centrality measurement is not performed with well-structured graph centrality metrics. In this study, a new community structure based event detection approach is proposed, such that *central nodes* of a community are determined with PageRank algorithm. By this way, the changes in the central nodes are tracked in terms of several indicators, such as inclusion and drop of central nodes from communities. Changes in such indicators are used for detecting events.
- In community structure based event detection, this study hypothesizes that considering all sizes of communities might prevent detecting events effectively since there are very small (with one or two members) and very large size communities in the daily routine. To overcome this problem, the concept of *community size range based change tracking* is proposed, such that, under the community size range parameters, a method can focus on certain sizes of communities. The experiments show that this mechanism can provide increase in event detection accuracy as well as time efficiency for some of the cases.
- The resolution of the time intervals (i.e., time windows) is a crucial parameter in analysis environment. The effect of time resolution (such as daily, 2-day, 3-day, etc.) on event detection performance for data sets of varying nature has been experimentally analyzed. The experiments reveal that relaxing the time resolution to a certain limit positively affects the accuracy and time efficiency, whereas the event detection performance deteriorates when the resolution is relaxed too much.
- Two novel approaches are proposed for constructing ensemble models. The first one combines the variations of the method that tracks the number of communities with different community size ranges. The second ensemble approach combines the methods tracking the number of communities with the methods tracking the central nodes.
- Various change tracking techniques using the *modularity* of the graphs are proposed, as alternatives to the communication trend based method in [20] that tracks the change in inter-intra communication ratio. Modularity is a function that gives higher values for graphs, as the community structures get denser with sparser interactions between the communities. According to the conducted experiments, in general, the proposed change tracking techniques are more effective.
- A comprehensive set of experiments are conducted comparing the proposed and previous event detection approaches over data sets representing different forms of communications including CDR (Turkish GSM Operator and Reality Mining), network of social media posts (Boston Bombing) and e-mail communication (Enron) data sets. The performance analyses are conducted in terms of accuracy, time efficiency and scalability, aiming to determine which method to prefer under which communication data characteristics. According to the experiments, proposed communication trend based methods tend to perform better than the competitors on CDR based data sets, whereas proposed community structure based methods tend to perform better on social media-based data sets. In general, community structure based methods have lower execution time and better scalability than the other methods.

This paper is organized as follows. In Section II, the community and event detection related concepts that are used in the paper are explained. In Section III, the proposed event detection methods by tracking the change in community structure and communication trend are described. In Section IV, the data sets and the results of the experiments are presented and discussed. In Section V, related studies are summarized and finally the paper is concluded with an overview in Section VI.

## II. PRELIMINARIES

In this section, the basic concepts that are used to introduce the problem and the proposed methods are defined.

*Definition 1 (Temporal Graphs):* The graphs that change over time are defined as temporal graphs. In our context, both nodes and edges can change (added or deleted) over discrete sequential time steps. Thus, temporal graph can be expressed as a sequence of graphs such as $< G_1, G_2, \ldots, G_t >$.

*Definition 2 (Resolution):* The time window length used to construct the temporal graphs is defined as the resolution. The resolution can be chosen according to the data set and the requirements of the event detection problem. For instance, if the resolution is set as 1 day, then each graph instance in the temporal graph corresponds to communications per day.

*Definition 3 (Community):* Although there is no universally accepted definition, in general, community in a graph is defined as a set of nodes that have links with each other more than they have with the other nodes of the network [15], [23], [34].

*Definition 4 (Modularity):* Modularity is a measure about the community structure of a graph. In order to compute the modularity value of a network, firstly, a null model of that network should be generated. A null model of a network can be constructed by removing the existing edges, and replacing them with the new ones randomly while maintaining the original degrees of the nodes. For a given network, modularity is defined as the fraction of the difference between the total number of internal edges in the real network and the total number of internal edges in its null model, to the total number of the edges in the whole network. It is calculated as given in Equation 1. The measure is motivated from the idea that a randomly created network is not expected to have a good community structure. Thus, as the real network differs from its null model, its community structure is considered to be stronger [15], [21].

$$Q = \frac{1}{2e} \sum_{ij} (O_{ij} - N_{ij}) \delta(C_i, C_j) \tag{1}$$

In the equation, $O_{ij}$ is the number of edges between node $i$ and node $j$ in the original network and $N_{ij}$ is the number of edges between node $i$ and node $j$ in the null model. The $\delta$ function ensures taking only internal edges (i.e., edges within a community) into account in the summation by generating 1 if node $i$ and node $j$ are in the same community ($C_i = C_j$), and it returns 0 otherwise. The summation iterates over all pairs of nodes of the network. Finally, $e$ represents the total number of edges of the network. Note that, the number of nodes and the number of edges are the same for both the original network and its null model, since null model preserves the nodes and their degrees.

*Definition 5 (Initial Network (IN)):* Given a sequence of time-stamped communications, *initial network* is the graph constructed for the initial time window ($G_1$) with respect to the given resolution.

*Definition 6 (Inter-Intra Links Ratio (IILR)):* For a given set of communities, inter-intra link ratio is defined as the ratio of the difference between the number of inter community links and the number of intra community links to the total number of links in the full network. The calculation of the number of inter community links, the number of intra community links and IILR are given in Equations 2, 3, and 4, respectively. In the equations, $O_{ij}$, $C_i$, $C_j$, and $\delta$ mean the same as in Equation 1. The $\omega$ function returns 0 if node $i$ and node $j$ are in the same community ($C_i = C_j$), and, 1 otherwise. This definition is used in [20] to detect events that trigger information flow changes across communities.

$$G_{inter} = \sum_{ij} O_{ij} \omega(C_i, C_j) \tag{2}$$

$$G_{intra} = \sum_{ij} O_{ij} \delta(C_i, C_j) \tag{3}$$

$$IILR = \frac{G_{inter} - G_{intra}}{G_{inter} + G_{intra}} \tag{4}$$

*Definition 7 (Central Nodes (CN)):* Central nodes of a community are the nodes whose centrality scores are higher than the average centrality score within the network. In our study, PageRank [9] is used as the centrality metric.

*Definition 8 (Bucket (Community Size Range)):* In determining the change in the community structures, taking all of the communities into account may lead to incorrect predictions since events may only affect communities of certain size. Additionally, for large data sets, using only the communities of certain sizes may positively affect execution time considerably compared to using all communities for change detection. With this motivation, the following groups (buckets) of communities are defined with respect to their sizes:

- Bucket A (all) includes all communities without any filtering.
- Bucket F (filtered) excludes the very small communities (of size one and two nodes) and very large ones and includes the rest of the communities. The filtered large communities contain less than 0.001 % of the total nodes.
- Bucket Q1 includes the first quarter of the filtered communities (Bucket F) (initial 25% partition) with respect to community distribution with respect to size.
- Bucket Q2 includes the second quarter of the filtered communities (the second 25% partition).
- Bucket Q3 includes the third quarter of the filtered communities (the third 25% partition).
- Bucket Q4 includes the last quarter of the filtered communities (the fourth 25% partition).
- Bucket H1 includes the first half of the filtered communities (initial 50% partition).
- Bucket H2 includes the second half of the filtered communities (the second 50% partition).

Note that H1 = Q1 ∪ Q2, H2 = Q3 ∪ Q4, and F = H1 ∪ H2. When change detection is applied for a given bucket, only the nodes of the communities in the bucket are considered.

*Definition 9 (Event):* This study considers that an event is expressed with a signal (reflection) in its corresponding time interval.

*Definition 10 (Event Detection):* It is aimed to determine the *time intervals that include event*. Hence, it is hypothesized that the change in the community structure and community communication trend within communication network provide signals (reflections) of events. Different ways to determine the change in community structure and community communication trend are elaborated.

## III. EVENT DETECTION METHODS

This study considers that a significant change in the community structure or the communication trend between the graphs of consecutive time steps indicates an event. To this aim, several basic and ensemble methods are proposed to track the changes in the graphs in order to detect events. The overview of the proposed approach's architecture is presented in Figure 1. As given in the figure, for the proposed methods, the event detection pipeline is composed of three basic steps: pre-processing, community detection and event detection.

#### 1) PRE-PROCESSING & GRAPH GENERATION

For the proposed methods, the communication data is represented as a graph. To analyze the effect of graph type, directed, undirected, weighted, unweighted graphs are constructed from raw data sets. Since the nature of raw data sets differ from each other, different processes must be applied to form these graphs.

#### 2) COMMUNITY DETECTION

As the community detection technique, the dSLM algorithm [4] is used for all data sets. The InfoMap community detection method is also used to compare our proposed methods with the method proposed in [20].

#### 3) EVENT DETECTION

Basically, there are two different kinds of change structures tracked in our study. In the first one, changes in the community structure are tracked. Also, ensembles of various community structure change methods are studied. As the second main tracking method, the changes in the communication trends among the communities are analyzed.

### A. PRE-PROCESSING & GRAPH GENERATION

Since the nature of the raw data sets differs from each other, pre-processing and graph generation step also includes different sub-tasks for different data sets. The details for each of the data sets used in the study are as follows:

1) **Enron** This well-known data set is a collection of email records of the Enron company collected between 1999-01-01 and 2002-04-30. It is structured as a network such that nodes represent the employees and edges represent the emails sent/received between employees. The number of exchanged emails is used as the weight of the edge. Therefore, a directed weighted network is obtained. For this data set, time interval granularity set as one week.
2) **Boston Bombing Tweets** This data set contains the networks of retweets and mentions on bombing event happened during the Boston marathon in April 2013. The data set is partitioned daily, thus 30 networks for each of the retweets and mentions collection are generated. The generated networks are directed and weighted where weights are the mention and retweet counts, respectively.
3) **Reality Mining** This data set is a Call Detail Record (CDR) collection including date of the call, caller id and call receiver id. The time interval is set as a week since the ground truth is provided over weeks. In the constructed graphs, nodes represent IDs of the participants, and the edges represent the calls between them. The weight of an edge represents the number of calls made between the participants. The data set contains both SMS and Voice Call collections. For both, directed weighted, directed unweighted and undirected weighted networks are generated, resulting with 6 networks in total.
4) **Turkish GSM Operator** This data set is also a CDR data, including the records of GPRS, SMS and voice calls made in a large metropolitan city in Turkey in September 2012. The voice call records are used in this study. The data set is partitioned into days, and a directed weighted graph is constructed for each partition. In the constructed graphs, nodes represent the users, the edges represent the calls. The weight of an edge represents the number of calls made between the users.

### B. COMMUNITY DETECTION

In the literature, there are several community detection algorithms [4], [23], [28]. In this work, dSLM [4] and InfoMap [28] are used as the community detection methods.

