Less is More: Building Selective Anomaly Ensembles

SHEBUTI RAYANA and LEMAN AKOGLU,
Department of Computer Science, Stony Brook University

Ensemble learning for anomaly detection has been barely studied, due to difficulty in acquiring ground truth and the lack of inherent objective functions. In contrast, ensemble approaches for classification and clustering have been studied and effectively used for long. Our work taps into this gap and builds a new ensemble approach for anomaly detection, with application to event detection in temporal graphs as well as outlier detection in no-graph settings. It handles and combines multiple heterogeneous detectors to yield improved and robust performance. Importantly, trusting results from *all* the constituent detectors may deteriorate the overall performance of the ensemble, as some detectors could provide inaccurate results depending on the type of data in hand and the underlying assumptions of a detector. This suggests that combining the detectors *selectively* is key to building effective anomaly ensembles—hence "less is more".

In this paper we propose a novel ensemble approach called SELECT for anomaly detection, which automatically and systematically selects the results from constituent detectors to combine in a fully unsupervised fashion. We apply our method to event detection in temporal graphs and outlier detection in multi-dimensional point data (no-graph), where SELECT successfully utilizes five base detectors and seven consensus methods under a unified ensemble framework. We provide extensive quantitative evaluation of our approach for event detection on five real-world datasets (four with ground truth events), including Enron email communications, RealityMining SMS and phone call records, New York Times news corpus, and World Cup 2014 Twitter news feed. We also provide results for outlier detection on seven real-world multi-dimensional point datasets from UCI Machine Learning Repository. Thanks to its selection mechanism, SELECT yields superior performance compared to the individual detectors alone, the full ensemble (naively combining all results), an existing diversity-based ensemble, and an existing weighted ensemble approach.

Categories and Subject Descriptors: D.2.8 [**Database Management**]: Database applications—*Data mining*

General Terms: Design, Algorithms, Measurement

Additional Key Words and Phrases: ensemble methods, anomaly mining, anomaly ensembles, unsupervised learning, rank aggregation, event detection, dynamic graphs

ACM Reference Format:
Shebuti Rayana and Leman Akoglu. 2016. Less is More: Building Selective Anomaly Ensembles. *ACM Trans. Knowl. Discov. Data.* X, X, Article X (April 2016), 33 pages.
DOI:http://dx.doi.org/10.1145/0000000.0000000

1. INTRODUCTION

Ensemble methods utilize multiple algorithms to obtain better performance than the constituent algorithms alone and produce more robust results [Dietterich 2000].

This material is based upon work supported by the DARPA Transparent Computing Program under Contract No. FA8650-15-C-7561, Army Research Office Young Investigator Program under Contract No. W911NF-14-1-0029, National Science Foundation CAREER 1452425 and IIS 1017181, an R&D grant from Northrop Grumman Aerospace Systems, and a gift from Facebook. Any conclusions expressed in this material are of the authors and do not necessarily reflect the views, expressed or implied, of the funding parties.
Author's addresses: S. Rayana and L. Akoglu, Department of Computer Science, Stony Brook University, 100 Nicholls Rd, Stony Brook, NY 11794; email: {srayana, leman}@cs.stonybrook.edu.

DOI:http://dx.doi.org/10.1145/0000000.0000000

Thanks to these advantages, a large body of research has been devoted to ensemble learning in classification [Hansen and Salamon 1990; Preisach and Schmidt-Thieme 2007; Rokach 2010; Valentini and Masulli 2002] and clustering [Fern and Lin 2008; Ghosh and Acharya 2013; Hadjitodorov et al. 2006; Topchy et al. 2005]. On the other hand, building effective ensembles for anomaly detection has proven to be a challenging task [Aggarwal 2012; Zimek et al. 2013a]. 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 [Gao et al. 2012; Gao and Tan 2006; Kriegel et al. 2011; Lazarevic and Kumar 2005], or induce diversity among their detectors to increase the chance that they make independent errors [Schubert et al. 2012; Zimek et al. 2013b]. However, as our prior work [Rayana and Akoglu 2014] 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. Moreover, using weighted aggregation approach to combine the constituent detectors as proposed by Klementiev et al. [Klementiev et al. 2007] also get hurt by the inaccurate detectors which we show in our experiments.

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 provide theoretical evidence for our SELECT approach to achieve better accuracy compared to the base detectors and other baseline approaches.

We apply our ensemble approach to the event detection problem in temporal graphs as well as outlier detection problem in multi-dimensional point data (no-graph), where SELECT utilizes five heterogeneous event/outlier 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, the diversity-based ensemble in [Schubert et al. 2012], as well as the weighted ensemble approach in [Klementiev et al. 2007].

2. BACKGROUND AND PRELIMINARIES

2.1. Anomaly Mining

Anomalies are points in the data that do not conform to the normal behavior. As such anomaly detection refers to the problem of finding unusual points in the data that deviate from usual behavior. These non-conforming unusual points are often referred to as anomalies, outliers, exceptions, rare events etc. Most often, anomalies and outliers are two terms used interchangeably in various application domains. In this work, we propose an ensemble approach for anomaly detection with application to (i) event detection in temporal graphs, and (ii) outlier detection in multi-dimensional point data (no-graph). In the following two sections we provide description of both event and outlier detection problems.

2.1.1. 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.

In this work, we consider temporal anomalies as events. Here, temporal anomalies are those time points at which the graph structure changes significantly. 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 $G_{t'-1}$.

2.1.2. Outlier Detection Problem. A well known characterization of an outlier is given by Hawkins as, ”an observation which deviates so much from other observations as to arouse suspicion that it was generated by a different mechanism” [Hawkins 1980]. A popular formulation of outlier detection is to find unusual points in multi-dimensional data by their distance to the neighboring points. Based on this notion there exist two most famous approaches for outlier detection (i) distance based, and (ii) density based methods. Specifically, distance outlier detection problem is to find data points which are far from the rest of the data and density based methods find the points which reside in a lower density region compared to its nearest neighbors. Formally, the problem can be stated as follows.
Given a multi dimensional data D with n individual points and d dimensions;
Find points which are far from the rest of the data or reside in a lower density region.

2.2. Motivation for Ensembles

Several different methods have been proposed for the above problems, survey of which are given in [Akoglu et al. 2014; Chandola et al. 2009]. 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 [Rayana and Akoglu 2014]. This suggests that *selecting* which detectors to assemble is a critical aspect of building effective and robust ensembles—which implies that “less is more”.

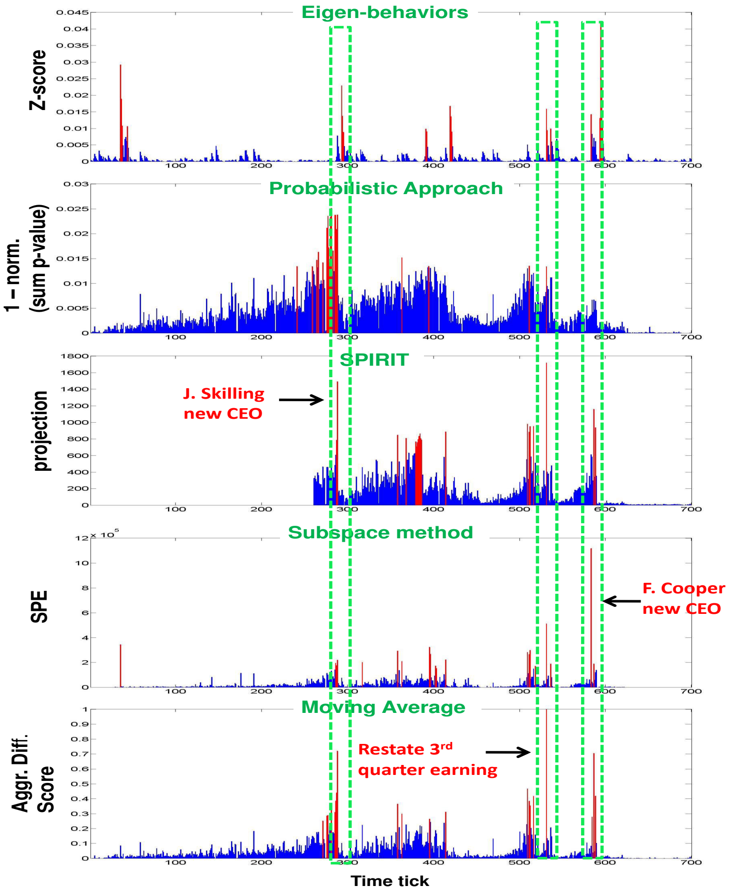

Fig. 1. Anomaly scores from five detectors (rows) for the Enron Inc. time line. Red bars depict top 20 anomalous time points.

To illustrate the motivation for (selective) ensemble building further, consider the event detection 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.

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 ensembles for anomaly mining. While we build ensembles for the event and outlier detection problems in this paper, our approach is general and can directly be employed on a collection of detection methods for other anomaly mining problems.

2.4. Important Notations

Table I lists the important notations used throughout this paper.

Table I. Symbols used in this work.

Symbol	Description
t, t'	time points
T	total time points
G_t	snapshot of the graph G at time point t
D	multi-dimensional point data
n	number of nodes or number of data points
d	dimension of point data
w	window size for first base detector
$u(t)$	eigen vector for time window t
$r(t)$	summary of past eigen vectors at time t
Z	Z-score (anomalousness score)
μ_t	moving average
σ_t	moving standard deviation
r_i	rank of a point by detector i
R	set of anomaly rank lists by different base detectors
O	target anomalies (pseudo ground truth)
fR	aggregated final rank list
$\mathbf{r}$	sorted normalized rank vector
$\hat{r}$	normalized ranks generated from uniform null distribution
$r_{(l)}$	normalized rank of a data point in list $l \in R$
$pVals$	binomial probability matrix for normalized rank vectors
S_{sort}	sorted index matrix for normalized rank vector
$p_{l,m}((\mathbf{r}))$	binomial probability of drawing at least l normalized rankings uniformly from $[0, 1]$ must be in the range $[0, r_{(l)}]$
ρ	minimum of p-values
S	set of anomaly score lists by different base detectors
P	set of probability of anomalousness lists by different base detectors
$target$	pseudo ground truth
$wP()$	weighted Pearson correlation function
E	set of selected lists by SELECT for ensemble
$class$	class labels, 1 for outliers and 0 for inliers
M	set of class labels list by different base detectors
m_{ind}	index of minimum p-value
F	list of inaccurate detectors for target anomalies
$count$	frequency of inaccurate detectors in F
C_l	cluster of detectors with low $count$
C_h	cluster of detectors with high $count$
w_i	relative weight of base detector i for ULARA

3. SELECT: SELECTIVE ENSEMBLE LEARNING FOR ANOMALY DETECTION

3.1. Overview

Our SELECT approach takes the input data, (i) for event detection 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 $1 \leq t \leq T$, and (ii) for outlier detection d dimensional point data in D, and outputs a rank list of those data points, ranked from most to least anomalous.

