Less is More: Building Selective Anomaly Ensembles
with Application to Event Detection in Temporal Graphs

Shebuti Rayana
Stony Brook University
srayana@cs.stonybrook.edu

Leman Akoglu
Stony Brook University
leman@cs.stonybrook.edu

Abstract

Ensemble techniques for classification and clustering have long proven effective, yet anomaly ensembles have been barely studied. In this work, we tap into this gap and propose a new ensemble approach for anomaly mining, with application to event detection in temporal graphs. Our method aims to combine results from heterogeneous detectors with varying outputs, and leverage the evidence from multiple sources to yield better performance. However, trusting *all* the results may deteriorate the overall ensemble accuracy, as some detectors may fall short and provide inaccurate results depending on the nature of the data in hand. This suggests that being *selective* in which results to combine is vital in building effective ensembles—hence "less is more".

In this paper we propose SELECT; an ensemble approach for anomaly mining that employs novel techniques to automatically and systematically select the results to assemble in a fully unsupervised fashion. We apply our method to event detection in temporal graphs, where SELECT successfully utilizes five base detectors and seven consensus methods under a unified ensemble framework. We provide extensive quantitative evaluation of our approach on five real-world datasets (four with ground truth), including Enron email communications, New York Times news corpus, and World Cup 2014 Twitter news feed. Thanks to its selection mechanism, SELECT yields superior performance compared to individual detectors alone, the full ensemble (naively combining all results), and an existing diversity-based ensemble.

1 Introduction

Ensemble methods utilize multiple algorithms to obtain better performance than the constituent algorithms alone and produce more robust results [5]. Thanks to these advantages, a large body of research has been devoted to ensemble learning in classification [13, 21, 24, 27] and clustering [8, 11, 12, 26]. On the other hand, building effective ensembles for anomaly detection has proven to be a challenging task [1, 28]. A key challenge is the lack of ground-truth; which makes it hard to measure detector accuracy and to accordingly select accurate detectors to combine, unlike in classification. Moreover, there exist no objective or 'fitness' functions for anomaly mining, unlike in clustering.

Existing attempts for anomaly ensembles either combine outcomes from all the constituent detectors [9, 10, 16, 19], or induce diversity among their detectors to increase the chance that they make independent errors [25, 29]. However, as our prior work [22] suggests, neither of these strategies would work well in the presence of inaccurate detectors. In particular, combining all, including inaccurate results would deteriorate the overall ensemble performance. Similarly, diversity-based ensembles would combine inaccurate results for the sake of diversity.

In this work, we tap into the gap between anomaly mining and ensemble methods, and propose SELECT, one of the first *selective* ensemble approaches for anomaly detection. As the name implies, the key property of our ensemble is its selection mechanism which carefully decides which results to combine from multiple different methods in the ensemble. We summarize our contributions as follows.

- We identify and study the problem of building selective anomaly ensembles in a fully unsupervised fashion.
- We propose SELECT, a new ensemble approach for anomaly detection, which utilizes not only multiple heterogeneous detectors, but also various consensus methods under a unified ensemble framework.
- SELECT employs two novel unsupervised selection strategies that we design to choose the detector/consensus results to combine, which render the ensemble not only more robust but improve its performance further over its non-selective counterpart.
- Our ensemble approach is general and flexible. It does not rely on specific data types, and allows other detectors and consensus methods to be incorporated.

We apply our ensemble approach to the event detection problem in temporal graphs, where SELECT utilizes five heterogeneous event detection algorithms and seven different consensus methods. Extensive evaluation on datasets with ground truth shows that SELECT outperforms the average individual detector, the full ensemble that naively combines all results, as well as the diversity-based ensemble in [25].

2 Background and Preliminaries

2.1 Event Detection Problem Temporal graphs change dynamically over time in which new nodes and edges arrive or existing nodes and edges disappear. Many dynamic systems can be modeled as temporal graphs, such as computer, trading, transaction, and communication networks.

Event detection in temporal graph data is the task of finding the points in time at which the graph structure notably differs from its past. These change points may correspond to significant events; such as critical state changes, anomalies, faults, intrusion, etc. depending on the application domain. Formally, the problem can be stated as follows.

Given a sequence of graphs $\{G_1, G_2, \ldots, G_t, \ldots, G_T\}$;
Find time points t' s.t. $G_{t'}$ differs significantly from $G_{t'-1}$.

2.2 Motivation for Ensembles Several different methods have been proposed for the above problem, a survey of which is given in [3]. To date, however, there exists no single method that has been shown to outperform all the others. The lack of a winner technique is not a freak occurrence. In fact, it is unlikely that a given method could perform consistently well on different data of varying nature. Further, different techniques may identify different classes or types of anomalies depending on their particular formulation. This suggests that effectively *combining* the results from various different detection methods (detectors from here onwards) could help improve the detection performance.

2.3 Motivation for Selective Ensembles Ensembles are expected to perform superior to their average constituent detector, however a naive ensemble that trusts results from *all* detectors may not work well. The reason is, some methods may not be as effective as desired depending on the nature of the data in hand, and fail to identify the anomalies of interest. As a result, combining accurate results with inaccurate ones may deteriorate the overall ensemble performance [22]. This suggests that *selecting* which detectors to assemble is a critical aspect of building robust ensembles—which implies that "less is more".

To illustrate the motivation for (selective) ensemble building further, consider the example in Figure 1. The rows show the anomaly scores assigned by five different detectors to time points in the Enron Inc.'s time line. Notice that the scores are of varying nature and scale, due to different formulations of the detectors. We realize that the detectors mostly agree on the events that they detect; e.g., 'J. Skilling new CEO'. On the other hand, they assign different magnitude of anomalousness to the time points; e.g., the top anomaly of methods varies. These suggest that combining the outcomes could help build improved ranking of the anomalies. Next notice the result provided by "Probabilistic Approach" which, while identifying one major event also detected by other detectors, fails to provide a reliable ranking for the rest; e.g., it scores many other time points higher than 'F. Cooper new CEO'. As such, including this detector in the ensemble is likely to deteriorate the overall performance.