#### 1) DYNAMIC MODULARITY OPTIMIZER (dSLM)

Modularity is a metric generally used for evaluating the quality of community detection algorithms. It is a function that outputs higher values if a given network can be partitioned better as communities. In some of the community detection algorithms it is also used as the objective to be maximized. dSLM [4] is the dynamic and faster version of SLM algorithm [36] which aims to determine communities by optimizing the modularity values of networks. It optimizes modularity by applying the following steps [36]:

1) Assign each node as a singleton community (i.e., each node is a community itself).
2) For each node and its neighbours, in random order, move the node from its own community to its neighbor's community and recalculate the modularity value. If it is increased, keep the change, else, revert the

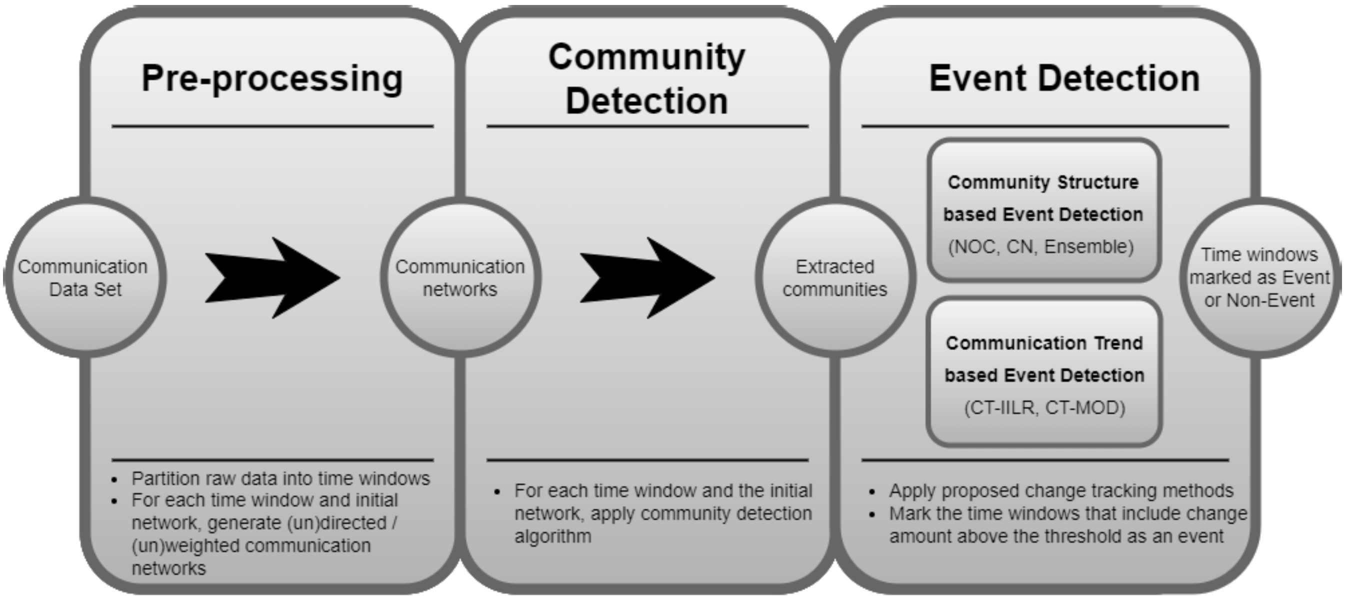


**FIGURE 1.** **The overview of the method.**

movement. This step is called as local moving heuristic since it heuristically tries to improve the modularity value by moving the nodes among communities.

3) After step 2 is completed, mark each community as as a subnetwork and apply step 1 and 2 to each one of the subnetworks. At the end of this step, a set of subnetworks, which have its own set of communities are generated.
4) Then, this structure is converted into a new (reduced) graph where each community is a node, and, each subnetwork is a community.
5) Apply steps 2, 3, and 4 to this reduced network recursively until the network that cannot be reduced further.

#### 2) MAP EQUATION OPTIMIZER (InfoMap)

The InfoMap algorithm detects the communities by minimizing the map equation value of the given network. InfoMap detects community structure by minimizing the description length in the map equation. Since it is infeasible to check all possible partitionings and choose the one that minimizes the map equation value, it uses heuristic search to find the near optimum structure [28].

The map equation formula that InfoMap minimizes is given in Equation 5.

$$L(M) = q_{\curvearrowright} H(\theta) + \sum_{i=1}^{m} p^{i}_{\circlearrowright} H(\alpha^{i}) \tag{5}$$

In the equation, M is a proposed community structure of the network such that each node belongs to a community. L(M) is the average length to describe an infinite random walk on the network whose community structure is M.

The map equation calculates the minimum description length of a random walk on the network based on the partition M. The first part of the equation gives the average number of bits necessary to describe movement between communities, and the second part determines the average number of bits necessary to describe movement within communities [28].

### C. EVENT DETECTION METHODS BY TRACKING THE CHANGE IN THE COMMUNITY STRUCTURES

In order to track the change in community structure between two consecutive time intervals, the following two indicators are considered: change in the number of communities and change in the central nodes of communities.

**Algorithm 1** Event Detection via Change on # of Comm

**Require:** setOfComm, bucket, threshold
**Ensure:** events
  prev ← NOC(setOfComm($t_1$), bucket)
  **for** i ← 2 **to** timeWindowCount **do**
    cur ← NOC(setOfComm($t_i$), bucket)
    change ← abs(cur − prev)/prev
    **if** change ≥ threshold **then**
      add $i$ to events
    **end if**
    prev ← cur
  **end for**
  **return** events

#### 1) TRACKING THE CHANGE IN THE NUMBER OF COMMUNITIES

The focus of this method is determining the change in the number of communities generated for consecutive time windows. The algorithm for the method is given in Algorithm 1. It takes set of communities, bucket, and event detection threshold values as parameters. For each time window, the algorithm finds the number of communities that are in the range of given bucket. Then, the change ratio in the number of communities is calculated by taking the absolute difference between the number of communities in the previous time step and the current time step and dividing the result by the number of communities in the previous time step. If a time window's change ratio is greater than or equal to the given event detection threshold value, that time window is marked to include an event.

#### 2) TRACKING THE CHANGE IN THE CENTRAL NODES

In this method, the focus is on the central nodes of the communities within the communication network and their change across time windows. The algorithm for this method is presented in Algorithm 2. Similar to the previous one, it takes set of communities, change detection method, bucket, and event detection threshold values as input parameters. For each time window, the algorithm finds the number of communities that are in the range of given bucket; furthermore, the central nodes of the communities are also determined. Based on the given change detection method, the algorithm can find change ratios in four different ways:

1) **SIZE:** If the given change detection method is based on Size, the change ratio is calculated according to the difference between the number of central nodes in the previous time window and the number of central nodes in the current time window.
2) **NOT_ANY_MORE:** If the given change detection method is Not Any More, the change ratio is calculated by computing the number of central nodes of previous time step that are not any more central nodes in the current time step.
3) **NEW:** If the given change detection method is New, the change ratio is calculated by computing the number of newly introduced central nodes in the current time window.
4) **NOT_ANY_MORE_NEW:** If the given change detection method is Not Any More and New, the change ratio is calculated by computing the number of central nodes of previous time step that are not anymore central nodes in the current time step added up with the number of newly introduced central nodes in the current time window.

As in the previous algorithm, time window is marked to include an event, if a time window's change ratio is greater than or equal to the given event detection threshold value.

#### 3) ENSEMBLE OF TRACKING THE CHANGE IN THE NUMBER OF COMMUNITIES METHOD WITH DIFFERENT BUCKET VARIATIONS

Normally, Algorithm 1 can be applied to any chosen bucket. Some events may affect more than one bucket. To detect such cases, an ensemble algorithm that combines change tracking of number of communities for different buckets is generated.

**Algorithm 2** Event Detection via Change On Central Nodes

**Require:** setOfComm, changeMethod, bucket, threshold
**Ensure:** events

```
prev ← setOfComm(t_1).centralNodes
for i=2 to timeWindowCount do
    cur ← setOfComm(t_i).centralNodes
    change ← comp(changeMethod, prev, cur)
    if change ≥ threshold then
        add i to events
    end if
    prev ← cur
end for
return events
```

For example, an event may affect the communities in the buckets Q1 and Q3. Then, an ensemble of change tracking for bucket Q1 and bucket Q3 can be constructed. At this point, there are two alternative ways to determine the final decision of the ensemble: ANDing vs ORing the results from individual event detectors. Since all kinds of communication data can be processed to generate buckets, this ensemble approach can be used very easily regardless of the data size or structure.

#### 4) ENSEMBLE OF TRACKING THE CHANGE IN THE NUMBER OF COMMUNITIES AND TRACKING THE CHANGE IN THE CENTRAL NODES

Sometimes both number of communities and central nodes are affected from the event. Therefore, as a natural extension, this study generates an ensemble combining two basic methods, one tacking the number of communities, and the other tracking the central nodes. As in the previous ensemble method, in this one, the final decision can be determined through logical operations of ANDing and ORing the results of the individual event detectors. Similar to the above approach, it is also possible to determine changes in community structures between consecutive time steps to all kinds of data sizes and structures.

### *D. EVENT DETECTION METHODS BY TRACKING THE CHANGE IN THE COMMUNICATION TRENDS*

In this method, firstly, the initial network is constructed. Then, the community detection algorithms, either dSLM or InfoMap, is applied on this initial network. Rather than applying community detection at each time interval, communities are determined only once, and then in each of the subsequent time intervals, the change in the communication trends is computed based on this initial structure.

The change is quantified using two different metrics: inter-intra links ratio (IILR) and modularity (MOD). Two different comparison strategies are used for change detection. In the first one, the standard deviation (SD) value obtained in a time interval is compared against a given threshold. Note that this value can be computed either with IILR (CT-IILR-SD) or

**Algorithm 3** Event Detection via Change on CT (St. Dev.)

**Require:** IN, setOfGraphs, equation, bucket, threshold
**Ensure:** events
```
for i ← 1 to timeWindowCount do
    cur ← CT(IN, setOfGraphs(t_i), equation, bucket)
    if cur ≥ threshold then
        add i to events
    end if
end for
return events
```

**Algorithm 4** Event Detection via Change on CT (Cur-Prev)

**Require:** IN, setOfGraphs, equation, bucket, threshold
**Ensure:** events
```
prev ← CT(IN, setOfGraphs(t_1), equation, bucket)
for i=2 to timeWindowCount do
    cur ← CT(IN, setOfGraphs(t_i), equation, bucket)
    change ← abs(cur − prev)/prev
    if change ≥ threshold then
        add i to events
    end if
    prev ← cur
end for
return events
```

modularity (CT-MOD-SD). In the second one, rather than the standard deviation value obtained for the network of the time interval, the amount of change with respect to the previous time interval is compared against a given threshold. For this strategy, the value of the network can be determined by IILR (CT-IILR) or modularity (CT-MOD). A similar approach is used by Moriano et al. in [20] such that the authors use InfoMap as the community detection algorithm, IILR as the value to track and standard deviation as the comparison method. This study extends the method in [20] by including dSLM as the community detection algorithm, modularity as the value to track and the change from the previous time interval as the comparison strategy.

The algorithm using the standard deviation as comparison method is given in Algorithm 3, whereas the other one using the change from the previous time interval is given in Algorithm 4. In both algorithms, the tracked value of graph corresponding to the given time interval can be computed by either IILR or modularity (denoted with *eqnarray* parameter in the algorithms).

## IV. EXPERIMENTS ON EVENT DETECTION PERFORMANCE

The effectiveness of the proposed methods is analyzed in terms of a variety of aspects including community size, group of communities, time resolution and scalability, through a set of experiments. In this section, the data sets are presented followed by experiment settings. Then the results of the experiments analyzing different aspects are given in separate subsections.

### A. DATA SETS & GROUND TRUTHS

In this section, the data sets used in the experiments and the ground truth events within the data sets are presented.

#### 1) ENRON

This is an email communication data set containing more than 125,000 emails sent by 184 employees of Enron company between 1999-01-01 and 2002-04-30. Enron is a U.S company that has filed for bankruptcy just before 2000 [12]. There are seven significant events marked in this time interval [11], [20]. The following dates include ground truth events: 2001-05-17, 2001-07-12, 2001-08-03, 2001-10-16, 2001-12-02, 2002-02-14, 2002-04-09. The details of the events are given in Table 1 which is taken from Moriano et. al. [20].