The main steps of SELECT are given in Algorithm 1. Step 1 employs (five) different anomaly detection algorithms as base detectors of the ensemble. Each detector has a specific and different measure to score the individual objects (time/point data) 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 anomaly 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 of results into the final rank list of objects using inverse rank aggregation (Section 3.3).

Algorithm 1 SELECT

```
Input: Data: graph sequence $\{G_1, \dots, G_t, \dots, G_T\}$
Output: Rank list of objects (time/point data) 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 5). 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, e.g. outlier detection, where the base detectors are replaced with a set of algorithms for the particular task at hand. As such we also utilize SELECT for building outlier ensemble with multi-dimensional point data.

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

In this work SELECT employs five base detectors (Algorithm 1, Line 1) in the Anomaly Ensemble. SELECT is a flexible approach, as such one can easily expand the ensemble with other base detectors. There exists various approaches for outlier detection [**?**] based on different aspects of outliers, or designed for distinct applications which require detection of domain specific outliers. In our work, we are interested about *unsupervised* outlier detection approaches that assign outlierness scores to the individual instances in the data, as such, allow ranking of instances based on outlierness.

There are a number of well known unsupervised approaches, e.g., “distance based” and “density based” methods for outlier detection. Distance based methods [Knorr and Ng 1997; Zhang et al. 2009] and its variants are mostly based on k nearest neighbor (kNN) distances between the instances, trying to find the *global* outliers far from the rest of the data. On the other hand, density based methods [Breunig et al. 2000; Papadimitriou et al. 2003] and its variants try to find the *local* outliers which are located in a lower density region compared to their k nearest neighbors.

In this work, for outlier ensemble with no-graph settings SELECT employs two distance based approaches (i) AvgKNN (average k nearest neighbor distance of individual instances is used as outlierness score), (ii) LDOF [Zhang et al. 2009], and three density based approaches (iii) LOF [Breunig et al. 2000], (iv) LOCI [Papadimitriou et al. 2003], and (v) LoOP [Kriegel et al. 2009]. For brevity we skip the detailed description of these well established outlier detection approaches.

Moreover, there exist various methods for the event detection problem in temporal graphs [Akoglu et al. 2014]. SELECT utilizes five base detectors for event detection, e.g., (1) eigen-behavior based event detection (EBED) from our prior work [Akoglu and Faloutsos 2010], (2) probabilistic time series anomaly detection (PTSAD) we developed recently [Rayana and Akoglu 2014], (3) Streaming Pattern DIscoveRy in multIple Time-Series (SPIRIT) by Papadimitriou *et al.* [Papadimitriou et al. 2005], (4) anomalous subspace based event detection (ASED) by Lakhina *et al.* [Lakhina et al. 2004], and (5) moving-average based event detection (MAED).

Event detection methods extract graph-centric features (e.g., degree) for all nodes over time and detect events in multi-variate time series. We provide brief descriptions of the methods in the following subsections.

3.2.1. Eigen Behavior based Event Detection (EBED). The multi-variate time series contain the feature values of each node over time and can be represented as a $n \times t$ data matrix, for n nodes and t time points. EBED [Akoglu and Faloutsos 2010] defines sliding time windows of length w over the series and computes the principal left singular vector of each $n \times w$ matrix W. This vector is the same as the principal eigenvector of WW^T and is always positive due to the Perron-Frobenius theorem [Perron 1907]. Each eigenvector $u(t)$ is treated as the "eigen-behavior" of the system during time window t, the entries of which are interpreted as the "activity" of each node.

To score the time points, EBED computes the similarity between eigen-behavior $u(t)$ and a summary of past eigen-behaviors $r(t)$, where $r(t)$ is the arithmetic average of $u(t')$'s for $t' < t$. The anomalousness score of time point t is then $Z = 1 - u(t) \cdot r(t) \in [0, 1]$, where high value of Z indicates a change point. For each anomalous time point $\bar{t}$, EBED performs attribution by computing the relative change $\frac{|u_i(\bar{t}) - r_i(\bar{t})|}{u_i(\bar{t})}$ of each node i at $\bar{t}$. The higher the relative change, the more anomalous the node is.

3.2.2. Probabilistic Time Series Anomaly Detection (PTSAD) . A common approach to time series anomaly detection is to probabilistically model a given series and detect anomalous time points based on their likelihood under the model. PTSAD models each series with four different parametric models and performs model selection to identify the best fit for each series. Our first model is the *Poisson*, which is used often for fitting count data. However, Poisson is not sufficient for sparse series with many zeros. Since real-world data is frequently characterized by over-dispersion and excess number of zeros, we employ a second model called *Zero-Inflated Poisson* (ZIP) [Lambert 1992] to account for data sparsity.

We further look for simpler models which fit data with many zeros and employ the *Hurdle models* [Porter and White 2012]. Rather than using a single but complex distribution, Hurdle models assume that the data is generated by two simple, separate processes; (i) the hurdle and (ii) the count processes. The hurdle process determines whether there exists activity at a given time point and in case of activity the count process determines the actual (positive) counts. For the hurdle process, we employ two different models. First is the independent *Bernoulli* and the second is the first order *Markov* model which better captures the dependencies, where an activity influences the probability of subsequent activities. For the count process, we use the *Zero-Truncated Poisson* (ZTP) [Cameron and Trivedi 1998].

Overall we model each time series with four different models: Poisson, ZIP, Bernoulli+ZTP and Markov+ZTP. We then employ Vuong's likelihood ratio test [Vuong 1989] to select the best model for individual series. Note that the best-fit model for each series may be different.

To score the time points, we perform a single-sided test to compute a p-value for each value x in a given series; i.e., $P(X \geq x) = 1 - cdf_H(x) + pdf_H(x)$, where H is the best-fit model for the series. The lower the p-value, the more anomalous the time point is. We then aggregate all the p-values from all the series per time point by taking the normalized sum of the p-values and inverting them to obtain scores $\in [0, 1]$ (s.t. higher is more anomalous). For each anomalous time point $\bar{t}$, attribution is done by sorting the nodes (i.e., the series) based on their p-values at $\bar{t}$.

3.2.3. Streaming Pattern DIscoveRy in multIple Time-Series (SPIRIT) . SPIRIT [Papadimitriou et al. 2005] can incrementally capture correlations, discover trends, and dynamically detect change points in multi-variate time series. The main idea is to represent the underlying trends of a large number of numerical streams with a few hidden variables, where the hidden variables are the *projections* of the observed streams onto the principal direction vectors (eigenvectors). These discovered trends are exploited for detecting change points in the series.

The algorithm starts with a specific number of hidden variables that capture the main trends of the data. Whenever the main trends change, new hidden variables are introduced or several of existing ones are discarded to capture the change. SPIRIT can further quantify the change in the individual time series for attribution through their *participation weights*, which are the entries in the principal direction vectors. For further details on the algorithm, we refer the reader to the original paper by Papadimitriou *et al.* [Papadimitriou et al. 2005].

3.2.4. Anomalous Subspace based Event Detection (ASED). ASED [Lakhina et al. 2004] is based on the separation of high-dimensional space occupied by the time series into two disjoint subspaces, the normal and the anomalous subspaces. Principal Component Analysis is used to separate the high-dimensional space, where the major principal components capture the most variance of the data and hence, construct the normal subspace and the minor principal components capture the anomalous subspace. The projection of the time series data onto these two subspaces reflect the normal and anomalous behavior. To score the time points, ASED uses the *squared prediction error* (SPE) of the residuals in the anomalous subspace. The residual values associated with individual series at the anomalous time points are used to measure the anomalousness of nodes for attribution. For the specifics of the algorithm, we refer to the original paper by Lakhina *et al.* [Lakhina et al. 2004].

3.2.5. Moving Average based Event Detection (MAED). MAED is a simple approach that calculates the moving average μ_t and the moving standard deviation σ_t of each time series corresponding to each node by extending the time window one point at a time. If the value at a specific time point is more than three moving standard deviations away from the mean, then the point is considered as anomalous and assigned a nonzero score. The anomalousness score is the difference between the original value and $(\mu_t + 3\sigma_t)$ at t. To score the time points collectively, MAED aggregates their scores across all the series. For each anomalous time point $\bar{t}$, attribution is done by sorting the nodes (i.e., the series) based on the individual scores they assign to $\bar{t}$.

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 (time points for event detection) 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 [Dwork et al. 2001]. SELECT employs three rank based consensus methods. *Kemeny-Young* [Kemeny 1959] 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) [Kolde et al. 2012] utilizes order statistics to compute the probability that a given ordering of ranks for a point

Algorithm 2 RobustRankAggregation