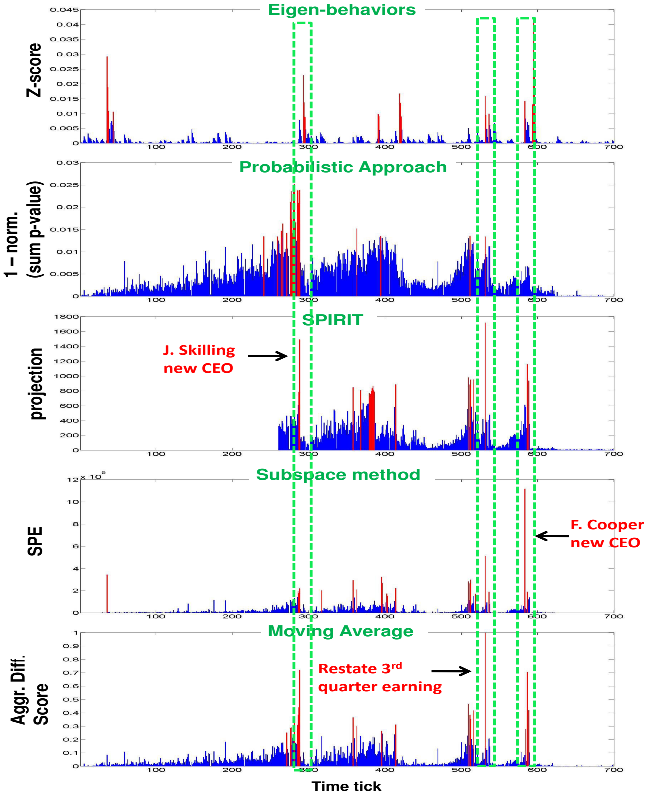

Figure 1: Anomaly scores from five detectors (rows) for the Enron Inc. time line. Red bars depict top 20 anomalous time points.

In summary, inspired by the success of classification and clustering ensembles and driven by the limited work on anomaly ensembles, we aim to systematically combine the strengths of accurate detectors while alleviating the weaknesses of the less accurate ones to build selective detection ensembles for anomaly mining. While we build ensembles for the event detection problem in this paper, our approach is general and can directly be employed on a collection of detection methods for other anomaly mining problems.

3 SELECT: Selective Ensemble Learning for anomaly detECTion — Application to Event Detection

3.1 Overview Our SELECT approach takes the input data, in this case a sequence of graphs $\{G_1, \ldots, G_t, \ldots, G_T\}$, and outputs a rank list R of objects, in this case of time points $1 \leq t \leq T$, ranked from most to least anomalous.

The main steps of SELECT are given in Algorithm 1. Step 1 employs (five) different event detection algorithms as base detectors of the ensemble. Each detector has a specific and different measure to score the individual time points by anomalousness. As such, the ensemble embodies heterogeneous detectors. As motivated earlier, Step 2 selects a subset of the detector results to assemble through a proposed selection strategy. Step 3 then combines the selected results into a consensus. Besides several different event detection algorithms, there also exist various different consensus finding approaches. In spirit of building ensembles, SELECT also leverages (seven) different consensus techniques to create intermediate aggregate results. Similar to Step 2, Step 4 then selects a subset of the consensus results to assemble. Finally, Step 5 combines this subset into the final rank list of time points using inverse rank aggregation (Section 3.3).

```
Algorithm 1 SELECT
Input: Data: graph sequence ${G_1, ..., G_t, ..., G_T}$
Output: Rank list of objects (time points) by anomaly
1: Obtain results from (5) base detectors
2: Select set $E$ of detectors to assemble
3: Combine $E$ by (7) consensus techniques
4: Select set $C$ of consensus results to assemble
5: Combine $C$ into final rank list
```

Different from prior works, (i) SELECT is a *two-phase* ensemble that not only leverages multiple detectors but also multiple consensus techniques, and (ii) it employs novel strategies to carefully select the ensemble components to assemble without any supervision, which outperform naive (no selection) and diversity-based selection (Section 4). Moreover, (iii) SELECT is the first ensemble method for event detection in temporal graphs, although the same general framework as presented in Algorithm 1 can be deployed for other anomaly mining tasks, where the base detectors are replaced with a set of algorithms for the particular task at hand.

Next we fill in the details on the three main components of the proposed SELECT ensemble. In particular, we describe the base detectors (Section 3.2), consensus techniques (Section 3.3), and the selection strategies (Section 3.4).

3.2 Base Detectors There exist various methods for the event detection problem in temporal graphs [3]. In this work SELECT employs five base detectors (Algorithm 1, Line 1), while one can easily expand the ensemble with others: (1) eigen-behavior based event detection (EBED) from our prior work [2], (2) probabilistic time series anomaly detection (PTSAD) we developed recently [22], (3) Streaming Pattern DIscoveRy in multIple Time-Series (SPIRIT) by Papadimitriou *et al.* [20], (4) anomalous subspace based event detection (ASED) by Lakhina *et al.* [18], and (5) moving-average based event detection (MAED). All methods extract graph-centric features (e.g., degree) for all nodes over time and detect events in multi-variate time series. We refer to [23] for the descriptions of these methods due to space limit.

3.3 Consensus Finding Our ensemble consists of heterogeneous detectors. That is, the detectors employ different anomaly scoring functions and hence their scores may vary in range and interpretation (see Figure 1). Unifying these various outputs to find a consensus among detectors is an essential step toward building an ensemble.

A number of different consensus finding approaches have been proposed in the literature, which can be categorized into two, as rank based and score based aggregation methods. Without choosing one over the other, we utilize seven well-established methods as we describe below.

Rank based consensus. Rank based methods use the anomaly scores to order the data points (here, time points) into a rank list. This ranking makes the algorithm outputs comparable and facilitates combining them. Merging multiple rank lists into a single ranking is known as rank aggregation, which has a rich history in theory of social choice and information retrieval [6]. SELECT employs three rank based consensus methods. *Kemeny-Young* [14] is a voting technique that uses preferential ballot and pair-wise comparison counts to combine multiple rank lists, in which the detectors are treated as voters and the points as the candidates they vote for. *Robust Rank Aggregation* (RRA) [15] utilizes order statistics to compute the probability that a given ordering of ranks for a point across detectors is generated by the null model where the ranks are sampled from a uniform distribution. The final ranking is done based on this probability, where more anomalous points receive a lower probability. The third approach is based on *Inverse Rank* aggregation, in which we score each point by $\frac{1}{r_i}$ where r_i denotes its rank by detector i and average these scores across detectors based on which we sort the points into a final rank list.