#### 2) BOSTON BOMBING TWEETS

The data set contains tweet communication networks extracted from more than 456 million English tweets posted in April 2013 [20]. The mention networks contain around 7 million nodes and 10 million edges for each day. The retweet networks contain around 3.5 million nodes and 4.2 million edges for each day. There are two major events in this time interval. These ground truth events are as follows:

- 2013-04-15 Bombing
- 2013-04-19 Manhunt

In the results, mention network of this data set is abbreviated as BBM. Similarly, retweet network is shown as BBR.

#### 3) REALITY MINING

This data set is a CDR collection containing 99,633 call record instances. It includes the communication data of 97 faculty, student, and staff at MIT, recorded by the software on their mobile devices over 50 weeks from August 2004 to July 2005 [14]. There are 50 weeks in this data set's time span and the weeks having id 6, 12, 13, 15, 16, 17, 19, 20, 21, 22, 23, 27, 31, 32, 34, 35 are marked as ground truth event weeks. These weeks involve semester breaks, exam and sponsor weeks, and holidays [27]. The name of this data set is abbreviated as RM in the result tables. The sub communication network obtained by SMS interactions is shown with RMS and the sub-network of voice calls is shown as RMV. Different graph structures based on edge directions (directed-D vs. undirected-U), and edge weight (weighted-W vs. unweighted-U) are constructed.

#### 4) TURKISH GSM OPERATOR

There are 297,009,183 call records in this CDR data set. In total, there are 12,521,352 individuals (nodes in the communication network), and 56,316,192 phone call relations (edges) recorded in September 2012.[1] When the days in the data set are labelled from 1 to 30, the days with id 2, 3, 4, 5, 6, 11, 16, 17, 18, 22, 24, 25, 29, 30 are marked as ground truth event days. These events are extracted from

[1] The 56,316,192 edges correspond to distinct interactions within 297,009,183 entries.

**TABLE 1.** Enron ground truth events.

| Event ID | Date | Description |
|---|---|---|
| 1 | 2001-05-17 | Schwarzenegger, Lay, Milken meeting. |
| 2 | 2001-07-12 | Quarterly conference call. |
| 3 | 2001-08-03 | Skilling makes a bullish speech on Enron Energy Services. That afternoon, he lays off 300 employees. |
| 4 | 2001-10-16 | Enron reports a 618 million third-quarter loss and declares a 1.01 billion non-recurring charge against its balance sheet, partly related to "structured finance" operations run by chief financial officer Andrew Fastow. In the analyst conference call that day, Lay also announces a 1.2 billion cut in shareholder equity. |
| 5 | 2001-12-02 | Enron, at the time the largest bankruptcy in U.S. history, files for Chapter 11 bankruptcy protection. |
| 6 | 2002-02-14 | Sherron Watkins, the Enron whistleblower, testifies before a Congressional panel against Skilling and Lay. |
| 7 | 2002-04-09 | David Duncan, Arthur Andersen's former top auditor, pleads guilty to obstruction. |

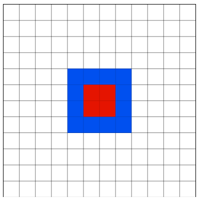

**FIGURE 2.** Illustration of big central square and central squares subsets of Turkish GSM operator data set.

public news resources in Turkey for September 2012. This full data set is denoted as TGSM in the result tables. This data set corresponds to calls covering a large area including a smaller and much more crowded city center. Some events in this data set are related with the whole country whereas some others are related with the city center. In order to evaluate the effect of considering only central area, two subsets of the data set are constructed. The whole area covered by the communication network is represented as a 12 by 12 grid. Then, the middle 4 by 4 grid region is marked as *Big Central Square (BCS)*, and 2 by 2 inmost grid is marked as *Central Square* as shown in Figure 2. The details of these subsets are as follows:

- **Big Central Square (BCS).** It contains the records collected from 11545 base stations in the central region out of total 13281 base stations. In total there are 2,853,473 nodes and 11,376,594 edges in this communication network.
- **Central Square (CS).** It contains the records collected from the central 85 base stations out of total 13281 base stations. In total there are 1,335,945 nodes and 3,568,967 edges in this subset.

### B. EXPERIMENT SETTINGS

In the experiments, the results are analyzed under precision, recall, f-measure, false positive rate, and true positive rate metrics. However, for the overall comparison the area under the Precision-Recall curve is used. The curve is plotted under 100 threshold values and the area under this curve is denoted as Average Precision [27].

The experiments are conducted on a computer with 32 GB RAM and Intel(R) Core(TM) i7-10750H CPU@2.60GHz processor.

A set of experiments is devised in order to answer the following research questions:

- RQ1. How effective are the proposed change tracking methods for event detection?
- RQ2. Is the size and amount of the communities to consider an effect for event detection? How does it affect the detection performance?
- RQ3. How does the time resolution affect the community detection?
- RQ4. Does community change comparison method effect the community detection performance?
- RQ5. Does community detection based event detection approach provide a feasible solution in terms of running time performance?

In the following subsections, the experiments conducted to answer the above research questions are presented. Additionally, the event detection performance of these experiments is compared with the previous studies.

### C. EVENT DETECTION PERFORMANCE OF THE METHODS UNDER THE BASIC SETTINGS

In this experiment, the performance of the proposed approaches are analyzed under the basic settings (i.e., full data set, all communities (Bucket A), finest time resolution (Resolution 1)). In Table 2, the list of methods and their abbreviations are presented.

**TABLE 2.** Abbreviations for the basic methods.

| Abbreviation | Method Description |
|---|---|
| CN-NEW | Track the number of new central nodes |
| CN-SIZE | Track the number of central nodes |
| CN-NOT | Track the number of not anymore central nodes |
| CN-NOT-NEW | Track the number of not anymore + new central nodes |
| NOC | Track the number of communities |
| CT-IILR | Track the IILR by comparing with prev time |
| CT-MOD | Track the Modularity by comparing with prev time |
| CT-IILR-SD | Track the IILR with respect to standard deviation |
| CT-MOD-SD | Track the Modularity with respect to standard deviation |

In Table 3, the results are given for Boston Bombing (BBM and BBR), Enron (ENR), Reality Mining (RMS and RMV) and Turkish GSM (TGSM) data sets. The best performing graph types are selected for each data set to present in Table 3. The detailed results for RM and TGSM data sets are presented in Table 4 and Table 5, respectively. In Table 4, data set names reflect the graph structure used for modeling the network. For example, S-DU denotes SMS communication network with Directed Unweighted graph.

**TABLE 3. Experiment results of basic methods (In Average Precision).**

| Method/Dataset | BBM | BBR | ENR | RMS | RMV | TGSM |
|---|---|---|---|---|---|---|
| CN-NEW | **0.58** | 0.13 | 0.07 | **0.26** | **0.35** | 0.6 |
| CN-NOT | 0.11 | 0.05 | 0.07 | 0.19 | 0.32 | **0.62** |
| CN-NOT-NEW | 0.03 | 0.03 | 0.13 | 0.18 | 0.3 | 0.24 |
| CN-SIZE | 0.12 | 0.2 | 0.13 | 0.25 | **0.35** | 0.57 |
| CT-IILR | 0.17 | 0.32 | 0.07 | 0.25 | 0.23 | 0.3 |
| CT-MOD | 0.35 | 0.13 | **0.19** | 0.25 | 0.25 | 0.6 |
| NOC | 0.18 | **0.6** | 0.08 | 0.25 | 0.33 | 0.41 |

**TABLE 4. Experiment results of basic methods: Detailed results on RM data set (In Average Precision).**

| Method/Dataset | S-DU | S-DW | S-UW | V-DU | V-DW | V-UW |
|---|---|---|---|---|---|---|
| CN-NEW | **0.26** | **0.34** | **0.4** | **0.35** | 0.26 | 0.3 |
| CN-NOT | 0.19 | 0.26 | 0.33 | 0.32 | 0.23 | 0.27 |
| CN-NOT-NEW | 0.18 | 0.24 | 0.23 | 0.3 | 0.23 | 0.17 |
| CN-SIZE | 0.25 | 0.27 | 0.36 | **0.35** | 0.23 | **0.33** |
| CT-IILR | 0.25 | 0.26 | 0.24 | 0.23 | 0.23 | 0.22 |
| CT-MOD | 0.25 | 0.22 | 0.24 | 0.25 | 0.23 | 0.25 |
| NOC | 0.25 | 0.26 | 0.21 | 0.33 | **0.33** | 0.24 |

**TABLE 5. Experiment results of basic methods: Detailed results on TGSM data set (In Average Precision).**

| Method/Dataset | TGSM | BCS | CS |
|---|---|---|---|
| CN-NEW | 0.6 | 0.62 | 0.5 |
| CN-NOT | **0.62** | 0.54 | 0.42 |
| CN-NOT-NEW | 0.24 | 0.24 | 0.22 |
| CN-SIZE | 0.57 | 0.56 | 0.53 |
| CT-IILR | 0.3 | 0.43 | 0.52 |
| CT-MOD | 0.6 | **0.75** | **0.62** |
| NOC | 0.41 | 0.55 | 0.58 |

According to the results, there is no single method that consistently outperforms the others on all data sets. CN-NEW gives the highest average precision on three data sets (BBM, RMS, RMV), and CN-NOT, CT-IILR, CT-MOD provide the best results for the other three data sets (TGSM, BBR, ENR), respectively.

At this point, it is worth examining the results given in Table 5. In these results, it is observed that using the central part of the geographical area increases the event detection performance up until 0.75 with CT-MOD on BCS. Narrowing down the area decreases the data set size as well. The effect of this change on running time is elaborated in Section IV-H.

### D. EVENT DETECTION PERFORMANCE OF THE ENSEMBLE METHODS UNDER THE BASIC SETTINGS

Detection performance of several ensemble event detectors are also analyzed under the basic setting. The list of methods with their abbreviations are presented in Table 6.

The results are presented in Table 7. In this table, the best performance obtained for data set modeling variations of the BBM, BBR, ENR, RMS, RMV, and TGSM data sets is given. The detailed results for RM and TGSM data sets are presented in Table 8 and Table 9, respectively.

In the results, it is seen that there is performance improvement only for some of the data sets. The ensemble method bringing the improvement varies for each of these cases.