```
Input: $R$ := set of anomaly rank lists, $O$ := target anomalies
Output: $fR$ := aggregated final rank list
    $pVals$ := probability matrix for normalized rank vectors
    $S_{sort}$ := sorted index matrix for normalized rank vector
$nR := \emptyset$, $m = length(R)$/*total item in rank list*/
/* calculate normalized rank vector */
for each column $l \in R$ do
    /* Rank() finds the rank of items in the rank list $l$ */
    $nR := nR \cup Rank(l)/m$
end for
$sR := \emptyset$, $S_{sort} := \emptyset$
for each row $l \in nR$ do
    $[sl, ind] = sort(l)$/* ascending order */
    $sR := sR \cup sl$
    $S_{sort} = S_{sort} \cup ind$
end for
$pVals := \emptyset$
for each row $\mathbf{r} \in sR$ do
    $\beta = zeros(1, m)$
    for $l := 1 \ldots m$ do
        $p_{l,m}(\mathbf{r}) := \sum_{t=l}^{m} \binom{m}{t} r_{(l)}^{t} (1 - r_{(l)})^{m-t}$
        $\beta(1, l) := \beta(1, l) + p_{l,m}(\mathbf{r})$
    end for
    $\rho(\mathbf{r}) = min(beta)$
    $pVals := pVals \cup \beta$
end for
$fR = sort(\rho)$/*ascending order*/
if $O \neq \emptyset$ then
    $S_{sort} := S_{sort}(O, :)$
    $pVals := pVals(O, :)$
end if
return $fR$, $S_{sort}$, $pVals$
```

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 steps of *Robust Rank Aggregation* are given in Algorithm 2.

Given a set of anomaly rank lists R, we first calculate the normalized rank of each data point by dividing its rank by the length of the rank list. For each data point, we get the normalized rank vector ($\mathbf{r} \in sR$) $\mathbf{r} = [r_{(1)}, \ldots, r_{(m)}]$, such that $r_{(1)} \leq \ldots \leq r_{(m)}$, where $r_{(l)}$ denotes the normalized rank of a data point in list $l \in R$. We also store and return the sorted indices in S_{sort} (for its further use in SELECT described in Section 3.4.2). We then compute the order statistics based on this sorted normalized rank vectors to calculate the final aggregated rank list. 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 from a uniform null distribution. This probability ($p_{l,m}(\mathbf{r})$) can be expressed as a binomial probability (in step 17) since at least l normalized rankings drawn uniformly from $[0, 1]$ must be in the range $[0, r_{(l)}]$. The accurate lists rank the anomalies at the top, and hence yield low normalized ranks $r_{(l)}$, so the probability is expected to drop with the ordering ($l \in 1, \ldots, m$). We also store and return these probabilities in $pVals$ (for its further use in SELECT described in Section 3.4.2). As the number of accurate detectors is not known, we define the final score $\rho(\mathbf{r})$ in step 20 for the normalized rank vector $\mathbf{r}$ as the minimum of p-values. Finally, we order the data points in fR according to this ρ values, where lower values means more anomalous.

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* [Gao and Tan 2006] converts the anomaly scores into probabilities by modeling them as sampled from a mixture of exponential (for inliers) and Gaussian (for outliers) distributions. We use an expectation maximization (EM) algorithm to minimize the negative log likelihood function of the mixture model to estimate the parameters. We calculate the final posterior probability with Bayes rule which represents the probability of anomalousness of the data points. *Mixture Modeling* also provides a binary decision (*class*) for the data points, where point with probability greater than 0.5 gets class 1 (for outliers) and 0 (for inliers) otherwise. *Unification* [Kriegel et al. 2011] also converts the scores into probability estimates through regularization, normalization, and Gaussian scaling steps. The probabilities are then comparable across detectors, which we aggregate by max or avg. This yields four score-based methods.

3.4. Proposed 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 our proposed approach for building anomaly 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

Algorithm 3 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 $E$
```

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.

3.4.1. 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 3.

Given a set of anomaly score lists S, we first unify the scores by converting them to probability estimates using *Unification* [Kriegel et al. 2011]. 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).

3.4.2. Strategy II: Horizontal Selection. We are interested in finding the data 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 (defined in Section 3.3) of the anomalies should come from a distribution skewed toward zero as the accurate lists are considered to have the anomalies at high positions. 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 4.