Score based consensus. Rank-based aggregation provides a crude ordering of the data points, as it ignores the actual anomaly scores and their spacing. For instance, quite different rankings can yield equal performance in binary decision. Score-based aggregation approaches tackle the calibration of different anomaly scores and unify them within a shared range. SELECT employs two score based consensus methods. *Mixture Modeling* [10] converts the anomaly scores into probabilities by modeling them as sampled from a mixture of exponential (for inliers) and Gaussian (for outliers) distributions. *Unification* [16] also converts the scores into probability estimates through regularization, normalization, and scaling steps. The probabilities are then comparable across detectors, which we aggregate by both max and avg. This yields four score based methods.

3.4 Ensemble Learning Given different base detectors and various consensus methods, the final task remains to utilize them under a unified ensemble framework. In this section, we discuss four different approaches for building anomaly ensembles. These approaches differ in whether and how they select their ensemble components.

3.4.1 Full ensemble The full ensemble selects all the detector results (Step 2 of Alg.1) and later all the consensus results (Step 4 of Alg.1) to aggregate at both phases of SELECT. As such, it is a naive approach that is prone to obtain inferior results in the presence of inaccurate detectors.

3.4.2 Selective ensembles As motivated earlier in Section 2.3, carefully selecting which detectors to assemble in Step 2 may help prevent the final ensemble from going astray, provided that some base detectors may fail to reliably identify the anomalies of interest to a given application. Similarly, pruning away consensus results that may be noisy in Step 4 could help reach a stronger final consensus. In anomaly mining, however, it is challenging to identify the components with inferior results given the lack of ground truth to estimate their generalization errors externally. In this section, we present two orthogonal selection strategies that leverage internal clues across detectors or consensuses and work in a fully unsupervised fashion: (i) a vertical strategy that exploits correlations among the results, and (ii) a horizontal strategy that uses order statistics to filter out far-off results.

Strategy I: Vertical Selection. Our first approach to selecting the ensemble components is through correlation analysis among the score lists from different methods, based on which we successively enhance the ensemble one list at a time (hence vertical). The work flow of the vertical selection strategy is given in Algorithm 2.

Given a set of anomaly score lists S, we first unify the scores by converting them to probability estimates using *Unification* [16]. Then we average the probability scores across lists to construct a $target$ vector, which we treat as the "pseudo ground-truth" (Lines 1-6).

We initialize the ensemble E with the list $l \in S$ that has the highest weighted Pearson correlation to $target$. In computing the correlation, the weights we use for the list elements are equal to $\frac{1}{r}$, where r is the rank of an element in $target$ when sorted in descending order, i.e., the more anomalous elements receive higher weight (Lines 7-11).

Next we sort the remaining lists $S \backslash l$ in descending order by their correlation to the current "prediction" of the ensemble, which is defined as the average probability of lists in the ensemble. We test whether adding the top list to the ensemble would increase the correlation of the prediction to $target$. If the correlation improves by this addition, we update the ensemble and reorder the remaining lists by their correlation to the updated prediction, otherwise we discard the list. As such, a list gets either included or discarded at each iteration until all lists are processed (Lines 12-19).

Algorithm 2 Vertical Selection

```
Input: $S$ := set of anomaly score lists
Output: $E$ := ensemble set of selected lists
 1: $P := \emptyset$
 2: /* convert scores to probability estimates */
 3: for each $s \in S$ do
 4:     $P := P \cup Unification(s)$
 5: end for
 6: $target := avg(P)$    /*target vector*/
 7: $r$ := ranklist after sorting $target$ in descending order
 8: $E := \emptyset$
 9: sort $P$ by weighted Pearson ($wP$) correlation to $target$
10: /* in descending order, weights: $\frac{1}{r}$ */
11: $l := fetchFirst(P)$,    $E := E \cup l$
12: while $P \neq \emptyset$ do
13:     $p := avg(E)$    /*current prediction of $E$*/
14:     sort $P$ by $wP$ correlation to $p$   /*descending order*/
15:     $l := fetchFirst(P)$
16:     if $wP(avg(E \cup l), target) > wP(p, target)$ then
17:         $E := E \cup l$    /*select list*/
18:     end if
19: end while
20: return $E$
```

Strategy II: Horizontal Selection. We are interested in finding time points that are ranked high in a set of accurate rank lists (from either base detectors or consensus methods), ignoring a (small) fraction of inaccurate rank lists. Thus, we also present an element-based (hence horizontal) approach for selecting ensemble components.

To identify the accurate lists, this strategy focuses on the anomalous elements. It assumes that the normalized ranks of the anomalies should come from a distribution skewed toward zero. Based on this, lists in which the anomalies are not ranked sufficiently high (i.e., have large normalized ranks) are considered to be inaccurate and voted for being discarded. The work flow of the horizontal selection strategy is given in Algorithm 3.

Similar to the vertical strategy we first identify a "pseudo ground truth", in this case a list of anomalies. In particular, we use *Mixture Modeling* [10] to convert each score list in S into a binary list in which outliers are denoted by 1, and inliers by 0. We then employ majority voting across lists to obtain a final set of target anomalies O (Lines 1-7).

Given that S contains m lists, we construct a normalized rank vector $\mathbf{r} = [r_{(1)}, \ldots, r_{(m)}]$ for each anomaly $o \in O$, such that $r_{(1)} \leq \ldots \leq r_{(m)}$, where $r_{(l)}$ denotes the rank of o in list $l \in S$ normalized by the total number of elements in l. Following similar ideas to *Robust Rank Aggregation* [15], we then compute order statistics based on these sorted normalized rank lists to identify the lists that provide statistically large ranks for each anomaly.