**TABLE 6. Abbreviations for the ensemble methods.**

| Abbreviation | Method Description |
|---|---|
| NOC-qx..qy-OR | OR ensemble of buckets x..y in NOC |
| NOC-qx..qy-AND | AND ensemble of buckets x..y in NOC |
| NOC-NEW-OR | OR ensemble of NOC and CN-NEW |
| NOC-NEW-AND | AND ensemble of NOC and CN-NEW |
| NOC-NOT-OR | OR ensemble of NOC and CN-NOT |
| NOC-NOT-AND | AND ensemble of NOC and CN-NOT |
| NOC-NOT-NEW-OR | OR ensemble of NOC and CN-NOT-NEW |
| NOC-NOT-NEW-AND | AND ensemble of NOC and CN-NOT-NEW |
| NOC-SIZE-OR | OR ensemble of NOC and CN-SIZE |
| NOC-SIZE-AND | AND ensemble of NOC and CN-SIZE |

**TABLE 7. Evaluation on ensemble methods (In Average Precision).**

| Method/Dataset | BBM | BBR | ENR | RMS | RMV | TGSM |
|---|---|---|---|---|---|---|
| NOC-q1q2-OR | 0.13 | 0.55 | 0.1 | 0.27 | 0.3 | 0.54 |
| NOC-q1q3-OR | 0.13 | 0.55 | 0.1 | 0.27 | 0.3 | 0.54 |
| NOC-q1q4-OR | 0.13 | NA | 0.1 | 0.27 | 0.3 | 0.54 |
| NOC-q2q3-OR | 0.13 | 0.55 | 0.1 | 0.27 | 0.3 | 0.54 |
| NOC-q2q4-OR | 0.13 | NA | 0.1 | 0.27 | 0.3 | 0.54 |
| NOC-q3q4-OR | 0.13 | NA | 0.1 | 0.27 | 0.3 | 0.54 |
| NOC-q1q2-AND | 0.19 | **0.6** | 0.08 | 0.2 | 0.29 | 0.4 |
| NOC-q1q3-AND | 0.19 | **0.6** | 0.08 | 0.2 | 0.29 | 0.4 |
| NOC-q1q4-AND | 0.19 | NA | 0.08 | 0.2 | 0.29 | 0.4 |
| NOC-q2q3-AND | 0.19 | **0.6** | 0.08 | 0.2 | 0.29 | 0.4 |
| NOC-q2q4-AND | 0.19 | NA | 0.08 | 0.2 | 0.29 | 0.4 |
| NOC-q3q4-AND | 0.19 | NA | 0.08 | 0.2 | 0.29 | 0.4 |
| NOC-q1q2q3-OR | 0.13 | 0.55 | **0.18** | 0.28 | 0.27 | 0.55 |
| NOC-q1q2q4-OR | 0.13 | NA | **0.18** | 0.28 | 0.27 | 0.55 |
| NOC-q2q3q4-OR | 0.13 | NA | **0.18** | 0.28 | 0.27 | 0.55 |
| NOC-q1q3q4-OR | 0.13 | NA | **0.18** | 0.28 | 0.27 | 0.55 |
| NOC-q1q2q3-AND | 0.16 | **0.6** | 0.08 | 0.08 | 0.25 | 0.43 |
| NOC-q1q2q4-AND | 0.16 | NA | 0.08 | 0.08 | 0.25 | 0.43 |
| NOC-q2q3q4-AND | 0.16 | NA | 0.08 | 0.08 | 0.25 | 0.43 |
| NOC-q1q3q4-AND | 0.16 | NA | 0.08 | 0.08 | 0.25 | 0.43 |
| NOC-q1q2q3q4-OR | 0.1 | NA | 0.1 | **0.33** | 0.27 | 0.57 |
| NOC-q1q2q3q4-AND | 0.17 | NA | 0.05 | 0.02 | 0.18 | 0.38 |
| NOC-NEW-OR | **0.58** | 0.13 | 0.08 | 0.28 | 0.32 | 0.6 |
| NOC-NEW-AND | 0.18 | **0.6** | 0.08 | 0.23 | 0.43 | 0.41 |
| NOC-NOT-OR | 0.11 | 0.05 | 0.08 | 0.26 | 0.32 | **0.62** |
| NOC-NOT-AND | 0.18 | **0.6** | 0.08 | 0.2 | 0.39 | 0.41 |
| NOC-NOT-NEW-OR | 0.03 | 0.03 | 0.12 | 0.27 | 0.32 | 0.24 |
| NOC-NOT-NEW-AND | 0.18 | **0.6** | 0.07 | 0.21 | 0.41 | 0.41 |
| NOC-SIZE-OR | 0.27 | 0.2 | 0.14 | 0.28 | 0.29 | 0.56 |
| NOC-SIZE-AND | 0.06 | **0.6** | 0.07 | 0.21 | **0.46** | 0.43 |

For RMS data set, average precision increases from 0.26 to 0.33 by NOC-q1q2q3q4-OR. For RMV, there is increase from 0.35 to 0.46 average precision by NOC-SIZE-AND.

The same situation is also observed for the variations of TGSM data set, which are BCS and CS. Some of the ensemble methods provide improvement over basic methods. For BCS data set, the highest performance obtained by CT-MOD is not exceeded by any ensemble method. However, for CS data set, average precision increases from 0.62 to 0.66 by NOC-q1q2q3-OR.

### E. ANALYSIS ON THE EFFECT OF BUCKET SIZES

As described in Definition 8, the detected communities are sorted in ascending order of community size and grouped under buckets. In this grouping, for instance, A refers to all communities, whereas Q3 refers to the set of communities in the third quartile. The results of the experiments under these buckets are presented in Table 10 for CDR data sets (RMS, RMV and TGSM) and in Table 11 for Social media data sets (BBM, BBR and ENR).

**TABLE 8.** Evaluation on ensemble methods: RM (In Average Precision).

| Method/Dataset | SDU | SDW | SUW | VDU | VDW | VUW |
|---|---|---|---|---|---|---|
| NOC-q1q2-OR | 0.27 | 0.26 | 0.31 | 0.3 | 0.28 | 0.29 |
| NOC-q1q3-OR | 0.27 | 0.26 | 0.31 | 0.3 | 0.28 | 0.29 |
| NOC-q1q4-OR | 0.27 | 0.26 | 0.31 | 0.3 | 0.28 | 0.29 |
| NOC-q2q3-OR | 0.27 | 0.26 | 0.31 | 0.3 | 0.28 | 0.29 |
| NOC-q2q4-OR | 0.27 | 0.26 | 0.31 | 0.3 | 0.28 | 0.29 |
| NOC-q3q4-OR | 0.27 | 0.26 | 0.31 | 0.3 | 0.28 | 0.29 |
| NOC-q1q2-AND | 0.2 | 0.1 | 0.15 | 0.29 | 0.22 | 0.18 |
| NOC-q1q3-AND | 0.2 | 0.1 | 0.15 | 0.29 | 0.22 | 0.18 |
| NOC-q1q4-AND | 0.2 | 0.1 | 0.15 | 0.29 | 0.22 | 0.18 |
| NOC-q2q3-AND | 0.2 | 0.1 | 0.15 | 0.29 | 0.22 | 0.18 |
| NOC-q2q4-AND | 0.2 | 0.1 | 0.15 | 0.29 | 0.22 | 0.18 |
| NOC-q3q4-AND | 0.2 | 0.1 | 0.15 | 0.29 | 0.22 | 0.18 |
| NOC-q1q2q3-OR | 0.28 | 0.27 | 0.31 | 0.27 | 0.27 | 0.3 |
| NOC-q1q2q4-OR | 0.28 | 0.27 | 0.31 | 0.27 | 0.27 | 0.3 |
| NOC-q2q3q4-OR | 0.28 | 0.27 | 0.31 | 0.27 | 0.27 | 0.3 |
| NOC-q1q3q4-OR | 0.28 | 0.27 | 0.31 | 0.27 | 0.27 | 0.3 |
| NOC-q1q2q3-AND | 0.08 | 0.07 | 0.13 | 0.25 | 0.15 | 0.11 |
| NOC-q1q2q4-AND | 0.08 | 0.07 | 0.13 | 0.25 | 0.15 | 0.11 |
| NOC-q2q3q4-AND | 0.08 | 0.07 | 0.13 | 0.25 | 0.15 | 0.11 |
| NOC-q1q3q4-AND | 0.08 | 0.07 | 0.13 | 0.25 | 0.15 | 0.11 |
| NOC-q1q2q3q4-OR | **0.33** | **0.3** | 0.28 | 0.27 | 0.25 | 0.3 |
| NOC-q1q2q3q4-AND | 0.02 | 0.02 | 0.01 | 0.18 | 0.11 | 0.02 |
| NOC-NEW-OR | 0.28 | 0.29 | **0.36** | 0.32 | 0.3 | **0.32** |
| NOC-NEW-AND | 0.23 | 0.28 | 0.23 | 0.43 | 0.28 | 0.22 |
| NOC-NOT-OR | 0.26 | 0.27 | 0.32 | 0.32 | **0.32** | 0.31 |
| NOC-NOT-AND | 0.2 | 0.25 | 0.24 | 0.39 | 0.23 | 0.22 |
| NOC-NOT-NEW-OR | 0.27 | 0.28 | 0.22 | 0.32 | 0.29 | 0.2 |
| NOC-NOT-NEW-AND | 0.21 | 0.26 | 0.23 | 0.41 | 0.25 | 0.22 |
| NOC-SIZE-OR | 0.28 | 0.27 | 0.34 | 0.29 | 0.28 | 0.31 |
| NOC-SIZE-AND | 0.21 | 0.24 | 0.24 | **0.46** | 0.26 | 0.27 |

**TABLE 9.** Evaluation on ensemble methods: TGSM (In Average Precision).

| Method/Dataset | TGSM | BCS | CS |
|---|---|---|---|
| NOC-q1q2-OR | 0.54 | 0.57 | 0.65 |
| NOC-q1q3-OR | 0.54 | 0.57 | 0.65 |
| NOC-q1q4-OR | 0.54 | 0.57 | 0.65 |
| NOC-q2q3-OR | 0.54 | 0.57 | 0.65 |
| NOC-q2q4-OR | 0.54 | 0.57 | 0.65 |
| NOC-q3q4-OR | 0.54 | 0.57 | 0.65 |
| NOC-q1q2-AND | 0.4 | 0.49 | 0.59 |
| NOC-q1q3-AND | 0.4 | 0.49 | 0.59 |
| NOC-q1q4-AND | 0.4 | 0.49 | 0.59 |
| NOC-q2q3-AND | 0.4 | 0.49 | 0.59 |
| NOC-q2q4-AND | 0.4 | 0.49 | 0.59 |
| NOC-q3q4-AND | 0.4 | 0.49 | 0.59 |
| NOC-q1q2q3-OR | 0.55 | 0.61 | **0.66** |
| NOC-q1q2q4-OR | 0.55 | 0.61 | **0.66** |
| NOC-q2q3q4-OR | 0.55 | 0.61 | **0.66** |
| NOC-q1q3q4-OR | 0.55 | 0.61 | **0.66** |
| NOC-q1q2q3-AND | 0.43 | 0.45 | 0.51 |
| NOC-q1q2q4-AND | 0.43 | 0.45 | 0.51 |
| NOC-q2q3q4-AND | 0.43 | 0.45 | 0.51 |
| NOC-q1q3q4-AND | 0.43 | 0.45 | 0.51 |
| NOC-q1q2q3q4-OR | 0.57 | **0.62** | 0.65 |
| NOC-q1q2q3q4-AND | 0.38 | 0.45 | 0.51 |
| NOC-NEW-OR | 0.6 | **0.62** | 0.54 |
| NOC-NEW-AND | 0.41 | 0.55 | 0.53 |
| NOC-NOT-OR | **0.62** | 0.54 | 0.45 |
| NOC-NOT-AND | 0.41 | 0.55 | 0.53 |
| NOC-NOT-NEW-OR | 0.24 | 0.24 | 0.25 |
| NOC-NOT-NEW-AND | 0.41 | 0.55 | 0.53 |
| NOC-SIZE-OR | 0.56 | 0.6 | 0.61 |
| NOC-SIZE-AND | 0.43 | 0.55 | 0.49 |

The results clearly show that, for CDR data sets, certain buckets of communities affect the event detection performance. In CDR data sets, for RMS data set, performance of CN-NEW method increases from 0.26 to 0.38 when H1 is used. Similarly, for RMV, the performance of CN-NEW rises from 0.35 to 0.48. However, the bucket to prefer is data dependent. On the other hand, for social media data sets, using buckets generally degrades the performance, the highest average precision results are obtained with all communities (bucket A).

### F. ANALYSIS ON THE EFFECT OF TIME RESOLUTION

In our data sets, the events are marked per day or per week. Therefore, the changes in the communities are tracked on the time resolution given in the data set. However, this resolution can be too strict when the effect of the event on the communication network is propagated with a delay. In order to analyze the effect of time resolution for change detection, the setting for ENR and TGSM data sets is modified, such that, time window is set as 2 days and 3 days, in addition to daily time window (window size 1). The results of the analysis are given in Table 12. The analysis shows a considerable increase in the performance of CT-MOD, which provided the highest accuracy performance with the finest grained time resolution for ENR data set, specifically for window size 2. The trend is similar also for the other methods.