Algorithm 4 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 $count$s into two clusters, $C_l$ and $C_h$
17: $E := S \setminus \{s \in C_h\}$  /* discard high-$count$ lists */
18: return $E$
```

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* [Gao and Tan 2006] to convert each score list in S into probability estimates by modeling them as sampled from a mixture of exponential (for inliers) and Gaussian (for outliers) distributions. We then generate binary lists from the probability estimates in which outliers are denoted by 1 (for probabilities > 0.5), and inliers by 0 (otherwise). We then employ majority voting across these binary 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* [Kolde et al. 2012] (in Section 3.3), we then compute order statistics based on these sorted normalized rank lists to identify the lists (inaccurate ones) 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 since at least l normalized rankings drawn uniformly from $[0, 1]$ must be in the range $[0, r_{(l)}]$.

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

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 $l \in \{1 \ldots m\}$. As with increasing ordering the probability of drawing more normalized ranks uniformly from $[0, 1]$ to be in a small range $[0, r_{(l)}]$ gets small. 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.

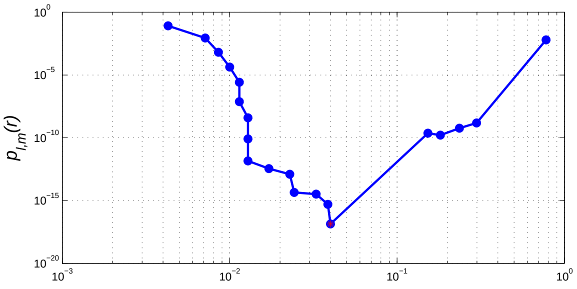

Fig. 2. Normalized rank $r_{(l)}$ vs. probability p that $\hat{r}_{(l)} \leq r_{(l)}$, where $\hat{r}$ are drawn uniformly at random from $[0, 1]$.

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.

3.5. Existing/Alternative Ensemble Learning Approaches

In this section, we discuss three alternative existing approaches for building anomaly ensembles, which differ in whether and how they select their ensemble components. We compare to these methods in the experiments (Section 5).

3.5.1. Full ensemble. The full ensemble [Rayana and Akoglu 2014] 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.

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

3.5.2. 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) [Dietterich 2000; Valentini and Masulli 2002]. 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 [Brown et al. 2005; Kuncheva and Whitaker 2003].

Following on these insights, Schubert *et al.* proposed a diversity-based ensemble [Schubert et al. 2012], which is similar to our vertical selection in Alg. 3. 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 and outlier detection ensembles.

3.5.3. Unsupervised Learning Algorithm for Rank Aggregation (ULARA). In selective ensemble approaches, base detectors and consensus approaches are selected in an unsupervised way to generate the final result. In doing so the algorithms which are not selected are discarded and do not contribute to the final result. An alternative way to using binary selection criteria is estimating weights for detectors/consensus results and applying a weighted rank aggregation technique to combine the results. [Klementiev et al. 2007] proposed an unsupervised learning algorithm (called ULARA) for this kind of rank aggregation, which adaptively learns a parameterized linear combination of ranklists to optimize the relative influence of individual detectors on the final ranking by learning relative weights w_i for the individual ranklists (where, $\sum_{i=1}^{n} w_i = 1$). Their approach is guided by the principle that the relative contribution of an individual ranklist to the final ranking should be determined by its tendency to agree with other ranklists in the pool. Those ranklists that agree with the majority are given large relative weights and those that disagree are given small relative weights. Agreement is measured by the total variance from the average ranking of individual data points. As a result, the goal is to assign weights such that the total weighted variance is minimized. ULARA has two different ways to estimate the detector weights, one based on additive and another based on exponential weight updates. In evaluation, we experiment with both of them and report the better performance for each dataset.

4. THEORETICAL FOUNDATIONS

In this section we present the theoretical underpinnings of our proposed anomaly ensemble. Although, classification and anomaly detection problems are significantly different, the theoretical foundation of both the problems can be explained with bias-variance tradeoff. We explain the theoretical analysis for our anomaly ensemble in light of the theoretical foundations provided by Aggarwal *et al.* [Aggarwal and Sathe 2015] for outlier ensemble in terms of well known ideas from classification. In Section 4.1 we describe the bias-variance trade-off for anomaly detection and in Section 4.2

[2] There are other differences between our vertical selection (Algorithm 3) and the diversity-based ensemble in [Schubert et al. 2012], such as the construction of the pseudo ground truth and the choice of weights in correlation computation.

we present evidence for error reduction by reducing bias-variance for our proposed selective anomaly ensemble approach SELECT.

4.1. Bias-Variance Tradeoff in Anomaly Detection

The bias-variance tradeoff is often explained in the context of supervised learning, e.g., in classification, as quantification of bias-variance requires labelled data. Although, anomaly detection problems lack the existence of ground truth (hence solved using unsupervised approaches), this bias-variance tradeoff can be quantified by treating the dependent variable (actual labels) as unobserved.

Unlike classification, most anomaly detection algorithms output “anomalousness” scores for the data points. We can consider these anomaly detection algorithms as two class classification problems having a majority class (normal points) and a rare class (anomalous points) by converting the anomalousness scores to class labels. The points which achieve scores above a threshold are considered as anomalies and get label 1 (label 0 for normal points below threshold). Deciding this threshold is a difficult task for heterogeneous detectors as they provide scores with different scaling and in different ranges. As such there exist unification approaches [Gao and Tan 2006; Kriegel et al. 2009] which convert these anomalousness scores to probability estimates to make them comparable with out changing the ranking of the data points.

Now that unsupervised anomaly detection problem looks similar like a classification problem with only unobserved actual labels, we can explain the bias-variance tradeoff for anomaly detection using ideas from classification. The expected error of anomaly detection can be split into two main components reducible error and irreducible error (i.e., noise). This reducible error can be minimized to maximize the accuracy of the detector. Furthermore, the reducible error can be decomposed into (i) error due to squared bias, and (ii) error due to variance. However, there is a tradeoff while minimizing both these sources of errors.

Bias of a detector is the amount by which the expected output of the detector differs from the true unobserved value, over the training data. On the other hand, variance of a detector is the amount by which the output of a detector over one training set differs from the expected output of the detector over all the training sets. The tradeoff between bias and variance can be viewed as, (i) a detector which has low bias is very flexible in fitting data well and it will fit each training set differently providing high variance, and (ii) inflexible detectors will have low variance and might provide high bias. Our goal is to improve the accuracy as much as possible by reducing both bias and variance using selective anomaly ensemble approach SELECT.

4.2. Bias-Variance Reduction in Anomaly Ensemble

Most classification ensemble generalize directly to anomaly ensemble for variance reduction, but controlled bias reduction is rather difficult due to lack of ground truth. It is evident from classification ensemble literature that combining results from multiple heterogeneous base algorithms will decrease the overall variance of the ensemble [Aggarwal and Sathe 2015] which is also true for anomaly ensemble. On the other hand, this combination does not provide enough ground for reducing bias in anomaly ensemble. Moreover, our SELECT approach is designed based on the assumption that, there exist inaccurate detectors which are able to hurt the overall ensemble if combined with the accurate ones.

In this work, we present two selective approaches SelectV and SelectH which discard these inaccurate detectors. For both the algorithms we utilize pseudo-ground truth which can be viewed as a low-bias output because it averages the outputs for SelectV and takes majority voting for SelectH across the diverse detectors, each of which might have biases in different directions. By eliminating the detectors which do not agree with the pseudo-ground truth for SelectV and provide inaccurate ranking (compared to others) of the target anomalies (pseudo-ground truth) in SelectH, we are effectively eliminating the detectors which have high bias. Furthermore, we are using SELECT in two phases to reduce the bias. Therefore, by carefully selecting detectors and combining their outputs in two phases we are reducing both bias and variance, and thus improving accuracy.