Specifically, for each ordered list l in a given $\mathbf{r}$, we compute how probable it is to obtain $\hat{r}_{(l)} \leq r_{(l)}$ when the ranks $\hat{r}$ are generated by a uniform null distribution. We denote the probability that $\hat{r}_{(l)} \leq r_{(l)}$ by $p_{l,m}(\mathbf{r})$. Under the uniform null model, the probability that $\hat{r}(l)$ is smaller or equal to $r_{(l)}$ can be expressed as a binomial probability

$$p_{l,m}(\mathbf{r}) = \sum_{t=l}^{m} \binom{m}{t} r_{(l)}^{t} (1 - r_{(l)})^{m-t},$$

since at least l normalized rankings drawn uniformly from $[0, 1]$ must be in the range $[0, r_{(l)}]$.

Algorithm 3 Horizontal Selection

```
Input: $S$ := set of anomaly score lists
Output: $E$ := ensemble set of selected lists
 1: $M := \emptyset$ , $R := \emptyset$ , $F := \emptyset$ , $E := \emptyset$
 2: for each $l \in S$ do
 3:     /* label score lists with 1 (outliers) & 0 (inliers) */
 4:     $class := MixtureModel(l)$ ,    $M := M \cup class$
 5:     $R := R \cup ranklist(l)$
 6: end for
 7: $O := majorityVoting(M)$    /*target anomalies*/
 8: $[S_{sort}, pVals] := RobustRankAggregation(R, O)$
 9: for each $o \in O$ do
10:     $m_{ind} := \min(pVals(o, :))$
11:     $F := F \cup S_{sort}(o, (m_{ind} + 1) : end)$
12: end for
13: for each $l \in S$ do
14:     $count$ := number of occurrences of $l$ in $F$
15: end for
16: Cluster non-zero $counts$ into two clusters, $C_l$ and $C_h$
17: $E := S \setminus \{s \in C_h\}$   /* discard high-$count$ lists */
18: return $E$
```

For a sequence of accurate lists that rank the anomalies at the top, and hence that yield low normalized ranks $r_{(l)}$, this probability is expected to drop with the ordering, i.e., for increasing $l \in \{1 \ldots m\}$. An example sequence of p probabilities (y-axis) are shown in Figure 2 for an anomaly based on 20 score lists. The lists are sorted by their normalized ranks of the anomaly on the x-axis. The figure suggests that the 5 lists at the end of the ordering are likely inaccurate, as the ranks of the given anomaly in those lists are larger than what is expected based on the ranks in the other lists.

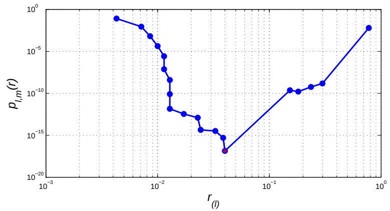

Figure 2: Normalized rank $r_{(l)}$ vs. probability p that $\hat{r}_{(l)} \leq r_{(l)}$, where $\hat{r}$ are drawn uniformly at random from $[0, 1]$.

Based on this intuition, we count the frequency that each list l is ordered *after* the list with $\min_{l=1,\ldots,m} p_{l,m}(\mathbf{r})$ among all the normalized rank lists $\mathbf{r}$ of the target anomalies (Lines 8-15). We then group these counts into two clusters[1] and discard the lists in the cluster with the higher average count (Lines 16-17). This way we eliminate the lists with larger counts, but retain the lists that appear inaccurate only a few times which may be a result of the inherent uncertainty or noise in which we construct the target anomaly set.

3.4.3 Diversity-based ensemble

In classification, two basic conditions for an ensemble to improve over the constituent classifiers are that the base classifiers are (i) accurate (better than random), and (ii) diverse (making uncorrelated errors) [5, 27]. Achieving better-than-random accuracy in supervised learning is not hard, and several studies have shown that ensembles tend to yield better results when there is a significant diversity among the models [4, 17].

Following on these insights, Schubert *et al.* proposed a diversity-based ensemble [25], which is similar to our vertical selection in Alg. 2. The main distinction is the ascending ordering in Lines 9 and 14, which yields a diversity-favored, in contrast to a correlation-favored, selection.[2]

Unlike classification ensembles, however, it is not realistic for anomaly ensembles to assume that all the detectors will be reasonably accurate (i.e., better than random), as some may fail to spot the (type of) anomalies in the given data. In the existence of inaccurate detectors, the diversity-based approach would likely yield inferior results as it is prone to selecting inaccurate detectors for the sake of diversity. As we show in our experiments, too much diversity is in fact bound to limit accuracy for event detection ensembles.

4 Evaluation

We evaluate our selective ensemble approach on the event detection problem using five real-world datasets, both previously used as well as newly collected by us, including email communications, news corpora, and social media. For four of these datasets we compiled ground truths for the temporal anomalies, for which we present quantitative results. We use the remaining data for illustrating case studies.

We compare the performance of SELECT with vertical selection (SelectV), and horizontal selection (SelectH) to that of individual detectors, the full ensemble with no selection (Full), and the diversity-based ensemble (DivE) [25]. This makes ours one of the few works that quantitatively compares and contrasts anomaly ensembles at a scale that includes as many datasets with ground truth.

In a nutshell, our results illustrate that (i) base detectors do not always all produce accurate results, (ii) ensemble approach alleviates the shortcomings of the inaccurate detectors, (iii) a careful selection of ensemble components increases the overall performance, and (iv) introducing noisy results decreases overall ensemble accuracy where the diversity-based ensemble is affected the most.

4.1 Dataset Description

In the following we describe the five real-world temporal graph datasets we used in this work. All datasets with ground truth events are made available at http://shebuti.com/SelectiveAnomalyEnsemble/.

[1] We cluster the counts by k-means clustering with $k = 2$, where the centroids are initialized with the smallest and largest counts, respectively.

[2] There are other differences between our vertical selection (Algorithm 2) and the diversity-based ensemble in [25], such as the construction of the pseudo ground truth and the choice of weights in correlation computation.

Dataset 1: EnronInc. We use four years (1999–2002) of Enron email communications. In the temporal graphs, the nodes represent email addresses and directed edges depict sent/received relations. Enron email network contains a total of $80,884$ nodes. We analyze the data with daily sample rate skipping the weekends (700 time points). The ground truth captures the major events in the company's history, such as CEO changes, revenue losses, restatements of earnings, etc.