### G. ANALYSIS ON THE EFFECT OF COMMUNITY CHANGE COMPARISON METHOD

In [20], Moriano et al. propose event detection through tracking communication trend changes in communities using InfoMap as the community detection algorithm, IILR as the value to track and standard deviation as the comparison method. In our proposed solution, solution of [20] is modified such that, dSLM is used as the community detection algorithm, modularity is used as the value to track and change from the previous time interval is used as change comparison method. In this analysis, the change comparison method in our setting is replaced with change from the previous time interval tracking in order to evaluate its effect on event detection performance. The results are presented in Table 13. The results reveal that the way change is computed is effective on the result. The proposed change tracking method has clear advantage for the CT-MOD method as it produces higher accuracy values for 6 out of 8 data sets, whereas it performs similar with the standard deviation based change tracking method for the CT-IILR method.

### H. ANALYSIS ON THE RUNNING TIME PERFORMANCE

Although all investigated methods include community detection as a core task, the change detection approaches and techniques differ considerably. This difference leads to variations in time costs for the methods. In Table 14, the running time duration of the basic methods are given in seconds. All the CN methods are computed in one run to increase performance. In the table, the average execution time for 4 CN based methods are reported. As seen in the results, the methods based on change tracking on the community structure (CN and NOC) are computationally lightweight compared to communication trend based methods (CT-IILR, CT-MOD).

The running time duration for variations of TGSM data set are given in Table 15. As expected, execution time decreases for BCS and CS, as the size of the processed data gets smaller compared to TGSM. As an interesting observation, focusing on central regions affects also event detection performance positively.

**TABLE 10. Evaluation on basic methods: Bucket size effect (CDR data sets) (In Average Precision).**

| Method/Dataset | RMS | | | | | | | | RMV | | | | | | | | TGSM | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bucket Size | A | F | H1 | H2 | Q1 | Q2 | Q3 | Q4 | A | F | H1 | H2 | Q1 | Q2 | Q3 | Q4 | A | F | H1 | H2 | Q1 | Q2 | Q3 | Q4 |
| CN-NEW | **0.26** | **0.35** | **0.38** | **0.33** | **0.28** | 0.2 | **0.33** | 0.2 | **0.35** | **0.4** | **0.48** | **0.41** | **0.4** | **0.49** | **0.32** | 0.42 | 0.6 | 0.61 | 0.61 | 0.46 | **0.78** | **0.62** | 0.5 | 0.37 |
| CN-NOT | 0.19 | 0.24 | 0.3 | 0.25 | 0.24 | 0.22 | 0.28 | **0.27** | 0.32 | 0.38 | 0.37 | 0.33 | 0.29 | 0.46 | 0.22 | 0.43 | **0.62** | 0.61 | 0.63 | 0.49 | 0.66 | 0.52 | 0.52 | 0.38 |
| CN-NOT-NEW | 0.18 | 0.23 | 0.21 | 0.17 | 0.16 | 0.16 | 0.22 | 0.15 | 0.3 | 0.36 | 0.26 | 0.29 | 0.21 | 0.23 | 0.2 | 0.29 | 0.24 | 0.24 | 0.24 | 0.22 | 0.25 | 0.24 | 0.24 | 0.25 |
| CN-SIZE | 0.25 | 0.24 | 0.3 | 0.27 | 0.26 | 0.24 | **0.33** | 0.25 | **0.35** | 0.37 | 0.36 | 0.33 | 0.3 | 0.41 | 0.3 | **0.46** | 0.57 | **0.71** | **0.67** | **0.67** | 0.64 | 0.56 | **0.71** | **0.63** |
| NOC | 0.25 | 0.21 | 0.18 | 0.25 | **0.28** | **0.28** | 0.32 | 0.12 | 0.33 | 0.25 | 0.26 | 0.18 | 0.1 | 0.37 | 0.26 | 0.4 | 0.41 | 0.51 | 0.5 | 0.51 | 0.57 | 0.52 | 0.49 | 0.41 |

**TABLE 11. Evaluation on basic methods: Bucket Size Effect (Social media data sets) (In Average Precision).**

| Method/Dataset | BBM | | | | | | | | BBR | | | | | | | | ENR | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Bucket Size | A | F | H1 | H2 | Q1 | Q2 | Q3 | Q4 | A | F | H1 | H2 | Q1 | Q2 | Q3 | Q4 | A | F | H1 | H2 | Q1 | Q2 | Q3 | Q4 |
| CN-NEW | **0.58** | **0.52** | 0.04 | 0.09 | **0.04** | 0.04 | 0.08 | 0.04 | 0.13 | 0.06 | 0.05 | 0.05 | 0.05 | 0.05 | 0.01 | 0.01 | 0.07 | 0.06 | 0.14 | 0.07 | 0.06 | 0.04 | **0.09** | 0.01 |
| CN-NOT | 0.11 | 0.27 | 0.04 | 0.09 | **0.04** | 0.04 | 0.07 | 0.04 | 0.05 | 0.06 | 0.06 | 0.05 | 0.06 | 0.05 | 0.01 | 0.01 | 0.07 | 0.06 | 0.12 | 0.08 | 0.03 | 0.03 | 0.06 | 0.05 |
| CN-NOT-NEW | 0.03 | 0.03 | 0.03 | 0.04 | 0.03 | 0.03 | 0.04 | 0.04 | 0.03 | 0.03 | 0.03 | 0.03 | 0.03 | 0.04 | 0.01 | 0.01 | **0.13** | 0.07 | 0.05 | **0.11** | 0.04 | 0.04 | 0.08 | **0.07** |
| CN-SIZE | 0.12 | 0.1 | 0.07 | 0.11 | 0.01 | 0.1 | **0.14** | 0.05 | 0.2 | 0.28 | 0.27 | 0.3 | 0.27 | 0.35 | 0.06 | 0.06 | **0.13** | **0.11** | **0.18** | 0.06 | 0.11 | **0.06** | 0.08 | 0.03 |
| NOC | 0.18 | 0.09 | **0.08** | **0.12** | 0.03 | **0.13** | 0.11 | **0.06** | **0.6** | **0.55** | **0.55** | **0.55** | **0.55** | **0.56** | **0.54** | **0.54** | 0.08 | 0.08 | 0.17 | 0.02 | **0.13** | 0.02 | 0.05 | 0.01 |

**TABLE 12. Evaluation on basic methods: Resolution effect (In Average Precision).**

| Method/Dataset | ENRON | | | TGSM | | | BCS | | | CS | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Resolution | 1 | 2 | 3 | 1 | 2 | 3 | 1 | 2 | 3 | 1 | 2 | 3 |
| CN-NEW | 0.07 | 0.07 | 0.2 | 0.6 | 0.85 | 0.83 | 0.62 | 0.68 | 0.86 | 0.5 | 0.81 | 0.71 |
| CN-NOT | 0.07 | 0.18 | 0.17 | **0.62** | 0.76 | 0.73 | 0.54 | 0.84 | 0.73 | 0.42 | 0.8 | 0.67 |
| CN-NOT-NEW | 0.13 | **0.26** | 0.21 | 0.24 | 0.32 | 0.33 | 0.24 | 0.32 | 0.33 | 0.22 | 0.32 | 0.33 |
| CN-SIZE | 0.13 | 0.17 | 0.09 | 0.57 | 0.64 | 0.67 | 0.56 | 0.77 | 0.58 | 0.53 | 0.79 | 0.75 |
| CT-IILR | 0.07 | 0.18 | **0.29** | 0.3 | 0.06 | 0.08 | 0.43 | 0.5 | 0.08 | 0.52 | 0.72 | 0.58 |
| CT-MOD | **0.19** | 0.22 | 0.18 | 0.6 | **0.94** | **0.9** | **0.75** | **0.94** | **0.89** | **0.62** | **0.92** | **0.9** |
| NOC | 0.08 | 0.13 | 0.24 | 0.41 | 0.79 | 0.46 | 0.55 | 0.82 | 0.65 | 0.58 | 0.81 | 0.78 |

**TABLE 13. Evaluation on basic methods: Standard Deviation Effect (In Average Precision).**

| Method/Dataset | BBM | BBR | ENRON | RMS | RMV | TGSM | BCS | CS |
|---|---|---|---|---|---|---|---|---|
| CT-IILR | 0.17 | **0.32** | 0.07 | 0.25 | 0.23 | 0.3 | 0.43 | 0.52 |
| CT-IILR-SD | 0.15 | 0.28 | 0.16 | **0.39** | 0.27 | 0.38 | 0.39 | 0.37 |
| CT-MOD | **0.35** | 0.13 | **0.19** | 0.25 | 0.25 | **0.6** | **0.75** | **0.62** |
| CT-MOD-SD | 0.19 | 0.09 | 0.05 | 0.34 | **0.5** | 0.36 | 0.44 | 0.51 |

**TABLE 14. Evaluation on basic methods: Run Times (In Seconds).**

| Method/Dataset | BBM | BBR | ENR | RMS | RMV | TGSM |
|---|---|---|---|---|---|---|
| CN | 158.22 | 58.49 | **0.08** | 0.06 | 0.11 | 70.93 |
| CT-IILR | 470.9 | 181.41 | 0.24 | 0.09 | 0.14 | 154.47 |
| CT-MOD | 93.62 | 45.6 | 0.15 | 0.12 | 0.14 | 51.95 |
| NOC | **13.59** | **4.89** | 0.41 | **0.03** | **0.03** | **4.12** |

**TABLE 15. Evaluation on basic methods run times: Variation of TGSM data set (In Seconds).**

| Method/Dataset | TGSM | BCS | CS |
|---|---|---|---|
| CN | 70.93 | 5.43 | 1.26 |
| CT-IILR | 154.47 | 13.29 | 5.11 |
| CT-MOD | 51.95 | 3.33 | 0.86 |
| NOC | **4.12** | **0.65** | **0.28** |

### *I. COMPARISON WITH BASELINE STUDIES*

As the baseline methods, the methods by Moriano et al. in [20] and Rayana et al. in [27] are used.

#### 1) COMPARISON WITH [20]

As the basic differences from the communication trend based solution of our work, Moriano et al. present experiments by using the InfoMap community detection algorithm and standard deviation for change computing. In [20], the event detection performance on Enron data set is presented only in terms of precision-recall curve. Our results on the same data set with CT based methods using dSLM are given in Table 13. Additionally, the average precision with InfoMap community detection algorithm (using the original source codes provided by Moriano et al.) on Enron data set is 0.14.

Moriano et. al. also conducted experiments on Boston Bombing data set. They provide the results of tracking the inter-intra ratio visually [20], but precision-recall curve or the average precision result are not reported. In our analysis, average precision values obtained on this data set is relatively low. This is possibly due to noise in the data set such that there are fluctuations, and on day 8, the data includes another event not related with Boston bombing and hence not considered in the ground truth event set. Additionally, there is missing data for day 18. As for execution time performance, for this data set, our community detection step takes 356 minutes in total for both mention and retweet networks. The event detection step takes 103 minutes in total. Thus, experiments including both networks are completed in 459 minutes (7.6 hours).

#### 2) COMPARISON WITH [27]

When the original source code of the methods in [27] is executed on Boston Bombing data set, the mention network experiment results in out of memory error under 32 GB RAM. The retweet network data is read in about 1 hour, but the codes cannot process it even after 7 hours. Therefore, the experiment is canceled after 8 hours without obtaining any result.

On reality mining data set, 5 basic and 5 ensemble methods of [27] are executed on both directed weighted voice call and SMS networks. The analysis on the voice call network takes 258 seconds, whereas it takes 36 seconds on the SMS network experiment. Thus, in total, it takes 294 seconds for all of the analysis on this data set. When all the proposed methods are performed (7 basic and 2 ensemble methods, with all bucket size variations) on all Reality mining networks, the community detection step takes 4 seconds, and the event detection step takes 13 seconds. Hence, on the total, it takes 17 seconds for our methods to conduct the whole analysis for this data set.