The reason why the state-of-the-art approaches (Full, DivE, ULARA) might fail to achieve better accuracy than SELECT can be describes with this bias-variance reduction. Full combines all the base detectors results including the ones with high bias and thus hurt the final ensemble. Although ULARA calculates relative weights based on the agreement between the detectors, it fails to totally discard the ones with high bias. For the sake of diversity DivE selects the more diverse detectors and thus end up selecting the ones with high bias, reducing the overall accuracy.

In selecting the detectors, SelectV utilizes the correlation between the ranklists provided by the base detectors. Therefore, SelectV considers all the data points to decide which detectors to select and thus affected by the majority inliers class. On the other hand, SelectH considers only the target anomalies to decide which detectors to select, as in this approach we emphasize on the anomalies being misclassified by the detectors. As a result, SelectV is sometimes prone to discarding accurate detectors and selecting inaccurate ones. Section 5 provides results justifying the above explanation.

We consider our SELECT approach to be a heuristic one as it is not guaranteed to provide optimal solution for different datasets. As such it can behave unpredictably for pathological data sets (see Section 5). Bias reduction in unsupervised learning, e.g., anomaly detection, is a hard problem and using heuristic method is quite reasonable to improve accuracy by achieving immediate goals.

5. 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. Furthermore, we evaluate SELECT on the outlier detection problem using seven real-world datasets from UCI machine learning repository.[3]

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), the diversity-based ensemble (DivE) by [Schubert et al. 2012], and weighted ensemble approach (ULARA) by [Klementiev et al. 2007]. 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.

5.1. Dataset Description

5.1.1. Temporal Graph Datasets. In the following we describe the five real-world temporal graph datasets we used in this work. Our datasets are collected from various domains. Four datasets contain ground truth events, and the last dataset is used

[3] http://archive.ics.uci.edu/ml/datasets.html

for illustrating case studies. All our datasets can be found at `http://shebuti.com/SelectiveAnomalyEnsemble/`, where we also share the source code for SELECT.

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 [Eagle et al. 2009]. 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.[4] 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 (specifically, at least one of 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 II. Summary of multi-dimensional point data sets

Data set	Instances	Attributes	% of outliers
WBC	378	30	5.6
Glass	214	9	4.2
Lymphography	148	18	4.1
Cardio	1831	21	9.6
Musk	3062	166	3.2
Thyroid	3772	6	2.5
Letter	1600	32	6.25

5.1.2. Multi-dimensional point Datasets. In Table II we provide the summary of seven real-world datasets that we utilize in outlier ensemble from UCI Machine Learning Repository. For these datasets further preprocessing was required to adapt them in

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

outlier detection problem having a rare class (outliers) and a majority class (inliers). WBC, Glass, Lymphography, Cardio, Musk and Thyroid datasets are the same as used in [Aggarwal and Sathe 2015] and Letter dataset is same as used in [Micenkovä et al. 2014].

Table III. Significance of accuracy results compared to random ensembles with same number of selected components as SELECTfor event detection. The accuracy of the alterntive approaches (Full, DivE, and ULARA) are also given in parentheses. These results show that (i) SELECT is superior to existing methods, and (ii) it selects significantly more important (i.e., accurate) detectors to combine.

	Accuracy	significance
EnronInc. (10 comp.) (Full: 0.7082, DivE: 0.6276, ULARA: 0.3652)		
(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, ULARA: 0.2961)		
(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, ULARA: 0.8125)		
(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, ULARA: 0.8250)		
(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, ULARA: 0.8437)		
(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, ULARA: 0.7937)		
(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, ULARA: 0.7312)		
(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, ULARA: 0.5733)		
(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**	$= \mu + 1.2435\sigma$

5.2. 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.

In Table III we report the summary of results for different datasets containing the average precision values for SELECT and other existing ensemble approaches (Full, DivE, and ULARA) to which we compare SELECT. 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 III we also 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 note that SELECT ensembles provide superior results to RandE, Full, DivE, and ULARA. Moreover, SelectH appears to be a better strategy than SelectV, where it either provides the best result (6/8 in Table III) 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 (6/8 in Table III). Putting relative weight depending on the agreement between base detector results does not even help according to our evaluation, as such ULARA ensemble yields poor results even worse than DivE (6/8 in Table III).

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

		Full	DivE	ULARA(w_i)	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	0.3652	0.7125	**0.7920**

Table IV shows the accuracies for all five ensemble methods on EnronInc., along with the accuracies of the base detectors and consensus methods. In Table IV the black bars for ULARA are the representatives of weights where the length of the bars are proportional to the relative weights assigned to corresponding detectors, we denote them as *weight bars* for ULARA. Also ULARA is a single phase ensemble approach, as such there is no second phase results. We note 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. SelectV ensemble provides sparser selection than SelectH ensemble, but SelectH provides better accuracy than SelectV, which indicates that SelectV possibly missing some valuable detectors. We also note that ULARA andDivE yield lower performance than all, including Full. The weight bars indicate that ULARA is putting high weights to relatively inaccurate detectors.

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.

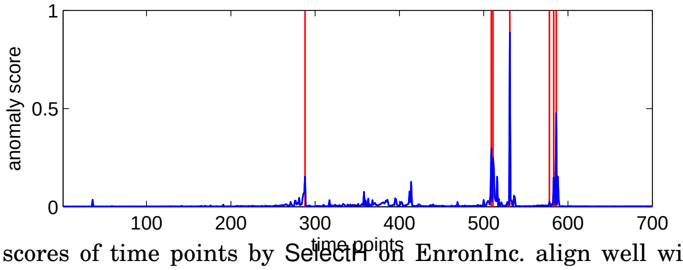

Fig. 3. Anomaly scores of time points by SelectH on EnronInc. align well with ground truth (vertical red lines).

Table V. Accuracy of ensembles for EnronInc. (directed) (20 components) (features: weighted in-/out-degree and unweighted in-/out-degree). $*$ depicts selected detector/consensus results.

		Full	DivE	ULARA(w_i)	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		▪	$*$	$*$
	EBED (uin)	0.0892	$*$	▬▬▬		
	PTSAD (uin)	0.1607	$*$	▪		$*$
	SPIRIT (uin)	0.3996	$*$	▪		$*$
	ASED (uin)	0.1395		▪	$*$	$*$
	MAED (uin)	0.4439		▪	$*$	$*$
	EBED (uout)	0.0225	$*$	▬▬		
	PTSAD (uout)	0.2546		▪		$*$
	SPIRIT (uout)	0.1012	$*$	▬		$*$
	ASED (uout)	0.0870	$*$	▪		
	MAED (uout)	0.4181		▪		$*$
Consensus	Inverse Rank	$*$ 0.7121	$*$ 0.5660	–	0.6577	$*$ 0.7496
	Kemeny-Young	$*$ 0.3033	$*$ 0.2495	–	0.5361	$*$ 0.5066
	RRA	$*$ 0.5948	$*$ 0.5348	–	0.4948	$*$ 0.5774
	Uni (avg)	$*$ 0.4838	$*$ 0.4325	–	$*$ 0.6047	$*$ 0.5336
	Uni (max)	$*$ 0.3020	$*$ 0.2242	–	0.6633	$*$ 0.4280
	MM (avg)	$*$ 0.5673	$*$ 0.4662	–	0.6761	$*$ 0.7217
	MM (max)	$*$ 0.0216	$*$ 0.0216	–	$*$ 0.5355	0.0222
Final Ensemble		0.5420	0.4697	0.2961	0.7018	**0.7798**

Table IV 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. Table V gives all the accuracy results, selections made and weight bars for ULARA, a summary of which is provided in Table III. We notice that the unweighted graph features are less informative and yield lower accuracies across detectors on average. This affects the performance of Full, DivE, and ULARA, where the accuracies drop significantly, specially for ULARA. On the other hand, SELECT ensembles are able to achieve comparable accuracies with increased significance under the additional noisy input.

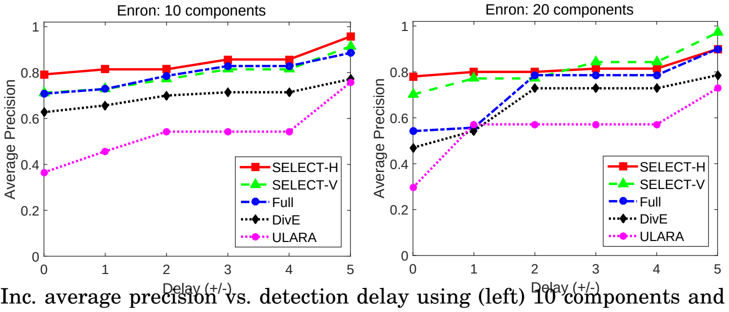

Fig. 4. EnronInc. average precision vs. detection delay using (left) 10 components and (right) 20 components.

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.

Table VI. Accuracy of ensembles for RealityMining Voice Call (directed) (10 components) (features: weighted in-/out-degree). $*$: selected detector/consensus results.