Dataset 2: RealityMining Reality Mining is comprised of communication and proximity data of 97 faculty, student, and staff at MIT recorded continuously via pre-installed software on their mobile devices over 50 weeks [7]. From the raw data we built sequences of weekly temporal graphs for three types of relations; voice calls, short messages, and bluetooth scans. For voice call and short message graphs a directed edge denotes an incoming/outgoing call or message, and for bluetooth graphs an edge depicts physical proximity between two subjects. The ground truth captures semester breaks, exam and sponsor weeks, and holidays.

Dataset 3: TwitterSecurity We collect tweet samples using the Twitter Streaming API for four months (May 12–Aug 1, 2014). We filter the tweets containing Department of Homeland Security keywords related to terrorism or domestic security.[3] After named entity extraction and resolution (including URLs, hashtags, @ mentions), we build entity-entity co-mention temporal graphs on daily basis (80 time ticks). We compile the ground truth to include major world news of 2014, such as the Turkey mine accident, Boko Haram kidnapping school girls, killings during Yemen raids, etc.

Dataset 4: TwitterWorldCup Our Twitter collection also spans the World Cup 2014 season (June 12–July 13). This time, we filter the tweets by popular/official World Cup hashtags, such as `#worldcup`, `#fifa`, `#brazil`, etc. Similar to TwitterSecurity, we construct entity-entity co-mention temporal graphs on 5 minute sample rate (8640 time points). The ground truth contains the goals, penalties, and injuries in all the matches that involve at least one of the renowned teams (Brazil, Germany, Argentina, Netherlands, Spain, France).

Dataset 5: NYTNews This corpus contains all of the published articles in New York Times over 7.5 years (Jan 2000–July 2007) (available from https://catalog.ldc.upenn.edu/LDC2008T19). The named entities (people, places, organizations) are hand-annotated by human editors. We construct weekly temporal graphs (390 time points) in which each node corresponds to a named entity and edges depict co-mention relations in the articles. The data contains around $320,000$ entities, however no ground truth events.

Table 1: Accuracy of ensembles for EnronInc. (features: weighted in-/out-degree). ∗ depicts selected detector/consensus results.

		Full	DivE	SelectV	SelectH
Base Algorithms	EBED (win)	0.1313	∗	∗	
	PTSAD (win)	0.1462	∗		
	SPIRIT (win)	0.7032	∗		∗
	ASED (win)	0.5470	∗	∗	∗
	MAED (win)	0.6670			∗
	EBED (wout)	0.2846	∗		
	PTSAD (wout)	0.2118	∗		
	SPIRIT (wout)	0.4563	∗		∗
	ASED (wout)	0.0580	∗		
	MAED (wout)	0.7328		∗	∗
Consensus	Inverse Rank	∗ 0.6829	∗ 0.5660	0.6738	∗ 0.8291
	Kemeny-Young	∗ 0.4086	∗ 0.3703	∗ 0.6586	∗ 0.6334
	RRA	∗ 0.6178	0.4871	0.5686	∗ 0.6590
	Uni (avg)	∗ 0.5292	∗ 0.5511	∗ 0.6375	∗ 0.6207
	Uni (max)	∗ 0.3333	∗ 0.3187	0.4314	∗ 0.7353
	MM (avg)	∗ 0.7513	∗ 0.5726	∗ 0.7663	∗ 0.7530
	MM (max)	∗ 0.0218	∗ 0.0218	0.2108	0.0224
Final Ensemble		0.7082	0.6276	0.7125	**0.7920**

4.2 Event Detection Performance

Next we quantitatively evaluate the ensemble methods on detection accuracy. The final result output by each ensemble is a rank list, based on which we create the precision-recall (PR) plot for a given ground truth. We report the area under the PR plot, namely *average precision*, as the measure of accuracy.

Table 1 shows the accuracies for all four ensemble methods on EnronInc., along with the accuracies of the base detectors and consensus methods. Notice that some detectors yield quite low accuracy (e.g., ASED (wout)) on this dataset. Further, MM (max) consensus provides low accuracy across ensembles no matter which detector results are combined. SELECT ensembles successfully filter out relatively inferior results and achieve higher accuracy. We also note that DivE yields lower performance than all, including Full.

To investigate the significance of the selections made by SELECT ensembles, we compare them to ensembles that randomly select the same number of components to assemble at each phase. In Table 2 we report the average and standard deviation of accuracies achieved by 100 such random ensembles, denoted by RandE, and the gain achieved by SelectV and SelectH over their respective random ensembles.

We show the final anomaly scores of the time points provided by SelectH on EnronInc. for visual analysis in Figure 3. The figure also depicts the ground truth events by vertical (red) lines, which we note to align well with the time points with high scores.

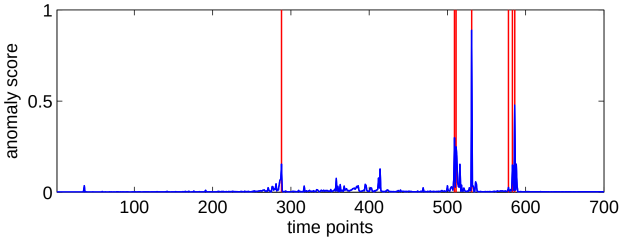

Figure 3: Anomaly scores of time points by SelectH on EnronInc. align well with ground truth (vertical red lines).

[3] http://www.huffingtonpost.com/2012/02/24/homeland-security-manual_n_1299908.html

Table 1 shows results when we use weighted node in-/out-degree features on the directed Enron graphs to construct the input time series for the base detectors. As such, the ensembles utilize 10 components in the first phase. We also build the ensembles using 20 components where we include the unweighted in-/out-degree features. We refer to [23] for all the accuracy results and selections made, a summary of which is provided in Table 2. We notice that the unweighted graph features are less informative and yield lower accuracies across detectors on average. This affects the performance of Full and DivE, where the accuracies drop significantly. On the other hand, SELECT ensembles are able to achieve comparable accuracies with increased significance under the additional noisy input.

Thus far, we used the exact time points of the events to compute precision and recall. In practice, some time delay in detecting an event is often tolerable. Therefore, we also compute the detection accuracy when delay is allowed; e.g., for delay 2, detecting an event that occurred at t within time window $[t-2, t+2]$ is counted as accurate. Figure 4 shows the accuracy for 0 to 5 time point delays (days) for EnronInc., where delay 0 is the same as exact detection. We notice that SELECT ensembles and Full can detect almost all the events within 5 days before or after each event occurs.