When the source code of [27] is executed on Turkish GSM Operator data set, it results with out of memory error, as well, under 32 GB RAM. For BCS and CS subsets, the code execution was stuck at PTSAD method of [27]. Therefore, the part of the source code related with this method is commented out and other base methods are used. The analysis on BCS data set takes 1272 seconds, and on CS data set, it takes 448 seconds. The execution times of all our proposed methods on TGSM, BCS and CS data sets are given in Table 16 for three different time resolutions. As seen in the table, for BCS it takes 701 seconds and for CS data set it takes 246 seconds (in resolution 1). Therefore, the execution time efficiency advantage of the proposed methods is clear.

**TABLE 16. Execution times of TGSM experiments (Total of 7 basic, 2 ensemble methods).**

| Dataset | Res. | Comm. Det. | Event Det. | Total Exec. |
|---|---|---|---|---|
| TGSM | 1 | 89 mins | 86 mins | 175 mins |
| TGSM | 2 | 69 mins | 63 mins | 132 mins |
| TGSM | 3 | 61 mins | 49 mins | 110 mins |
| BCS | 1 | 253 secs | 448 secs | 701 secs |
| BCS | 2 | 280 secs | 295 secs | 575 secs |
| BCS | 3 | 290 secs | 224 secs | 514 secs |
| CS | 1 | 60 secs | 186 secs | 246 secs |
| CS | 2 | 63 secs | 118 secs | 181 secs |
| CS | 3 | 66 secs | 93 secs | 159 secs |

Table 17 presents the event detection performance of the methods in [27] on Reality Mining, BCS and CS data sets, and the best results obtained by the proposed methods on the same data sets. For BCS data set, CT-MOD on bucket A (abbreviated as CT-MOD - A in the table) provides the highest average precision with a gap of 8% (compared to EBED outdegree). For the other data sets, although the methods in [27] provide higher event detection performance, the results of the proposed methods can be considered comparable also considering their time performance advantage, particularly for RMS-DW and CS data sets.

**TABLE 17. Evaluation of methods in [27] (In Average Precision).**

| Method/Dataset | RMS-DW | RMV-DW | BCS | CS |
|---|---|---|---|---|
| EBED indeg | 0.39 | 0.34 | 0.63 | **0.67** |
| PTSAD indeg | 0.51 | 0.48 | NA | NA |
| SpiritTest indeg | **0.68** | 0.56 | 0.48 | 0.49 |
| ASED indeg | 0.47 | 0.58 | 0.4 | 0.4 |
| MAED indeg | 0.53 | **0.66** | 0.48 | 0.47 |
| EBED outdeg | 0.37 | 0.47 | 0.67 | 0.65 |
| PTSAD outdeg | 0.39 | 0.53 | NA | NA |
| SpiritTest outdeg | 0.63 | 0.5 | 0.4 | 0.4 |
| ASED outdeg | 0.5 | **0.66** | 0.44 | 0.66 |
| MAED outdeg | 0.58 | 0.65 | 0.46 | 0.47 |
| Full Ensemble | 0.53 | 0.6 | 0.46 | 0.63 |
| SELECT-H Ensemble | 0.6 | 0.64 | 0.42 | 0.5 |
| SELECT-V Ensemble | 0.47 | 0.65 | 0.53 | 0.62 |
| DivE Ensemble | 0.49 | 0.5 | 0.63 | **0.67** |
| ULARA Ensemble | 0.57 | 0.61 | 0.52 | 0.62 |
| CT-IILR-SD - F | 0.6 | 0.25 | 0.39 | 0.35 |
| CT-MOD-SD - A | 0.23 | 0.52 | 0.44 | 0.51 |
| CT-MOD - A | 0.22 | 0.23 | **0.75** | 0.62 |

### *J. OVERVIEW & DISCUSSION*

The overview of the performance of all proposed methods on all data sets is presented in Table 18. Additionally, an overview of the performance for all buckets on all data sets is provided in Table 19. For each data set, the top result is colored with dark brown, the second top result is colored with orange, and the third top result is colored with light yellow. In Table 18, the methods are ordered based on the number of colored cells.

The results and contributions are analyzed under 4 parts. First, the overall winners of the experimented algorithms are presented. After that, the contributions of this study to community structure based, communication trend based, and general event detection areas are discussed. Additionally, a data set based analysis and discussion is presented.

#### 1) OVERALL WINNERS

Based on Table 18 which is sorted from best to worst method, the winner is the proposed CN-NEW method. The second one is an ensemble, NOC-NEW-OR. These two community structure based methods are followed by two communication trend based ones, namely CT-IILR-SD and CT-MOD-SD.

As seen in Table 18, communication trend based methods tend to perform better on CDR based data sets, whereas community structure based methods tend to perform better on social media-based data sets.

Community structure based methods have better execution time performance than the communication trend based approaches as seen in Table 14.

**TABLE 18. The overview of the best results: Method (In Average Precision).**

| Method/Dataset | BBM | BBR | ENR | RMS-DU | RMS-DW | RMS-UW | RMV-DU | RMV-DW | RMV-UW | TGSM | BCS | CS |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CN-NEW | 0.58 - A | | | 0.38 - H1 | 0.37 - H1 | 0.40 - A | 0.49 - Q2 | 0.45 - Q4 | 0.39 - H1 | 0.78 - Q1 | 0.67 - H1 | |
| NOC-NEW-OR | 0.58 - A | | 0.20 - H1 | 0.38 - H2 | | 0.40 - H1 | 0.49 - Q4 | | | 0.78 - Q1 | 0.67 - H1 | |
| CT-IILR-SD | | | | 0.39 - A | 0.60 - F | 0.55 - F | | | | | | |
| CT-MOD-SD | | | | | | | 0.50 - A | 0.52 - A | 0.46 - A | | | |
| CN-NOT | 0.27 - F | | | | | 0.39 - Q3 | | 0.46 - Q4 | 0.38 - Q3 | | 0.66 - Q1 | 0.69 - Q2 |
| NOC-NOT-AND | | 0.60 - A | | 0.37 - Q3 | | | 0.53 - Q4 | | | | | |
| CT-IILR | | | | | 0.43 - Q3 | | | | | 0.86 - Q2 | | 0.69 - H1 |
| CT-MOD | 0.35 - A | | 0.19 - A | | | | | | | | 0.75 - A | |
| NOC-SIZE-OR | 0.27 - F | | | 0.37 - Q3 | | | | | 0.38 - Q4 | 0.71 - F | 0.67 - Q2 | |
| NOC-SIZE-AND | | 0.60 - A | 0.21 - H1 | | | | | | | | | |
| NOC | | 0.60 - A | | | | | | | | | | 0.70 - Q4 |
| NOC-NEW-AND | | 0.60 - A | | | | | | 0.46 - Q3 | | | | |
| NOC-NOT-OR | 0.27 - F | | | | | | | | | | 0.66 - Q1 | 0.73 - Q2 |
| NOC-NOT-NEW-AND | | 0.60 - A | 0.20 - H1 | | | | | | | | | |
| CN-SIZE | | 0.35 - Q2 | | | | | | | 0.39 - F | 0.71 - F | | |
| NOC-q1q2-AND | | 0.60 - A | | | | | | | | | | |
| NOC-q1q3-AND | | 0.60 - A | | | | | | | | | | |
| NOC-q2q3-AND | | 0.60 - A | | | | | | | | | | |
| NOC-q1q2q3-AND | | 0.60 - A | | | | | | | | | | |
| NOC-q1q2-OR | | 0.55 - A | | | | | | | | | | |
| NOC-q1q3-OR | | 0.55 - A | | | | | | | | | | |
| NOC-q2q3-OR | | 0.55 - A | | | | | | | | | | |
| NOC-q1q2q3-OR | | 0.55 - A | | | | | | | | | | |

The points are given such that first results are 3, the second results are 2, and the third results are 1 point. Thus, each method has a total point. The methods are ordered descending by total points. Thus, the best method is on top.

**TABLE 19. The overview of the best results: Bucket (In Average Precision).**

| Bucket/Dataset | BBM | BBR | ENR | RMS-DU | RMS-DW | RMS-UW | RMV-DU | RMV-DW | RMV-UW | TGSM | BCS | CS |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A | 0.58 | 0.6 | 0.19 | 0.39 | | 0.4 | 0.5 | 0.52 | 0.46 | | 0.75 | |
| H1 | | 0.56 | 0.21 | 0.38 | | 0.4 | | 0.44 | 0.39 | | 0.67 | 0.69 |
| F | 0.52 | 0.56 | | | 0.6 | 0.55 | | | 0.39 | 0.71 | | |
| Q2 | | 0.57 | | | | 0.39 | 0.49 | | | 0.86 | 0.67 | 0.73 |
| Q3 | 0.25 | | | 0.37 | 0.56 | 0.39 | | 0.46 | 0.38 | 0.71 | | |
| Q4 | | | | | | | 0.53 | 0.46 | 0.39 | | | 0.7 |
| Q1 | | 0.56 | 0.14 | | | | | | | 0.78 | 0.66 | |
| H2 | | | | 0.38 | 0.53 | | | | | | | |

The points are given such that first results are 3, the second results are 2, and the third results are 1 point. Thus, each bucket has a total point. The buckets are ordered descending by total points. Thus, the best bucket is on top.

### 2) CONTRIBUTIONS TO COMMUNITY STRUCTURE BASED EVENT DETECTION

In this work, a new bucket concept is introduced. It is experimented on various scale of data sets. As seen in Table 10, using a bucket of the data set improves the accuracy of the community structure based event detection algorithms, particularly for the CDR based data sets. For instance, CN-NEW has 0.49 average precision value for the second quarter (Q2) of RMV dataset, whereas it has 0.35 average precision value for RMV data set. So, it can be stated that bucket concept increased average precision of CN-NEW method from 0.35 to 0.49 in RMV, from 0.62 to 0.78 in TGSM, from 0.26 to 0.38 in RMS data sets. As a summary, it can be stated that the bucket concept improves the accuracy of several methods for CDR based data sets. On the other hand, as it can be seen in Table 11, the similar improvement has not been observed for social media based data sets.

Table 19 also shows that using buckets H1, Q2, and Q4 produce best results for some of the cases. Regarding the execution time, using buckets considerably improves the execution time of the proposed methods since it decreases the amount of data to be processed.

The ensemble strategies devised in this work give clearer results than [3], since logical and/or combinations of the quarter buckets are used to form the ensembles. Furthermore, the proposed ensemble methods increase the best average precision values for some cases, such as, from 0.26 to 0.33 for RMS, from 0.35 to 0.46 for RMV, and from 0.62 to 0.66 for CS data sets, as it can be seen in Tables 3, 5, 7, 9.

The proposed resolution strategy provides the capability to detect events that come with a delayed communication representation. As seen in Table 12, this strategy improves the highest average precision values. For example, for different resolution values, average precision increases for ENR from 0.19 to 0.29, for TGSM from 0.62 to 0.94, for BCS from 0.75 to 0.89, and for CS from 0.62 to 0.92.

In this work, as a contribution to community structure based event detection, multiple centrality score calculation methods such as Betweenness, Katz, Harmonic, and PageRank centrality metrics are tested. As a result of these experiments, the highest average precision values for the CN methods have been obtained with the PageRank centrality metric. Thus, only the results with the PageRank centrality metric have been presented.