		Full	DivE	ULARA(w_i)	SelectV	SelectH
Base Algorithms	EBED (win)	0.3508	$*$	▬▬▬▬▬▬		
	PTSAD (win)	0.6284		▬		$*$
	SPIRIT (win)	0.8309	$*$	▬		$*$
	ASED (win)	0.9437		▬	$*$	$*$
	MAED (win)	0.8809	$*$	▬▬▬		$*$
	EBED (wout)	0.4122	$*$	▬▬▬▬▬		
	PTSAD (wout)	0.6273		▬		$*$
	SPIRIT (wout)	0.7346		▬	$*$	$*$
	ASED (wout)	0.9500		▬▬		$*$
	MAED (wout)	0.8758		▬▬▬		$*$
Consensus	Inverse Rank	$*$ 0.7544	0.6169	–	0.8880	$*$ 0.8222
	Kemeny-Young	$*$ 0.8221	$*$ 0.7708	–	0.8619	$*$ 0.9309
	RRA	$*$ 0.8154	0.5936	–	0.8901	$*$ 0.9416
	Uni (avg)	$*$ 0.7798	$*$ 0.6413	–	$*$ 0.8370	$*$ 0.9098
	Uni (max)	$*$ 0.6704	0.5757	–	0.7786	$*$ 0.7833
	MM (avg)	$*$ 0.9190	$*$ 0.9162	–	0.8835	$*$ 0.9183
	MM (max)	$*$ 0.4380	$*$ 0.8934	–	0.7569	0.4380
Final Ensemble		0.7302	0.8724	0.8125	0.8370	**0.9045**

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, weight bars for ULARA as well as selections made are given in Table VI and Table VII for Voice Call, Table VIII for Bluetooth, Table IX and Table X for SMS. We provide the summary of results in Table III. We note that SELECT ensembles provide superior results to Full, DivE, and ULARA.

Table VII. Accuracy of ensembles for RealityMining Voice Call (directed) (20 components) (features: weighted in-/out-degree and unweighted in-/out-degree) ∗: selected detector/consensus results.

		Full	DivE	ULARA(w_i)	SelectV	SelectH
Base Algorithms	EBED (win)	0.3508	∗	▬▬▬		
	PTSAD (win)	0.6284		▪		∗
	SPIRIT (win)	0.8309		▪	∗	∗
	ASED (win)	0.9437		▪	∗	∗
	MAED (win)	0.8809	∗	▬		∗
	EBED (wout)	0.4122	∗	▬▬▬		
	PTSAD (wout)	0.6273		▪		∗
	SPIRIT (wout)	0.7346		▪		∗
	ASED (wout)	0.9500		▬		∗
	MAED (wout)	0.8758		▬		∗
	EBED (uin)	0.4173		▬▬▬		
	PTSAD (uin)	0.8636	∗	▪		∗
	SPIRIT (uin)	0.8313		▪		∗
	ASED (uin)	0.9191		▪		∗
	MAED (uin)	0.8706	∗	▬		∗
	EBED (uout)	0.4800		▬▬		∗
	PTSAD (uout)	0.8665		▪		∗
	SPIRIT (uout)	0.7480		▪		∗
	ASED (uout)	0.9229	∗	▬		∗
	MAED (uout)	0.9115		▬		∗
Consensus	Inverse Rank	∗ 0.8035	0.7952	–	0.9240	∗ 0.8681
	Kemeny-Young	∗ 0.9064	0.9018	–	0.9076	∗ 0.9158
	RRA	∗ 0.8866	∗ 0.7771	–	0.9013	∗ 0.9311
	Uni (avg)	∗ 0.8598	0.9192	–	∗ 0.8448	∗ 0.9102
	Uni (max)	∗ 0.6844	∗ 0.6863	–	0.8517	∗ 0.7611
	MM (avg)	∗ 0.9321	∗ 0.9083	–	∗ 0.8312	∗ 0.9134
	MM (max)	∗ 0.4380	∗ 0.8858	–	0.8015	0.4380
Final Ensemble		0.8011	0.8335	0.8250	0.8847	**0.8949**

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.

Finally, we study event detection using Twitter. Table XI contains accuracy details for detecting world news on TwitterSecurity, a summary of which is included in Table III. 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

Table VIII. Accuracy of ensembles for RealityMining Bluetooth (undirected) (10 components) (feature: weighted and unweighted degree). $*$: selected detector/consensus results.

		Full	DivE	ULARA(w_i)	SelectV	SelectH
Base Algorithms	EBED (wdeg)	0.4363	$*$	▬▬▬▬		
	PTSAD (wdeg)	0.5820	$*$	▬		$*$
	SPIRIT (wdeg)	0.9499	$*$	▬		$*$
	ASED (wdeg)	0.8601		▬▬	$*$	$*$
	MAED (wdeg)	0.8359	$*$	▬		$*$
	EBED (udeg)	0.4966	$*$	▬▬▬▬		
	PTSAD (udeg)	0.8694		▬	$*$	$*$
	SPIRIT (udeg)	0.9162		▬	$*$	$*$
	ASED (udeg)	0.7662		▬▬	$*$	$*$
	MAED (udeg)	0.8788	$*$	▬		$*$
Consensus	Inverse Rank	$*$ 0.8646	$*$ 0.8255	–	0.8790	$*$ 0.8538
	Kemeny-Young	$*$ 0.9534	0.9169	–	0.9698	$*$ 0.9361
	RRA	$*$ 0.9413	0.8318	–	0.9693	$*$ 0.9684
	Uni (avg)	$*$ 0.9071	0.8654	–	$*$ 0.9193	$*$ 0.9225
	Uni (max)	$*$ 0.6973	$*$ 0.6122	–	0.8270	$*$ 0.7126
	MM (avg)	$*$ 0.9407	$*$ 0.9340	–	0.8596	$*$ 0.8892
	MM (max)	$*$ 0.6461	$*$ 0.6374	–	0.8830	0.6461
Final Ensemble		0.8398	0.7735	0.8437	**0.9193**	0.8886

Table IX. Accuracy of ensembles for RealityMining SMS (directed) (10 components) (features: weighted in-/out-degree). $*$: selected detector/consensus results.

		Full	DivE	ULARA(w_i)	SelectV	SelectH
Base Algorithms	EBED (win)	0.6117	$*$	▬▬▬		
	PTSAD (win)	0.7003		▬		$*$
	SPIRIT (win)	0.9256		▬	$*$	$*$
	ASED (win)	0.6338	$*$	▬▬		$*$
	MAED (win)	0.9002	$*$	▬▬		$*$
	EBED (wout)	0.5595	$*$	▬▬▬▬		
	PTSAD (wout)	0.7023		▬	$*$	$*$
	SPIRIT (wout)	0.8656	$*$	▬		$*$
	ASED (wout)	0.9102		▬	$*$	$*$
	MAED (wout)	0.9259		▬▬	$*$	$*$
Consensus	Inverse Rank	$*$ 0.8309	0.8174	–	0.8933	$*$ 0.8044
	Kemeny-Young	$*$ 0.9491	$*$ 0.8779	–	0.9511	$*$ 0.9386
	RRA	$*$ 0.8761	$*$ 0.8424	–	0.9578	$*$ 0.9516
	Uni (avg)	$*$ 0.8531	0.8247	–	$*$ 0.9283	$*$ 0.8684
	Uni (max)	$*$ 0.8205	$*$ 0.7632	–	0.8829	$*$ 0.8678
	MM (avg)	$*$ 0.9276	$*$ 0.9487	–	0.9492	$*$ 0.9084
	MM (max)	$*$ 0.8907	$*$ 0.8577	–	0.9410	0.9011
Final Ensemble		0.9092	0.8598	0.7937	**0.9283**	0.9217

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 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. Here, ULARA and SelectV perform quite closely.

Table X. Accuracy of ensembles for RealityMining SMS (directed) (20 components) (features: weighted in-/out-degree and unweighted in-/out-degree). $*$: selected detector/consensus results.

		Full	DivE	ULARA(w_i)	SelectV	SelectH
Base Algorithms	EBED (win)	0.6117	$*$	▬		$*$
	PTSAD (win)	0.7003		▪		$*$
	SPIRIT (win)	0.9256		▪		$*$
	ASED (win)	0.6338	$*$	▬		$*$
	MAED (win)	0.9002		▬		$*$
	EBED (wout)	0.5595		▬▬		$*$
	PTSAD (wout)	0.7023		▪		
	SPIRIT (wout)	0.8656		▪		$*$
	ASED (wout)	0.9102		▪		$*$
	MAED (wout)	0.9259		▬	$*$	$*$
	EBED (uin)	0.4407	$*$	▬▬		
	PTSAD (uin)	0.7809	$*$	▪		$*$
	SPIRIT (uin)	0.7841		▪		$*$
	ASED (uin)	0.6248	$*$	▬		$*$
	MAED (uin)	0.8297	$*$	▬		$*$
	EBED (uout)	0.3246		▬▬		
	PTSAD (uout)	0.9157		▪	$*$	$*$
	SPIRIT (uout)	0.8744		▪		$*$
	ASED (uout)	0.9150		▪		$*$
	MAED (uout)	0.8005		▬		$*$
Consensus	Inverse Rank	$*$ 0.9135	0.6751	–	0.9634	$*$ 0.9230
	Kemeny-Young	$*$ 0.9286	$*$ 0.7567	–	0.9094	$*$ 0.9325
	RRA	$*$ 0.9568	0.6465	–	0.9418	$*$ 0.9583
	Uni (avg)	$*$ 0.8791	0.6499	–	$*$ 0.9294	$*$ 0.9156
	Uni (max)	$*$ 0.7173	$*$ 0.6696	–	0.9342	0.8650
	MM (avg)	$*$ 0.9107	$*$ 0.8942	–	0.8519	$*$ 0.9138
	MM (max)	$*$ 0.8895	$*$ 0.8480	–	0.9307	0.8895
Final Ensemble		0.9542	0.8749	0.7312	0.9294	**0.9621**

Table XI. Accuracy of ensembles for TwitterSecurity (undirected) (10 components) (features: weighted and unweighted degree). $*$: selected detector/consensus results.

		Full	DivE	ULARA(w_i)	SelectV	SelectH
Base Algorithms	EBED (wdeg)	0.4000	$*$	▬▬		$*$
	PTSAD (wdeg)	0.5400	$*$	▬▬		$*$
	SPIRIT (wdeg)	0.4467	$*$	▬	$*$	$*$
	ASED (wdeg)	0.6200	$*$	▬	$*$	$*$
	MAED (wdeg)	0.4933		▬		$*$
	EBED (udeg)	0.4133	$*$	▬▬	$*$	$*$
	PTSAD (udeg)	0.5467		▬▬	$*$	$*$
	SPIRIT (udeg)	0.3867	$*$	▬		$*$
	ASED (udeg)	0.5400	$*$	▬		$*$
	MAED (udeg)	0.4533	$*$	▬▬		
Consensus	Inverse Rank	$*$ 0.4467	0.4267	–	0.5133	$*$ 0.4667
	Kemeny-Young	$*$ 0.5667	0.5333	–	0.5333	0.5800
	RRA	$*$ 0.5867	$*$ 0.5333	–	0.5467	$*$ 0.5933
	Uni (avg)	$*$ 0.5600	0.5000	–	$*$ 0.5467	$*$ 0.6000
	Uni (max)	$*$ 0.4533	$*$ 0.4400	–	0.5800	0.4533
	MM (avg)	$*$ 0.5333	$*$ 0.5667	–	0.5267	0.5600
	MM (max)	$*$ 0.3667	$*$ 0.3667	–	0.5533	0.5733
Final Ensemble		0.5200	0.4800	0.5733	0.5467	**0.5867**

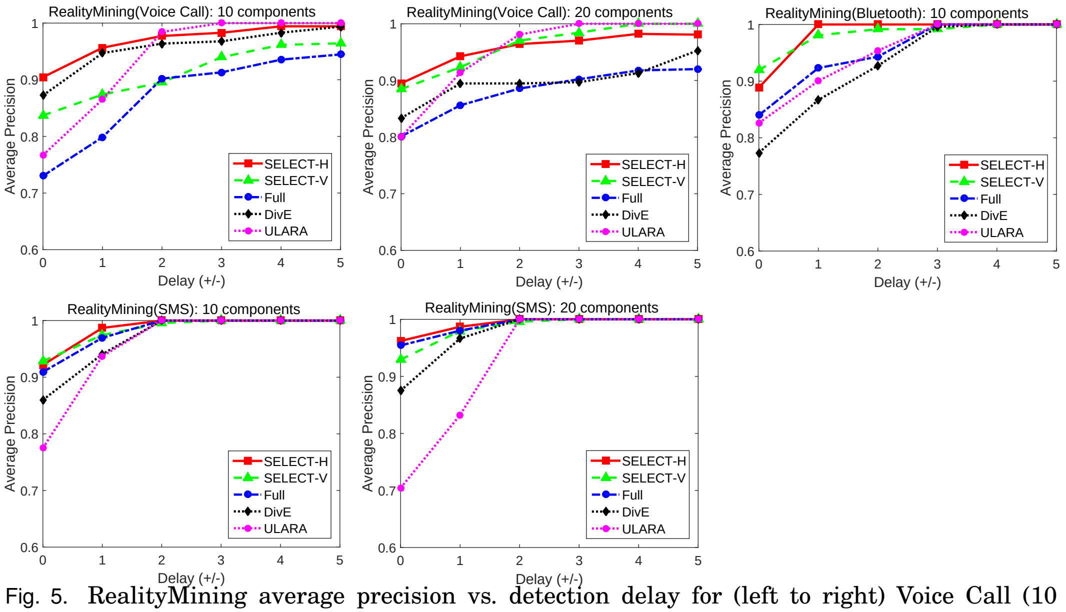

Fig. 5. RealityMining average precision vs. detection delay for (left to right) Voice Call (10 comp.), Voice Call (20 comp.), Bluetooth (10 comp.), SMS (10 comp.), and SMS (20 comp.).

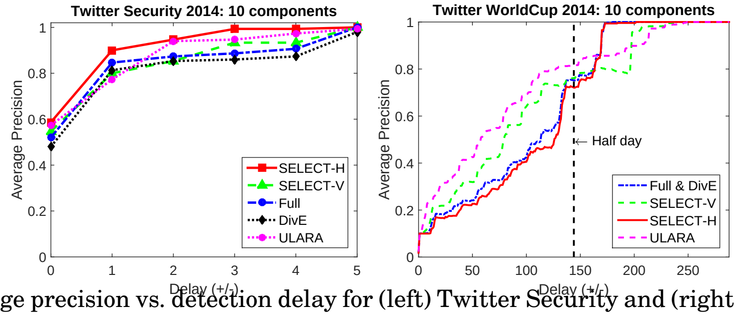

Fig. 6. Average precision vs. detection delay for (left) Twitter Security and (right) Twitter WorldCup 2014.

5.3. 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 Figure 8. 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. We note that ULARA remains stable in the presence of noise for RealityMining and TwitterSecurity, but degrade in performance for EnronInc.

5.4. 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

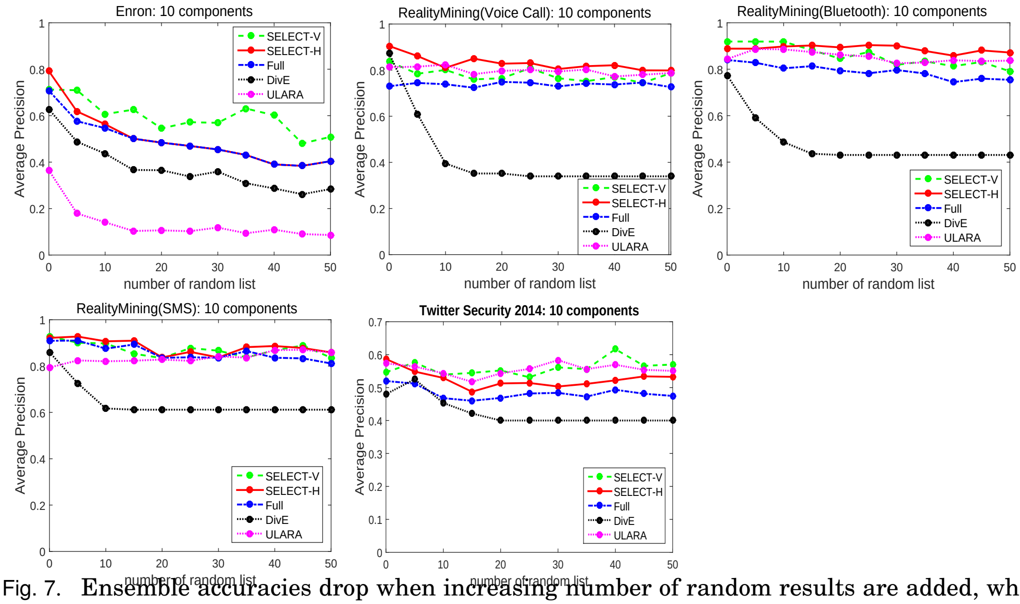

Fig. 7. Ensemble accuracies drop when increasing number of random results are added, where decrease is most prominent for DivE.

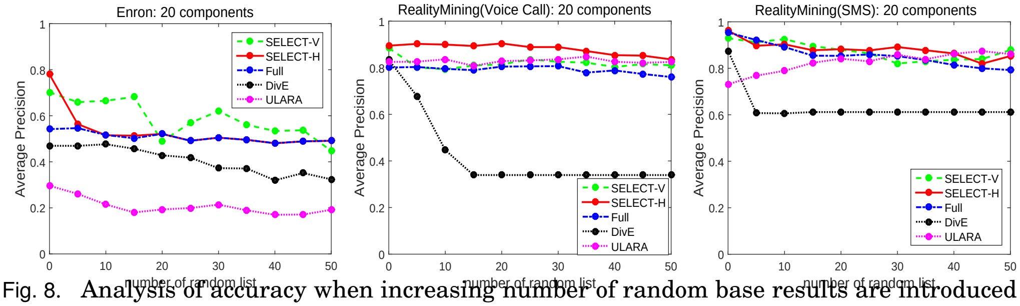

Fig. 8. Analysis of accuracy when increasing number of random base results are introduced for ensembles with 20 components. Decline in accuracy under noise is most prominent for DivE.

9 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 10 shows that a heavy clique with high degree nodes emerges in the graph structure at the time of the event. We also note that for the 9/11 WTC terrorist attacks in 2001, heavily linked entities e.g. World Trade Center, New York City, Washington (DC), George W. Bush, White House, Congress are rank at the top. The visualization of the change in Figure 11 shows that the heavy links emerge between the top ranked entities in the graph right after the event has occurred.

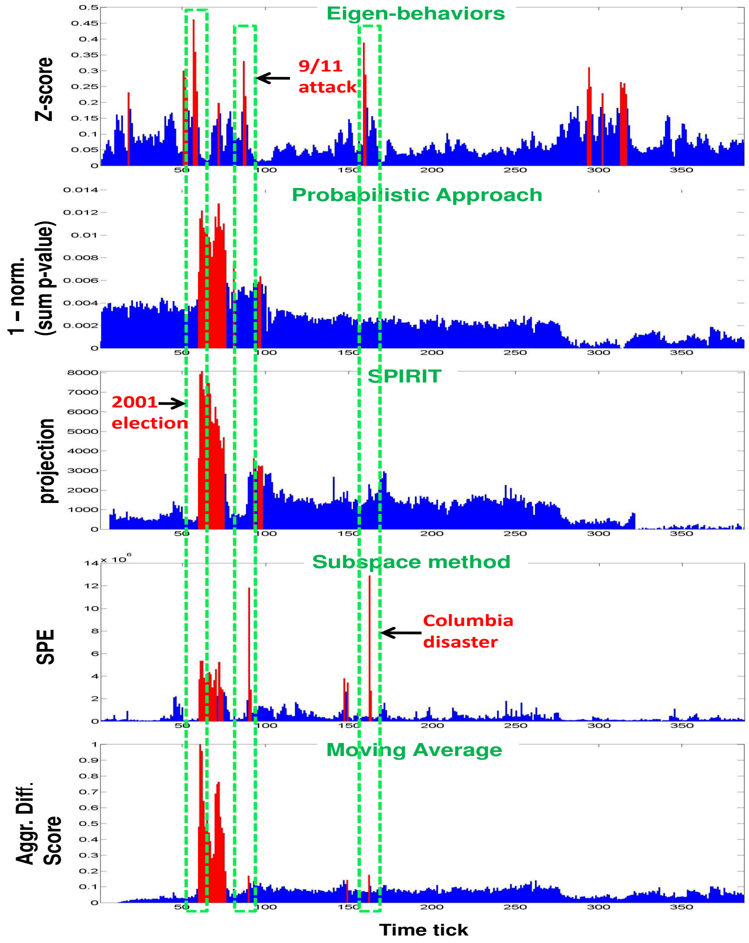

Fig. 9. Anomaly scores from five base detectors (rows) for NYT news corpus. red bars: top 20 anomalous time points per detector, green boxes: top 3 events by the final ensemble.

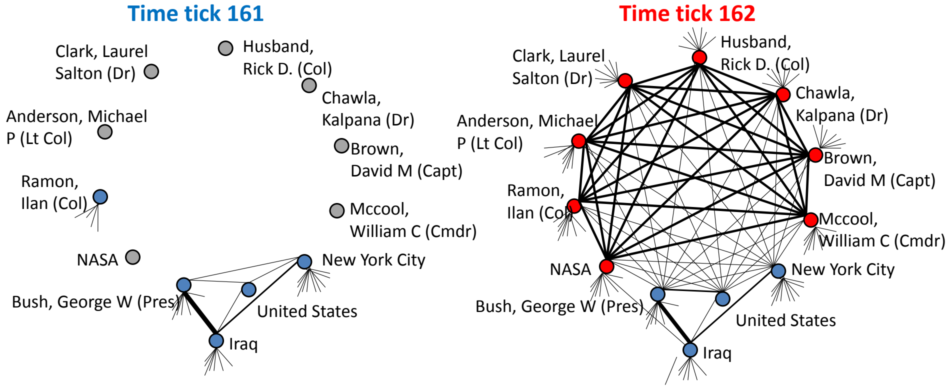

Fig. 10. During 2003 Columbia disaster a clique of NASA and the seven killed astronauts emerges from time tick 161 to 162.

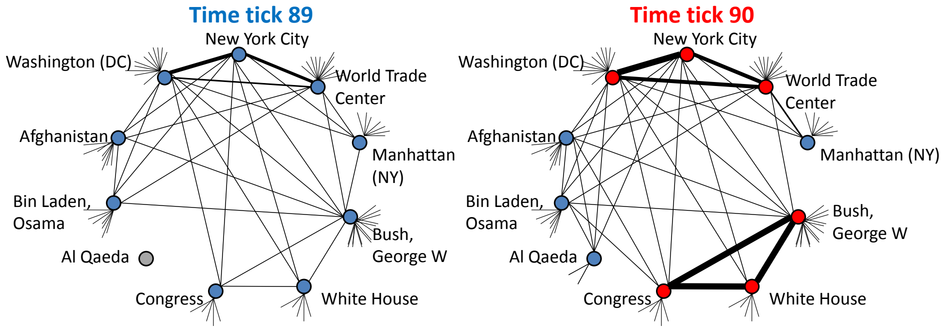

Fig. 11. During 2001, 9/11 terrorist attacks in WTC heavy links emerge between top ranked entities from time tick 89 to 90.