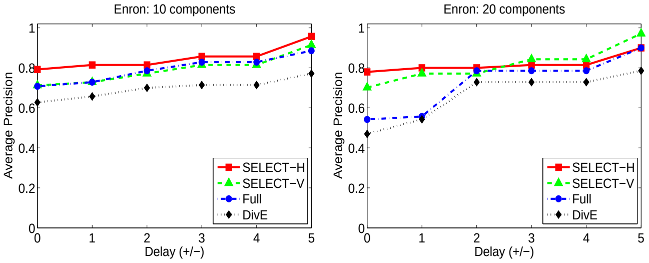

Figure 4: EnronInc. average precision vs. detection delay using (left) 10 components and (right) 20 components.

Next we analyze the results for RealityMining. Similar to EnronInc., we build the ensembles using both 10 and 20 components for the directed Voice Call and SMS graphs. Bluetooth graphs are undirected, as they capture (symmetric) proximity of devices, for which we build ensembles with 10 components using weighted and unweighted degree features. All the details on detector and consensus accuracies as well as selections made are given in [23] due to space limit. We provide the summary of results in Table 2. We note that SELECT ensembles provide superior results to Full and DivE.

Figure 5 illustrates the accuracy-delay plots which show that SELECT ensembles for Bluetooth and SMS detect almost all the events within a week before or after they occur, while the changes in Voice Call are relatively less reflective of the changes in the school year calendar.

Table 2: Significance of accuracy results compared to random ensembles with same number of selected components as SELECT.

	Accuracy	significance
EnronInc. (10 comp.) (Full: 0.7082, DivE: 0.6276)		
(i) RandE (3/10, 3/7)	0.4804 (μ)	0.1757 (σ)
SelectV	0.7125	$= \mu + 1.3210\sigma$
(ii) RandE (5/10, 6/7)	0.5509 (μ)	0.1406 (σ)
SelectH	**0.7920**	$= \mu + 1.7148\sigma$
EnronInc. (20 comp.) (Full: 0.5420, DivE: 0.4697)		
(i) RandE (4/20, 2/7)	0.4047 (μ)	0.1732 (σ)
SelectV	0.7018	$= \mu + 1.7154\sigma$
(ii) RandE (15/20, 6/7)	0.5707 (μ)	0.0864 (σ)
SelectH	**0.7798**	$= \mu + 2.4201\sigma$
RM-VoiceCall (10 comp.) (Full: 0.7302, DivE: 0.8724)		
(i) RandE (2/10, 1/7)	0.7370 (μ)	0.1551 (σ)
SelectV	0.8370	$= \mu + 0.6447\sigma$
(ii) RandE (8/10, 6/7)	0.7653 (μ)	0.0714 (σ)
SelectH	**0.9045**	$= \mu + 1.9496\sigma$
RM-VoiceCall (20 comp.) (Full: 0.8011, DivE: 0.8335)		
(i) RandE (2/20, 2/7)	0.7752 (μ)	0.1494 (σ)
SelectV	0.8847	$= \mu + 0.7329\sigma$
(ii) RandE (17/20, 6/7)	0.8187 (μ)	0.0497 (σ)
SelectH	**0.8949**	$= \mu + 1.5332\sigma$
RM-Bluetooth (10 comp.) (Full: 0.8398, DivE: 0.7735)		
(i) RandE (4/10, 1/7)	0.8269 (μ)	0.1129 (σ)
SelectV	**0.9193**	$= \mu + 0.8184\sigma$
(ii) RandE (8/10, 6/7)	0.8410 (μ)	0.0322 (σ)
SelectH	0.8886	$= \mu + 1.4783\sigma$
RM-SMS (10 comp.) (Full: 0.9092, DivE: 0.8598)		
(i) RandE (4/10, 1/7)	0.8328 (μ)	0.0978 (σ)
SelectV	**0.9283**	$= \mu + 0.9765\sigma$
(ii) RandE (8/10, 6/7)	0.8976 (μ)	0.0620 (σ)
SelectH	0.9217	$= \mu + 0.3887\sigma$
RM-SMS (20 comp.) (Full: 0.9542, DivE: 0.8749)		
(i) RandE (2/20, 1/7)	0.7685 (μ)	0.1521 (σ)
SelectV	0.9294	$= \mu + 1.0579\sigma$
(ii) RandE (17/20, 5/7)	0.9217 (μ)	0.0296 (σ)
SelectH	**0.9621**	$= \mu + 1.3649\sigma$
TwitterSecurity (10 comp.) (Full: 0.5200, DivE: 0.4800)		
(i) RandE (4/10, 1/7)	0.5068 (μ)	0.0755 (σ)
SelectV	0.5467	$= \mu + 0.5285\sigma$
(ii) RandE (9/10, 3/7)	0.5198 (μ)	0.0538 (σ)
SelectH	**0.5867**	$= \mu + 1.2435\sigma$

Finally, we perform event detection using our Twitter datasets. Table 2 includes a summary of results for detecting world news on TwitterSecurity, the details of which can be found in [23]. Results are in agreement with prior ones, where SelectH outperforms the other ensembles. This further becomes evident in Figure 6 (left), where SelectH can detect all the ground truth events within 3 days delay.

The detection dynamics change when TwitterWorldCup is analyzed. The events in this data such as goals and injuries are quite instantaneous (recall the 4 goals in 6 minutes by Germany against Brazil), where we use a sample rate of 5 minutes. Moreover, such events are likely to be reflected on Twitter with some delay by social media users. As such, it is extremely hard to pinpoint the exact time of the events by the ensembles. As we notice in Figure 6 (right), the

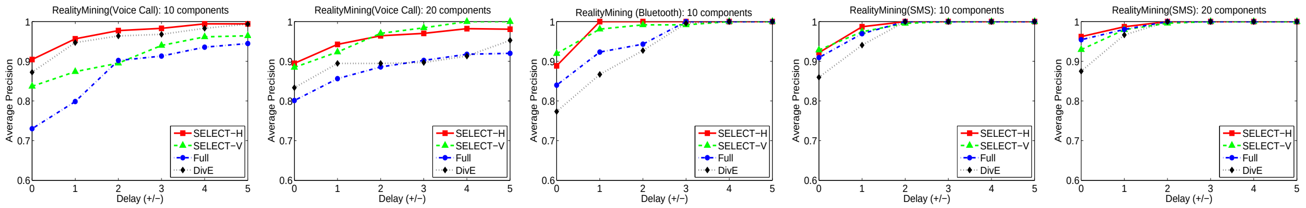

Figure 5: RealityMining average precision vs. detection delay for (left to right) Voice Call (10 comp.), Voice Call (20 comp.), Bluetooth (10 comp.), SMS (10 comp.), and SMS (20 comp.).

initial accuracies at zero delay are quite low. When delay is allowed for up to 288 time points (i.e., one day), the accuracies incline to a reasonable level within half a day delay. In addition, all the detector and consensus results seem to contain signals in this case where most of them are selected by the ensembles, hence comparable accuracies. In fact, DivE selects all of them and performs the same as Full.