#### 3) CONTRIBUTIONS TO COMMUNICATION TREND BASED EVENT DETECTION

The study in [20] uses InfoMap as community detection method, whereas the proposed method, CT-IILR-SD, uses dSLM for community detection. For comparison, CT-IILR-SD with InfoMap is applied on ENR data set. CT-IILR-SD produces 0.16 average precision and outperforms CT-IILR-SD with InfoMap which produces 0.14 average precision on ENR data set. Furthermore, as seen in Table 13, when the tracked value and the tracking strategy are changed, it is observed that the event detection performance further improves over [20].

Among the communication trend based methods that are experimented, CT-MOD provides the best execution time performance. For instance, it is about 4 times faster than CT-IILR as seen in Tables 14, 15.

#### 4) COMPARISON WITH GRAPH FEATURE BASED EVENT DETECTION

The proposed community structure and communication trend based methods can be executed without any resource problems for big data sets (BBM, BBR, TGSM) whereas the graph feature based event detection methods [27] cannot reach the similar scalability level. For small data sets such as RM, community structure and communication trend based methods execute about 20 times faster than the graph feature based event detection methods. As seen in Table 17, the best performing proposed methods for these small data sets produce comparable event detection accuracy values with the graph feature based event detection methods in significantly less execution time.

#### 5) DATA SET BASED ANALYSIS AND DISCUSSION

The best resulting techniques with respect to data sets are summarized in Figure 3. In the figure, the data sets are grouped with respect to type (CDR vs. social media, e-mail) and size (small vs. large). For example, RM data set is small in size, and it is a CDR collection. For each data set, the technique giving the best result is also given together with the bucket size and the threshold used in the analysis

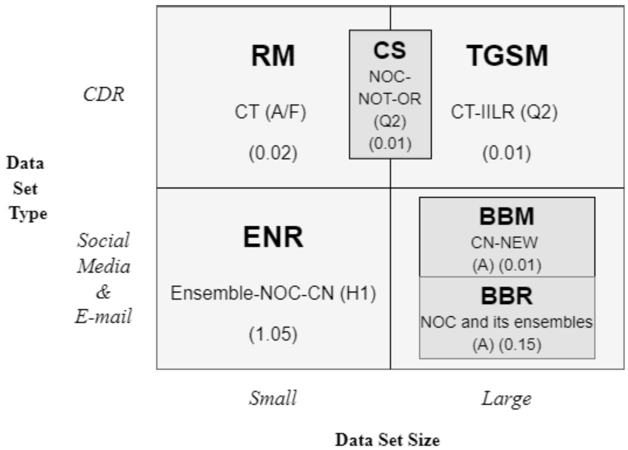


**FIGURE 3.** Data matrix to summarize data set based analysis.

(given in parenthesis). The denoted threshold value is the one that gives the highest f-measure for that case. Notice that CS data set is medium in size, and hence it is placed separately.

As seen in the figure, the best technique varies with respect to the nature of the data set in terms of size and type. It is observed that ensemble of tracking the change in the number of communities and tracking the change in the central nodes gives the best results for two cases, particularly for ENR and CS data sets.

Additionally, the thresholds used for the algorithms vary with respect to the data set, as well. For example, for CT based method on TGSM data set, a low threshold value of 0.01 gives the best result. On the other hand, for ENR data set, a much higher change threshold value of 1.05 gives the best result. Apparently, the ground truth events in ENR data set lead to a much stronger change in the community structures within communication networks. Another interesting observation is that, for Boston Bombing data set, different threshold values are set for mention (BBM) and retweet (BBR) communication networks. Since higher threshold values are used for BBR data set, we can deduce that retweet communication reflects the events more strongly.

## V. RELATED WORK

In the literature, there is a variety of studies that focus on detecting events from data collections. Some of them take the textual content into account and aim to detect events by analyzing the textual content [5], [16], [24], [25], [38], whereas some others focus on detecting events in temporal graphs and analyze the graph properties [2], [3], [20], [27]. A subset of these studies first detect the communities in temporal graphs and track the community based features to detect events in temporal graphs [2], [3], [20].

Community detection is also a popular and recent research problem [4], [7], [23], [34], [36]. In all these studies, different ways are used to model and extract community structures. They can be grouped under the categories of modularity, compression, significance, and diffusion based community detection methods [22]. In this work, dSLM and InfoMap are used as community detection algorithms. dSLM is a modularity based communty detection algorithm, whereas InfoMap is a compression based method.

For temporal graph based event detection, methods presented in [3], [20], and [27] can be considered as related to our study. These methods compute a value for each time interval in temporal graph and aim to detect the time intervals whose values differs from others with respect to a given threshold. The difference between these studies are basically the graph values considered and tracked at each time interval. Some of them use graph features such as eigen-behavior, node degrees [1], [19], [26], [27], whereas some others use community structure based properties [3], or communication trends such as inter-intra ratio links based on initial community structure [20]. These studies also differ in the way of computing the change and comparion. Some of them consider the change with respect to the previous time interval [3], while the others consider mean and standard deviation for comparison [20].

Rayana et. al. propose an ensemble of five basic graph feature based event detection methods [27]. These five basic methods are eigen-behavior based event detection (EBED), probabilistic time series anomaly detection (PTSAD) [1], Streaming Pattern DIscoveRy in multIple Time-Series (SPIRIT) [26], anomalous subspace based event detection (ASED) [19], and moving-average based event detection (MAED). They make experiments on Reality Mining data sets and provide the average precision values. This study also makes experiments on these data sets, provides average precision values, and compares the results based on both event detection and execution time performance.

Aktunc et. al. consider event detection as a problem of change detection in community structures [3]. Particularly, communities extracted from communication network are focused on, and various versions of the community change detection methods are developed using different models. Ensemble methods combining the change models are proposed and their event detection performances are analyzed, as well. They run experiments on Reality Mining data sets for both basic and ensemble methods. This study extends and stabilizes their methodology by including bucket size and resolution concepts, as well as modifying the central nodes based change detection approach. Additionally, the analyses are conducted on broader range of data sets.

Moriano et. al. propose that events can be detected by tracking the change in inter and intra community communications in temporal graphs [20]. The authors also introduce the resolution concept which expands granularity of time intervals. The experiments are conducted on Enron and Boston Bombing tweets data sets. The authors provide precision recall curves for analysis on Enron data sets, however analysis on Boston Bombing tweets data sets is presented as a case study. This study also performs experiments on both Enron and Boston Bombing tweets data sets and compares the results in terms of event detection and execution time performance.

Bommakanti et. al. focus on community mining in temporal graphs and track the evolution of communities to detect events in temporal graphs [8]. The authors detect communities at each time step, identify similar communities through the time steps, and track their evolution. In order to detect event time steps, they compare communities against those in the previous time interval. The community changes can be in the form of appear (new community born), disappear (community lost), merge, split, and survive. They use Louvain algorithm which is a modularity based community detection algorithm [7]. Since identifying communities through time steps depends on similarity check, various similarity thresholds are experimented with. The study focuses on the individual community level events and not the global events. Thus, the work does not include results on network level event detection performance. Therefore, this method could not be used as a baseline for comparison.

Zhu e. al. approaches community detection based event detection differently [39]. Instead of detecting communities at each time interval, the authors consider each time interval as nodes of a larger network and detect communities in that large network. First, they extract graph features for the network of each time steps and thus construct a feature vector per time step. Then, they pairwise similarity of feature vectors are calculate to determine edge between them. Then, communities are detected on within in this network, a community label is assigned to each node. When the nodes are sorted in time order with respect to their community labels, if a change is detected in the community label, that time step is considered as change point (event). They experiment their methods on four small data sets and provide precision, recall and f1 measure values. Since their approach are considerably different than our methods and the results are obtained for only small data sets, it is not included as a baseline study in our experiments.

This study contributes to temporal graph based event detection in terms of both community structure and communication trend based change detection. In community structure based approach, the bucket and resolution concepts are proposed. Additionally, central node based change tracking approach is revised and modified by using PageRank for determining central nodes. In communication trend based approaches, using modularity besides IILR as communication trend metric is proposed. Additionally, change computation method is modified such that the graph value is compared against the previous time interval. The data set variety is also extended including one small, and one large social media data set, and one small, and one large CDR data sets. This enables us to have observations on which methods are more effective on which kind of data sets.

## VI. CONCLUSION

In this work, event detection methods that track changes in community structures and communication trends are proposed. The focus of the study is on temporal graphs corresponding to social interaction and communication among users, and the changes occurring in the communication graph is considered to denote an event.

The proposed event detection algorithms are basically in two main approaches. The first one tracks the changes in the community structures between consecutive time steps. There are several indicators whose change in the community structure can denote an event, such as number of communities, and central nodes. Therefore, the proposed methods in this group consider the change in the number of communities and central nodes as the event indicators. The change in central nodes is defined in four different ways, which are considering all central nodes, the number of newly introduced ones, the number of central nodes not anymore existing as central nodes, and the sum of the number of newly introduced and the not anymore existing central nodes. The second main approach is to track the changes in communication trends within consecutive time steps. In this group, various indicators including the change in the inter-intra communication ratio and modularity of the graph are used. There are also proposed variations as to how these indicators are measured. Either the difference in values within consecutive time steps or the standard deviation values are examined in order to detect an event.

Another novelty proposed by the study is the use of the *buckets*, which denote a set of communities of certain size range. To this aim, 8 different buckets (community size ranges) are defined. Community structure based methods are performed with different buckets (hence considering only communities of certain sizes), and the event detection performance are analyzed with respect to the buckets.

The results of the experiments show that the proposed methods are scalable and can be used on a wide range of social interaction and communication data. According to the results, the proposed methods execute faster than the graph feature based event detection methods of [27]. The proposed methods generally provide better accuracy than the method of Moriano et. al [20] for most of the data sets that they experimented on. Focusing on certain community size ranges improves both speed and accuracy of the proposed community structure based event detection methods. It is observed that communication trend based methods perform better on mobile phone communication data sets while the community structure based methods perform better on social media communication data sets. As it can be inferred from this observation, mobile phone communications react more dynamically to events while the social media communications react more structurally to events.

As future work, the proposed methods can be enhanced in various aspects. As one of the research directions, the effect of using different community detection algorithms, other than dSLM and InfoMap, can be analyzed within community structure based methods. In the study, the change in the modularity is proposed as an event indicator to track in communication trend based methods. Modularity reflects the quality of the community structure of a network and it is hypothesized that a change in the quality of community structure can indicate an event. As a future study, besides modularity, other community structure quality indicators, such as the number of intra-edges, contraction, the number of inter-edges, expansion, conductance [37], modularity density [10], community score, and community fitness [18] can be used as the event indicators to track in the communication trend based event detection methods. In the presented study, it is observed that the performance of the proposed event detection methods may depend on the nature of the data set and the bucket size to be used. The proposed study can be further extended towards a meta learning such that data and the results of the conducted analysis can be used to construct a machine learning model to determine the optimal event detection method and the bucket size, as well as other parameters, for a given data set. As another future work, the proposed methods can be enhanced to work on streaming temporal graphs in near real time. For this, in addition to the use of dynamic community detection methods, such as dSLM, event detection methods should be enhanced to perform on the fly.

## ACKNOWLEDGMENT

The authors would like to thank Dr. Moriano and Prof. Ahn for sharing the Boston Bombing data set and the source code of their method with us [20].

## REFERENCES

[1] L. Akoglu and C. Faloutsos, "Event detection in time series of mobile communication graphs," in *Proc. Army Sci. Conf.*, 2010, pp. 1–8.

[2] R. Aktunc, I. H. Toroslu, and P. Karagoz, "Event detection by change tracking on community structure of temporal networks," in *Proc. IEEE/ACM Int. Conf. Adv. Social Netw. Anal. Mining (ASONAM)*, Aug. 2018, pp. 928–931.