5.5. Outlier Detection Performance

Next we quantitatively evaluate the ensemble methods on outlier detection accuracy. Here again we utilize the *average precision*, as the measure of accuracy. We utilize different values of the parameter k (number of nearest neighbors) for the individual based detectors to provide 25 base components for the outlier ensemble. For Cardio dataset we use $k = 5, 10, 50, 100, 500$, and for all the other datasets we use $k = 5, 10, 15, 20, 25$.

In Table XII we report the summary of results for different multi-dimensional point datasets containing the average precision values for SELECT and other existing ensemble approaches (Full, DivE and ULARA) to which we compare SELECT. Similar to the event detection results we further investigate the significance of SELECT and provide the results for random ensemble (RandE) where we randomly select the same number of components to assemble at each phase. We note that SELECT ensemble either SelectHor SelectVprovides superior results to RandE, Full, DivE, and ULARA. Moreover, SelectH appears to be a better strategy than SelectV, as in most cases it provides the best result(in Table XII). Whereas, in some cases SelectV fall short to even beat Full, DivE, and ULARA. For brevity we skip the detailed accuracy tables and noise analysis plots as provided for event detection results.

6. RELATED WORK

Event detection in temporal data and outlier detection in multi-dimensional point data are fundamental research problems that find numerous applications in the real world. As such, a large body of research has focused on building effective techniques for these problems. For information on such techniques we refer the reader to detailed surveys [Akoglu et al. 2014; Chandola et al. 2009; Gupta et al. 2014] and devote this section to discuss related work on ensemble learning, which is the main focus of our work.

Ensemble techniques leverage multiple different methods to obtain better performance than the individual methods in the ensemble [Rokach 2010]. This is achieved by combining the strengths of accurate methods and alleviating the weaknesses of the less accurate ones. For example, boosting [Schapire 1990] and stacking [Wolpert 1992] directly integrate accuracy estimation within their iterative ensemble learning. Others assign expertise weights proportional to the accuracy of independent learners and externally combine their results [Hoeting et al. 1999; Juditsky et al. 2008; Monteith et al. 2011; Tsybakov 2004]. Ensembles are also known to produce more robust results. For example, bootstrap aggregating (or bagging) tends to reduce problems related to over-fitting to the training data [Breiman 1996].

Table XII. Significance of accuracy results compared to random ensembles with same number of selected components as SELECTfor outlier detection. The accuracy of the alterntive approaches (Full, DivE, and ULARA) are also given in parentheses. These results show that (i) SELECT is superior to existing methods, and (ii) it selects significantly more important (i.e., accurate) detectors to combine.

	Accuracy	significance
WBC (25 comp.) (Full: 0.3283, DivE: 0.2416, ULARA: 0.2439)		
(i) RandE (19/25, 6/7)	0.3209 (μ)	0.0091 (σ)
SelectV	0.2950	$= \mu - 2.8462\sigma$
(ii) RandE (21/25, 6/7)	0.3256 (μ)	0.0093 (σ)
SelectH	**0.3840**	$= \mu + 6.2796\sigma$
Glass (25 comp.) (Full: 0.2134, DivE: 0.1326, ULARA: 0.2022)		
(i) RandE (19/25, 3/7)	0.2032 (μ)	0.0224 (σ)
SelectV	0.2079	$= \mu + 0.2098\sigma$
(ii) RandE (21/25, 5/7)	0.2136 (μ)	0.0060 (σ)
SelectH	**0.2194**	$= \mu + 0.9667\sigma$
Lympho (25 comp.) (Full: 0.7287, DivE: 0.7593, ULARA: 0.7121)		
(i) RandE (15/25, 1/7)	0.6199 (μ)	0.1334 (σ)
SelectV	0.6347	$= \mu + 0.1109\sigma$
(ii) RandE (24/25, 6/7)	0.7488 (μ)	0.0291 (σ)
SelectH	**0.7843**	$= \mu + 1.2199\sigma$
Musk (25 comp.) (Full: 0.1195, DivE: 0.0981, ULARA: 0.1106)		
(i) RandE (14/25, 6/7)	0.1228 (μ)	0.0233 (σ)
SelectV	**0.1355**	$= \mu + 0.5451\sigma$
(ii) RandE (18/25, 5/7)	0.1162 (μ)	0.0239 (σ)
SelectH	0.1138	$= \mu - 0.1004\sigma$
Cardio (25 comp.) (Full: 0.3202, DivE: 0.2932, ULARA: 0.2331)		
(i) RandE (11/25, 6/7)	0.2709 (μ)	0.0185 (σ)
SelectV	0.2767	$= \mu + 0.3135\sigma$
(ii) RandE (19/25, 6/7)	0.3193 (μ)	0.0185 (σ)
SelectH	**0.4389**	$= \mu + 6.4649\sigma$
Thyroid (25 comp.) (Full: 0.0815, DivE: 0.0773, ULARA: 0.0925)		
(i) RandE (1/25, 0/7)	0.1445 (μ)	0.1036 (σ)
SelectV	**0.2342**	$= \mu + 0.9765\sigma$
(ii) RandE (20/25, 4/7)	0.1054 (μ)	0.0213 (σ)
SelectH	0.1412	$= \mu + 1.6808\sigma$
Letter (25 comp.) (Full: 0.4292, DivE: 0.4335, ULARA: 0.4298)		
(i) RandE (25/25, 6/7)	0.4303 (μ)	0.0043 (σ)
SelectV	0.4286	$= \mu - 0.3953\sigma$
(ii) RandE (18/25, 6/7)	0.4489 (μ)	0.0110 (σ)
SelectH	**0.5504**	$= \mu + 9.2273\sigma$

Thanks to these advantages, ensemble learning has spurned a large body of work devoted to the study of ensemble classification and clustering [Dietterich 2000; Fern and Brodley 2003; Fern and Lin 2008; Ghosh and Acharya 2013; Hadjitodorov et al. 2006; Hansen and Salamon 1990; Preisach and Schmidt-Thieme 2007; Topchy et al. 2005; Valentini and Masulli 2002; Verma and Rahman 2012; Zaman and Hirose 2011]. On the other hand, building effective ensembles for anomaly detection has remained to be a challenging task, due to lack of ground truth and inherent objective functions [Aggarwal 2012; Zimek et al. 2013a]. As such, there exist only a handful of mostly recent works on building anomaly ensembles [Gao et al. 2012; Gao and Tan 2006;

Kriegel et al. 2011; Lazarevic and Kumar 2005; Rayana and Akoglu 2014; Schubert et al. 2012; Vu et al. 2010; Zimek et al. 2013b].

Feature bagging [Lazarevic and Kumar 2005] is the earliest work formalizing an outlier ensemble. It uses the same base algorithm (i.e., LOF [Breunig et al. 2000]) on different feature subsets and employs a rank based merging to create the final consensus. Feature bagging is better calibrated by [Gao and Tan 2006; Kriegel et al. 2011] which convert the outlier scores to probability estimates and use score based merging. Different from those, [Gao et al. 2012; Rayana and Akoglu 2014; Schubert et al. 2012; Vu et al. 2010; Zimek et al. 2013b] utilize heterogeneous detectors (e.g., LOF [Breunig et al. 2000], LOCI [Papadimitriou et al. 2003], k-distance, etc.).

Specifically, [Gao et al. 2012; Rayana and Akoglu 2014] unify and combine results from all detectors, [Vu et al. 2010] uses accuracy on synthetically generated datasets to assign weights to the detectors, [Klementiev et al. 2007] calculates relative weights for the ranklists returned by the base detectors in an unsupervised way and [Schubert et al. 2012; Zimek et al. 2013b] select the most diverse set of results to combine. Most of these methods are designed for outlier detection in clouds of data points. There exist very few anomaly ensemble approaches for data represented as graphs [Gao et al. 2011; Rayana and Akoglu 2014]. In [Gao et al. 2011] Jing et al. proposed a weighted ensemble approach by iteratively maximizing the probabilistic consensus among the output of the base detectors. In our previous work [Rayana and Akoglu 2014] we combine all the results from the base detectors uniformly which reveals the fact that too much diversity among the base detectors hurts the final ensemble. Similar problem can occur in data fusion where web data from different sources are combined to build a knowledge base. Too much noise in several individual sources hurts the knowledge base. Srivastava et al. [Dong et al. 2013] provide a greedy approach to select the correlated data sources for data fusion to remove the erroneous data sources.

7. 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 and outlier detection problem in multi-dimensional point data (no-graph). SELECT is a two-phase approach that combines multiple detector results and then multiple consensuses, respectively. Motivated by our earlier observations [Rayana and Akoglu 2014] 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, DivE, an existing ensemble [Schubert et al. 2012] that combines diverse, i.e., least correlated results, and ULARA, a weighted rank aggregation approach [Klementiev et al. 2007].

Our quantitative evaluation for both event and outlier ensemble 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 (6/8 in Table III and 5/7 in Table XII) 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 (6/8 in Table III and 5/7 in Table XII). Noise analysis for event detection further corroborates the fact that DivE selects inaccurate/noisy results for the sake of diversity and declines in accuracy much faster than the rest. Table III and XII also show that ULARA is worse than Full in 4/8 cases and in 5/7 cases respectively, suggesting no clear winner. In comparison, SelectV and SelectH respectively outperform Full in 7/8 and 8/8 cases for event detection, 6/7 and 2/7 cases for outlier detection. This suggests that while the Full ensemble is inferior in the presence of inaccurate detectors, as a selective ensemble SELECT, specifically SelectH is superior to existing approaches like DivE and ULARA.