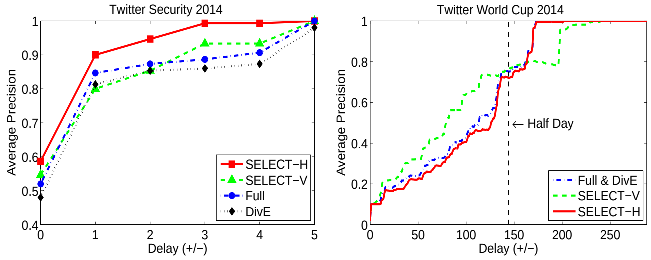

Figure 6: Twitter average precision vs. detection delay for (left) Security and (right) WorldCup 2014.

4.3 Noise Analysis Provided that selecting which results to combine would especially be beneficial in the presence of inaccurate detectors, we design experiments where we introduce increasing number of noisy results into our ensembles. In particular, we create noisy results by randomly shuffling the rank lists output by the base detectors and treat them as additional detector results. Figure 7 shows accuracies (avg.'ed over 10 independent runs) on all of our datasets for 10 component ensembles (results using 20 components are similar, and provided in [23]). We notice that SELECT ensembles provide the most stable and effective performance under increasing number of noisy results. More importantly, these results show that DivE degenerates quite fast in the presence of noise, i.e., when the assumption that all results are reasonably accurate fails to hold.

4.4 Case Studies In this section we evaluate our ensemble approach qualitatively using the NYTNews corpus dataset, for which we do not have a compiled list of ground truth events. Figure 8 shows the anomaly scores for the 2000-2007 time line, provided by the five base detectors using weighted degree feature (we have demonstrated a similar figure for EnronInc. in Figure 1 for additional qualitative analysis).

Top three events by SelectH are marked within boxes in the figure, and corresponds to major events such as the 2001 elections, 9/11 WTC attacks, and the 2003 Columbia Space Shuttle disaster. SelectH also ranks entities by association to a detected event for attribution. We note that for the Columbia disaster, NASA and the seven astronauts killed in the explosion rank at the top. The visualization of the change in Figure 9 shows that a heavy clique with high degree nodes emerges in the graph structure at the time of the event.

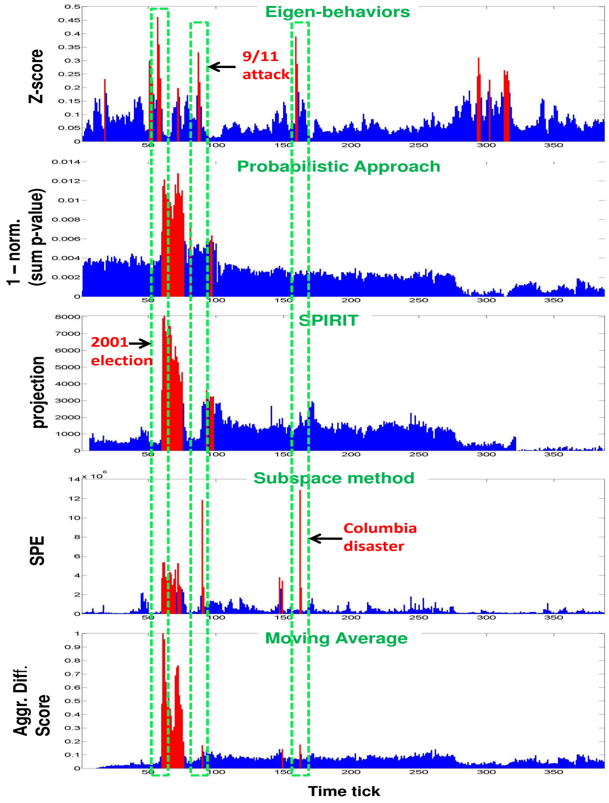

Figure 8: Anomaly scores from five base detectors (rows) for NYT news corpus. Top 3 events by the final ensemble are marked with green boxes. (red bars: top 20 anomalous time points per detector)

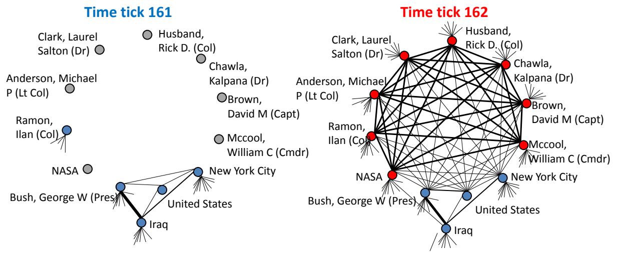

Figure 9: During 2003 Columbia disaster a clique of NASA and the seven killed astronauts emerges from time tick 161 to 162.

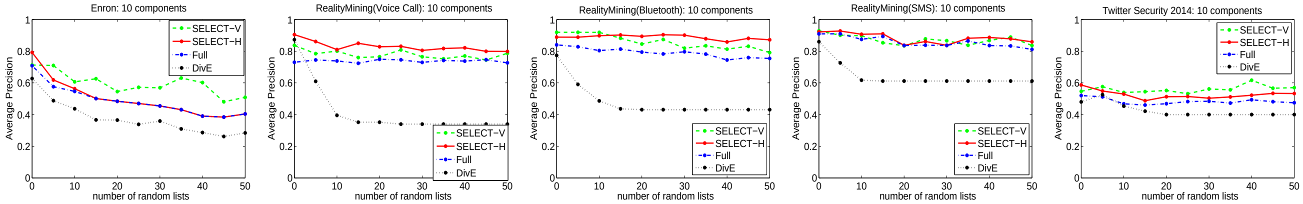

Figure 7: Ensemble accuracies drop when increasing number of random results are added, where decrease is most prominent for DivE.