[3] R. Aktunc, I. H. Toroslu, and P. Karagoz, *Event Detection on Communities: Tracking the Change in Community Structure Within Temporal Communication Networks*. Cham, Switzerland: Springer, 2020, pp. 75–96, doi: 10.1007/978-3-030-33698-1_5.

[4] R. Aktunc, I. H. Toroslu, M. Ozer, and H. Davulcu, "A dynamic modularity based community detection algorithm for large-scale networks: DSLM," in *Proc. IEEE/ACM Int. Conf. Adv. Social Netw. Anal. Mining*, Aug. 2015, pp. 1177–1183.

[5] F. Atefeh and W. Khreich, "A survey of techniques for event detection in Twitter," *Comput. Intell.*, vol. 31, no. 1, pp. 132–164, Feb. 2015.

[6] P. H. Barros, I. Cardoso-Pereira, H. Allende-Cid, O. A. Rosso, and H. S. Ramos, "Leveraging phase transition of topics for event detection in social media," *IEEE Access*, vol. 8, pp. 70505–70518, 2020.

[7] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *J. Stat. Mech., Theory Exp.*, vol. 2008, no. 10, Oct. 2008, Art. no. P10008.

[8] S. A. S. R. Bommakanti and S. Panda, "Events detection in temporally evolving social networks," in *Proc. IEEE Int. Conf. Big Knowl. (ICBK)*, Nov. 2018, pp. 235–242, doi: 10.1109/ICBK.2018.00039.

[9] S. Brin and L. Page, "The anatomy of a large-scale hypertextual web search engine," *Comput. Netw. ISDN Syst.*, vol. 30, nos. 1–7, pp. 107–117, Apr. 1998.

[10] M. Chen, T. Nguyen, and B. K. Szymanski, "On measuring the quality of a network community structure," in *Proc. Int. Conf. Social Comput.*, Sep. 2013, pp. 122–127. [Online]. Available: http://dblp.uni-trier.de/db/conf/socialcom/socialcom2013.html#ChenNS13

[11] R. K. Darst, C. Granell, A. Arenas, S. Gómez, J. Saramäki, and S. Fortunato, "Detection of timescales in evolving complex systems," *Sci. Rep.*, vol. 6, no. 1, pp. 1–8, Dec. 2016, doi: 10.1038/srep39713.

[12] J. Diesner, T. L. Frantz, and K. M. Carley, "Communication networks from the enron email corpus 'it's always about the people. Enron is no different,'" *Comput. Math. Org. Theory*, vol. 11, no. 3, pp. 201–228, Oct. 2005, doi: 10.1007/s10588-005-5377-0.

[13] Y. Dong, F. Pinelli, Y. Gkoufas, Z. Nabi, F. Calabrese, and N. V. Chawla, "Inferring unusual crowd events from mobile phone call detail records," 2015, *arXiv:1504.03643*.

[14] N. Eagle, A. Pentland, and D. Lazer, "Inferring friendship network structure by using mobile phone data," *Proc. Nat. Acad. Sci. USA*, vol. 106, no. 36, pp. 15274–15278, Sep. 2009. [Online]. Available: https://www.pnas.org/content/106/36/15274

[15] S. Fortunato, "Community detection in graphs," *Phys. Rep.*, vol. 486, nos. 3–5, pp. 75–174, Feb. 2010.

[16] M. Imran, C. Castillo, F. Diaz, and S. Vieweg, "Processing social media messages in mass emergency: Survey summary," in *Proc. Companion Web Conf. Web Conf. (WWW)*, 2018, pp. 507–511.

[17] I. A. Karatepe and E. Zeydan, "Anomaly detection in cellular network data using big data analytics," in *Proc. Eur. Wireless 20th Eur. Wireless Conf.*, May 2014, pp. 1–5.

[18] S. Kaur, S. Singh, S. Kaushal, and A. K. Sangaiah, "Comparative analysis of quality metrics for community detection in social networks using genetic algorithm," *Neural Netw. World*, vol. 26, no. 6, pp. 625–641, 2016, doi: 10.14311/NNW.2016.26.036.

[19] A. Lakhina, M. Crovella, and C. Diot, "Diagnosing network-wide traffic anomalies," in *Proc. Conf. Appl., Technol., Archit., Protocols Comput. Commun. (SIGCOMM)*, vol. 34, Oct. 2004, pp. 219–230.

[20] P. Moriano, J. Finke, and Y.-Y. Ahn, "Community-based event detection in temporal networks," *Sci. Rep.*, vol. 9, no. 1, pp. 1–9, Dec. 2019.

[21] M. E. J. Newman, "Modularity and community structure in networks," *Proc. Nat. Acad. Sci. USA*, vol. 103, no. 23, pp. 8577–8582, Feb. 2006.

[22] G. K. Orman and V. Labatut, "A comparison of community detection algorithms on artificial networks," in *Discovery Science. DS 2009* (Lecture Notes in Computer Science), vol. 5808, J. Gama, V. S. Costa, A. M. Jorge, and P. B. Brazdil, Eds. Berlin, Germany: Springer, 2009. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-642-04747-3_20#citeas, doi: 10.1007/978-3-642-04747-3_20.

[23] G. K. Orman, V. Labatut, and H. Cherifi, "Comparative evaluation of community detection algorithms: A topological approach," 2012, *arXiv:1206.4987*.

[24] O. Ozdikis, P. Karagoz, and H. Oğuztüzün, "Incremental clustering with vector expansion for online event detection in microblogs," *Social Netw. Anal. Mining*, vol. 7, no. 1, pp. 1–17, Dec. 2017.

[25] O. Ozdikis, P. Senkul, and H. Oguztuzun, "Semantic expansion of tweet contents for enhanced event detection in Twitter," in *Proc. IEEE/ACM Int. Conf. Adv. Social Netw. Anal. Mining*, Aug. 2012, pp. 20–24.

[26] S. Papadimitriou, J. Sun, and C. Faloutsos, "Streaming pattern discovery in multiple time-series," in *Proc. 31st Int. Conf. Very Large Data Bases*, 2005, pp. 697–708.

[27] S. Rayana and L. Akogli, "Less is more: Building selective anomaly ensembles," *ACM Trans. Knowl. Discovery From Data*, vol. 10, no. 4, pp. 42:1–42:33, May 2016.

[28] M. Rosvall and C. T. Bergstrom, "Maps of random walks on complex networks reveal community structure," *Proc. Nat. Acad. Sci. USA*, vol. 105, no. 4, pp. 1118–1123, Jan. 2008. [Online]. Available: https://www.pnas.org/content/105/4/1118

[29] T. Sakaki, M. Okazaki, and Y. Matsuo, "Earthquake shakes Twitter users: Real-time event detection by social sensors," in *Proc. 19th Int. Conf. World Wide Web (WWW)*, 2010, pp. 851–860.

[30] J. Sankaranarayanan, H. Samet, B. E. Teitler, M. D. Lieberman, and J. Sperling, "TwitterStand: News in tweets," in *Proc. 17th ACM SIGSPATIAL Int. Conf. Adv. Geographic Inf. Syst. (GIS)*, 2009, pp. 42–51.

[31] R. Shang, W. Zhang, L. Jiao, X. Zhang, and R. Stolkin, "Dynamic immunization node model for complex networks based on community structure and threshold," *IEEE Trans. Cybern.*, vol. 52, no. 3, pp. 1539–1552, Mar. 2022, doi: 10.1109/TCYB.2020.2989427.

[32] R. Shang, W. Zhang, J. Zhang, J. Feng, and L. Jiao, "Local community detection based on higher-order structure and edge information," *Phys. A, Stat. Mech. Appl.*, vol. 587, Feb. 2022, Art. no. 126513. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S037843712100786X

[33] R. Shang, W. Zhang, J. Zhang, L. Jiao, Y. Li, and R. Stolkin, "Local community detection algorithm based on alternating strategy of strong fusion and weak fusion," *IEEE Trans. Cybern.*, early access, Mar. 25, 2022, doi: 10.1109/TCYB.2022.3159584.

[34] L. Tang and H. Liu, "Community detection and mining in social media," *Synth. Lectures Data Mining Knowl. Discovery*, vol. 2, no. 1, pp. 1–137, Jan. 2010, doi: 10.2200/S00298ED1V01Y201009DMK003.

[35] V. A. Traag, A. Browet, F. Calabrese, and F. Morlot, "Social event detection in massive mobile phone data using probabilistic location inference," in *Proc. IEEE 3rd Int. Conf. Privacy, Secur., Risk Trust IEEE 3rd Int. Conf. Social Comput.*, Oct. 2011, pp. 625–628.

[36] L. Waltman and N. J. van Eck, "A smart local moving algorithm for large-scale modularity-based community detection," 2013, *arXiv:1308.6604*.

[37] J. Yang and J. Leskovec, "Defining and evaluating network communities based on ground-truth," 2012, *arXiv:1205.6233*.

[38] X. Zhou and L. Chen, "Event detection over Twitter social media streams," *VLDB J.*, vol. 23, no. 3, pp. 381–400, Jun. 2014.

[39] T. Zhu, P. Li, L. Yu, K. Chen, and Y. Chen, "Change point detection in dynamic networks based on community identification," *IEEE Trans. Netw. Sci. Eng.*, vol. 7, no. 3, pp. 2067–2077, Jul. 2020.

**RIZA AKTUNC** was born in Kayseri, Turkey, in 1989. He received the B.S. degree from the Department of Computer Science, Bilkent University, Turkey, in 2011, and the master's degree from the Department of Computer Engineering, Middle East Technical University (METU), Turkey, in 2015, where he is currently pursuing the Ph.D. degree.

He worked as a Software Engineer at TUBITAK, from 2011 to 2018. Then, he worked as a Senior Software Engineer and the Team Leader at Bayzat, from 2018 to 2020. From 2020 to 2022, he worked as a Vendor Senior Software Engineer at Microsoft, where he is currently working as a full-time Senior Software Engineer. He has five publications, which are published during his master's and Ph.D. degree studies. His research interests include data mining, information retrieval, and intelligent data analysis.

**PINAR KARAGOZ** (Member, IEEE) received the Ph.D. degree from the Department of Computer Engineering, Middle East Technical University (METU). During her Ph.D. studies, she worked as a Researcher at Stony Brook University, Stony Brook, NY, USA. She had research visits at MIT CSAIL, University of Ostrava, Aalto University, and Southern Denmark University. She is currently a Full Professor with the Department of Computer Engineering, METU. She has publications in internationally recognized and indexed international journals, including IEEE Transactions on Knowledge and Data Engineering, *ACM Transactions on the Web* (TWEB), IEEE Transactions on Industrial Informatics, and *The Computer Journal*. She has about 100 papers in international conferences in the area. Her research interests include data mining, machine learning algorithms, information retrieval, and social media analysis and mining. In 2016, she received the Best Paper Award in IEEE Transactions on Industrial Informatics. In 2017, her article was nominated for Wilkes Award of *The Computer Journal*.

**ISMAIL HAKKI TOROSLU** received the B.S. degree in computer engineering from Middle East Technical University (METU), Ankara, in 1987, the M.S. degree in computer engineering from Bilkent University, Ankara, in 1989, and the Ph.D. degree from the Department of Electrical Engineering and Computer Science, Northwestern University, IL, USA, in 1993. He has been with the Department of Computer Engineering, METU, since 1993. He was a Visiting Professor with the Department of Computer Science, University of Central Florida, from 2000 to 2002. His current research interests include data mining, information retrieval, and intelligent data analysis. He has published more than 90 technical papers in variety of areas of computer science. He has also received IBM Faculty Award, in 2010.