Future work will investigate how to go beyond binary selection and estimate appropriate weights for the detector/consensus results. One can also continue to enhance SELECT with other detectors and consensus methods as they become available.

All source code of our methods and datasets used in this work are shared openly at http://shebuti.com/SelectiveAnomalyEnsemble/.

Acknowledgments

The authors thank the anonymous reviewers for their useful comments. This material is based upon work supported by the DARPA Transparent Computing Program under Contract No. FA8650-15-C-7561, Army Research Office Young Investigator Program under Contract No. W911NF-14-1-0029, National Science Foundation CAREER 1452425, IIS 1017181, and IIP1069147, an R&D grant from Northrop Grumman Aerospace Systems, and a gift from Facebook. Any conclusions expressed in this material are of the authors and do not necessarily reflect the views, expressed or implied, of the funding parties.

REFERENCES

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

Charu C Aggarwal and Saket Sathe. 2015. Theoretical Foundations and Algorithms for Outlier Ensembles. *ACM SIGKDD Explorations Newsletter* 17, 1 (2015), 24–47.

Leman Akoglu and Christos Faloutsos. 2010. Event Detection in Time Series of Mobile Communication Graphs. In *27th Army Science*.

Leman Akoglu, Hanghang Tong, and Danai Koutra. 2014. Graph-based Anomaly Detection and Description: A Survey. *DAMI* 28, 4 (2014). DOI:http://dx.doi.org/DOI10.1007/s10618-014-0365-y

Leo Breiman. 1996. Bagging Predictors. *Machine Learning* 24, 2 (1996), 123–140.

Markus M. Breunig, Hans-Peter Kriegel, Raymond T. Ng, and Jörg Sander. 2000. LOF: Identifying Density-Based Local Outliers.. In *SIGMOD*.

Gavin Brown, Jeremy L. Wyatt, Rachel Harris, and Xin Yao. 2005. Diversity creation methods: A Survey and Categorisation. *Information Fusion* 6, 1 (2005), 5–20.

A. Cameron and P.K. Trivedi. 1998. *Regression Analysis of Count Data* (1st ed.). Cambridge Univ. Press.

Varun Chandola, Arindam Banerjee, and Vipin Kumar. 2009. Anomaly detection: A survey. *ACM Comput. Surv.* 41, 3 (2009).

Thomas G. Dietterich. 2000. Ensemble Methods in Machine Learning.. In *Multiple Classifier Systems (Lecture Notes in Computer Science)*, Vol. 1857. Springer, 1–15.

Xin Luna Dong, Laure Berti-Equille, and Divesh Srivastava. 2013. Data Fusion: Resolving Conflicts from Multiple Sources.. In *WAIM*.

Cynthia Dwork, Ravi Kumar, Moni Naor, and D Sivakumar. 2001. Rank aggregation methods for the Web. In *WWW*.

Nathan Eagle, Alex (Sandy) Pentland, and David Lazer. 2009. Inferring friendship network structure by using mobile phone data. *PNAS* (2009).

Xiaoli Zhang Fern and Carla E. Brodley. 2003. Random Projection for High Dimensional Data Clustering: A Cluster Ensemble Approach.. In *ICML*. AAAI Press, 186–193.

Xiaoli Z. Fern and Wei Lin. 2008. Cluster Ensemble Selection.. In *SDM* (2008-06-11). SIAM, 787–797.

Jing Gao, Wei Fan, Deepak Turaga, Olivier Verscheure, Xiaoqiao Meng, Lu Su, and Jiawei Han. 2011. Consensus Extraction from Heterogeneous Detectors to Improve Performance over Network Traffic Anomaly Detection.. In *IEEE International Conference on Computer Communications Mini-Conference*.

Jun Gao, Weiming Hu, Zhongfei (Mark) Zhang, and Ou Wu. 2012. Unsupervised Ensemble Learning for Mining Top-n Outliers.. In *PAKDD*.

Jing Gao and Pang-Ning Tan. 2006. Converting Output Scores from Outlier Detection Algorithms to Probability Estimates.. In *ICDM*.

Joydeep Ghosh and Ayan Acharya. 2013. Cluster Ensembles: Theory and Applications. In *Data Clustering: Alg. and Appl.*

Manish Gupta, Jing Gao, Charu C. Aggarwal, and Jiawei Han. 2014. Outlier Detection for Temporal Data: A Survey. *IEEE Trans. Knowl. Data Eng.* 26, 9 (2014), 2250–2267. DOI:http://dx.doi.org/10.1109/TKDE.2013.184

Stefan Todorov Hadjitodorov, Ludmila I. Kuncheva, and Ludmila P. Todorova. 2006. Moderate diversity for better cluster ensembles. *Information Fusion* 7, 3 (2006), 264–275.

Lars Kai Hansen and Peter Salamon. 1990. Neural Network Ensembles. *IEEE Trans. Pattern Anal. Mach. Intell.* 12, 10 (1990).

Douglas M Hawkins. 1980. *Identification of outliers*. Vol. 11. Springer.

Jennifer A. Hoeting, David Madigan, Adrian E. Raftery, and Chris T. Volinsky. 1999. Bayesian model averaging: A tutorial. *Statist. Sci.* 14, 4 (1999), 382–417.

A. Juditsky, P. Rigollet, and A. B. Tsybakov. 2008. Learning by Mirror Averaging. *Annals of Stat.* 36, 5 (2008), 2183–2206. DOI:http://dx.doi.org/10.1214/07-AOS546

John Kemeny. 1959. Mathematics without numbers.. In *Daedalus*. 577–591.

Alexandre Klementiev, Dan Roth, and Kevin Small. 2007. An Unsupervised Learning Algorithm for Rank Aggregation.. In *ECML*.

Edwin M Knorr and Raymond T Ng. 1997. A Unified Notion of Outliers: Properties and Computation.. In *KDD*. 219–222.

Raivo Kolde, Sven Laur, Priit Adler, and Jaak Vilo. 2012. Robust rank aggregation for gene list integration and meta-analysis. *Bioinformatics* 28, 4 (2012), 573–580.

Hans-Peter Kriegel, Peer Kröger, Erich Schubert, and Arthur Zimek. 2009. LoOP: local outlier probabilities. In *CIKM*. ACM, 1649–1652.

Hans-Peter Kriegel, Peer Kröger, Erich Schubert, and Arthur Zimek. 2011. Interpreting and Unifying Outlier Scores.. In *SDM*. 13–24.

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

Anukool Lakhina, Mark Crovella, and Christophe Diot. 2004. Diagnosing Network-Wide Traffic Anomalies.. In *SIGCOMM*. 219–230.

Diane Lambert. 1992. Zero-inflated Poisson regression with an application to defects in manufacturing.. In *Technometrics*. 1–14.

Aleksandar Lazarevic and Vipin Kumar. 2005. Feature bagging for outlier detection.. In *KDD*. ACM, 157–166.

Barbora Micenkovä, Brian McWilliams, and Ira Assent. 2014. Learing Outlier Ensembles: The Best of Both Worlds - Supervised and Unsupervised.. In *KDD ODD^2 Workshop*.

Kristine Monteith, James L. Carroll, Kevin D. Seppi, and Tony R. Martinez. 2011. Turning Bayesian model averaging into Bayesian model combination.. In *IJCNN*. IEEE, 2657–2663.

Spiros Papadimitriou, Hiroyuki Kitagawa, Phillip B. Gibbons, and Christos Faloutsos. 2003. LOCI: Fast Outlier Detection Using the Local Correlation Integral. In *ICDE*. 315–326.

Spiros Papadimitriou, Jimeng Sun, and Christos Faloutsos. 2005. Streaming Pattern Discovery in Multiple Time-Series. In *VLDB*.

Oskar Perron. 1907. Zur Theorie der Matrices. In *Mathematische Annalen*.

Michael D. Porter and Gentry White. 2012. Self-Exciting Hurdle Models for Terrorist Actovity. In *The Annals of Applied Statistics*. 106–124.

Christine Preisach and Lars Schmidt-Thieme. 2007. Ensembles of Relational Classifiers. *Knowl. and Info. Sys.* 14 (2007), 249–272.

Shebuti Rayana and Leman Akoglu. 2014. An Ensemble Approach for Event Detection in Dynamic Graphs.. In *KDD ODD^2 Workshop*.

Lior Rokach. 2010. Ensemble-based classifiers. *Artif. Intell. Rev.* 33, 1-2 (2010), 1–39.

Robert E. Schapire. 1990. The Strength of Weak Learnability. In *Machine Learning*. 197–227.

Erich Schubert, Remigius Wojdanowski, Arthur Zimek, and Hans-Peter Kriegel. 2012. On Evaluation of Outlier Rankings and Outlier Scores.. In *SDM*. 1047–1058.

Alexander P. Topchy, Anil K. Jain, and William F. Punch. 2005. Clustering Ensembles: Models of Consensus and Weak Partitions. *IEEE Trans. Pattern Anal. Mach. Intell.* 27, 12 (2005), 1866–1881.

Alexandre B. Tsybakov. 2004. Optimal aggregation of classifiers in statistical learning. *Annals of Statistics* 32 (2004), 135–166.

Giorgio Valentini and Francesco Masulli. 2002. Ensembles of Learning Machines.. In *WIRN*.

Brijesh Verma and Ashfaqur Rahman. 2012. Cluster-Oriented Ensemble Classifier: Impact of Multicluster Characterization on Ensemble Classifier Learning. *IEEE Trans. Knowl. Data Eng.* 24, 4 (2012), 605–618.

Nguyen Hoang Vu, Hock Hee Ang, and Vivekanand Gopalkrishnan. 2010. Mining Outliers with Ensemble of Heterogeneous Detectors on Random Subspaces.. In *DASFAA*, Vol. 5981. 368–383.

Quang H. Vuong. 1989. Likelihood Ratio Tests for Model Selection and non-nested Hypotheses. In *Econometrica*.

D. H. Wolpert. 1992. Stacked Generalization. *Neural Networks* 5 (1992), 214–259.

Faisal Zaman and Hideo Hirose. 2011. Classification Performance of Bagging and Boosting Type Ensemble Methods with Small Training Sets. *New Gen. Comput.* 29, 3 (2011), 277–292.

Ke Zhang, Marcus Hutter, and Huidong Jin. 2009. A new local distance-based outlier detection approach for scattered real-world data. In *PAKDD*. Springer, 813–822.

Arthur Zimek, Ricardo J.G.B. Campello, and Jörg Sander. 2013a. Ensembles for Unsupervised Outlier Detection: Challenges and Research Questions. *SIGKDD Explor. Newsl.* 15, 1 (2013), 11–22. DOI:http://dx.doi.org/10.1145/2594473.2594476

Arthur Zimek, Matthew Gaudet, Ricardo J. G. B. Campello, and Jörg Sander. 2013b. Subsampling for efficient and effective unsupervised outlier detection ensembles.. In *KDD*. ACM, 428–436.