5 Conclusion

In this work we have proposed SELECT, a new selective ensemble approach for anomaly mining, and applied it to the event detection problem in temporal graphs. SELECT is a two-phase approach that combines multiple detector results and then multiple consensuses, respectively. Motivated by our earlier observations [22] that inaccurate detectors may deteriorate overall ensemble accuracy, we designed two unsupervised selection strategies, SelectV and SelectH, which carefully choose which detector/consensus outcomes to assemble. We compared SELECT to Full, the ensemble that combines all results, and DivE, an existing ensemble [25] that combines diverse, i.e., least correlated results.

Our quantitative evaluation on real-world datasets with ground truth show that building selective ensembles is effective in boosting detection performance. SelectH appears to be a better strategy than SelectV, where it either provides the best result or achieves comparable accuracy when SelectV is the winner. Selecting results based on diversity turns out to be a poor strategy for anomaly ensembles as DivE yields even worse results than the Full ensemble. Noise analysis further corroborates the fact that DivE selects inaccurate/noisy results for the sake of diversity and declines in accuracy faster.

Acknowledgments This material is based upon work supported by the ARO Young Investigator Program with Contract No. W911NF-14-1-0029, NSF CAREER 1452425, NSF IIS-1017181, and a gift from Northrop Grumman. Any findings and conclusions expressed in this material are of the authors' and do not necessarily reflect the views of the funding parties.

References

[1] C. C. Aggarwal. Outlier ensembles: position paper. *SIGKDD Explor. Newsl.*, 14(2):49–58, 2012.

[2] L. Akoglu and C. Faloutsos. Event detection in time series of mobile communication graphs. In *27th Army Science*, 2010.

[3] L. Akoglu, H. Tong, and D. Koutra. Graph-based anomaly detection and description: A survey. *DAMI*, 28(4), 2014.

[4] G. Brown, J. L. Wyatt, R. Harris, and X. Yao. Diversity creation methods: A survey and categorisation. *Information Fusion*, 6(1):5–20, 2005.

[5] T. G. Dietterich. Ensemble methods in machine learning. In *Multiple Classifier Systems*, volume 1857. Springer, 2000.

[6] C. Dwork, R. Kumar, M. Naor, and D. Sivakumar. Rank aggregation methods for the Web. In *WWW*, 2001.

[7] N. Eagle, A. S. Pentland, and D. Lazer. Inferring friendship network structure by using mobile phone data. *PNAS*, 2009.

[8] X. Z. Fern and W. Lin. Cluster ensemble selection. In *SDM*, pages 787–797. SIAM, 2008.

[9] J. Gao, W. Hu, Z. M. Zhang, and O. Wu. Unsupervised ensemble learning for mining top-n outliers. In *PAKDD*, 2012.

[10] J. Gao and P.-N. Tan. Converting output scores from outlier detection algorithms to probability estimates. In *ICDM*, 2006.

[11] J. Ghosh and A. Acharya. Cluster ensembles: Theory and applications. In *Data Clustering: Alg. and Appl.* 2013.

[12] S. T. Hadjitodorov, L. I. Kuncheva, and L. P. Todorova. Moderate diversity for better cluster ensembles. *Information Fusion*, 7(3):264–275, 2006.

[13] L. K. Hansen and P. Salamon. Neural network ensembles. *IEEE Trans. Pattern Anal. Mach. Intell.*, 12(10), 1990.

[14] J. Kemeny. Mathematics without numbers. In *Daedalus*, pages 577–591, 1959.

[15] R. Kolde, S. Laur, P. Adler, and J. Vilo. Robust rank aggregation for gene list integration and meta-analysis. *Bioinformatics*, 28(4):573–580, 2012.

[16] H.-P. Kriegel, P. Krger, E. Schubert, and A. Zimek. Interpreting and unifying outlier scores. In *SDM*, pages 13–24, 2011.

[17] L. Kuncheva and C. Whitaker. Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. *Machine Learning*, 51:181–207, 2003.

[18] A. Lakhina, M. Crovella, and C. Diot. Diagnosing network-wide traffic anomalies. In *SIGCOMM*, pages 219–230, 2004.

[19] A. Lazarevic and V. Kumar. Feature bagging for outlier detection. In *KDD*, pages 157–166. ACM, 2005.

[20] S. Papadimitriou, J. Sun, and C. Faloutsos. Streaming pattern discovery in multiple time-series. In *VLDB*, 2005.

[21] C. Preisach and L. Schmidt-Thieme. Ensembles of relational classifiers. *Knowl. and Info. Sys.*, 14:249–272, 2007.

[22] S. Rayana and L. Akoglu. An ensemble approach for event detection in dynamic graphs. In *KDD ODD^2 Workshop*, 2014.

[23] S. Rayana and L. Akoglu. Less is more: Building selective anomaly ensembles. *CoRR*, abs/1501.01924, 2015.

[24] L. Rokach. Ensemble-based classifiers. *Artif. Intell. Rev.*, 33(1-2):1–39, 2010.

[25] E. Schubert, R. Wojdanowski, A. Zimek, and H.-P. Kriegel. On evaluation of outlier rankings and outlier scores. In *SDM*, pages 1047–1058, 2012.

[26] A. P. Topchy, A. K. Jain, and W. F. Punch. Clustering ensembles: Models of consensus and weak partitions. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(12):1866–1881, 2005.

[27] G. Valentini and F. Masulli. Ensembles of learning machines. In *WIRN*, 2002.

[28] A. Zimek, R. J. Campello, and J. Sander. Ensembles for unsupervised outlier detection: Challenges and research questions. *SIGKDD Explor. Newsl.*, 15(1):11–22, 2013.

[29] A. Zimek, M. Gaudet, R. J. G. B. Campello, and J. Sander. Subsampling for efficient and effective unsupervised outlier detection ensembles. In *KDD*, pages 428–436. ACM, 2013.