ND-SYNC: Detecting Synchronized Fraud Activities

Maria Giatsoglou[1], Despoina Chatzakou[1], Neil Shah[2], Alex Beutel[2], Christos Faloutsos[2], and Athena Vakali[1]

[1] Informatics Department, Aristotle University of Thessaloniki, Greece,
{mgiatsog, deppych, avakali} @csd.auth.gr,
[2] School of Computer Science, Carnegie Mellon University, USA,
{neilshah, abeutel, christos} @cs.cmu.edu

Abstract. Given the retweeting activity for the posts of several Twitter users, how can we distinguish organic activity from spammy retweets by paid followers to boost a post's appearance of popularity? More generally, given groups of observations, can we spot strange groups? Our main intuition is that organic behavior has more variability, while fraudulent behavior, like retweets by botnet members, is more synchronized. We refer to the detection of such *synchronized* observations as the *Synchonization Fraud* problem, and we study a specific instance of it, *Retweet Fraud Detection*, manifested in Twitter. Here, we propose: (A) ND-SYNC, an efficient method for detecting *group fraud*, and (B) a set of carefully designed features for characterizing retweet threads. ND-SYNC is *effective* in spotting retweet fraudsters, *robust* to different types of abnormal activity, and *adaptable* as it can easily incorporate additional features. Our method achieves a 97% accuracy on a real dataset of 12 million retweets crawled from Twitter.

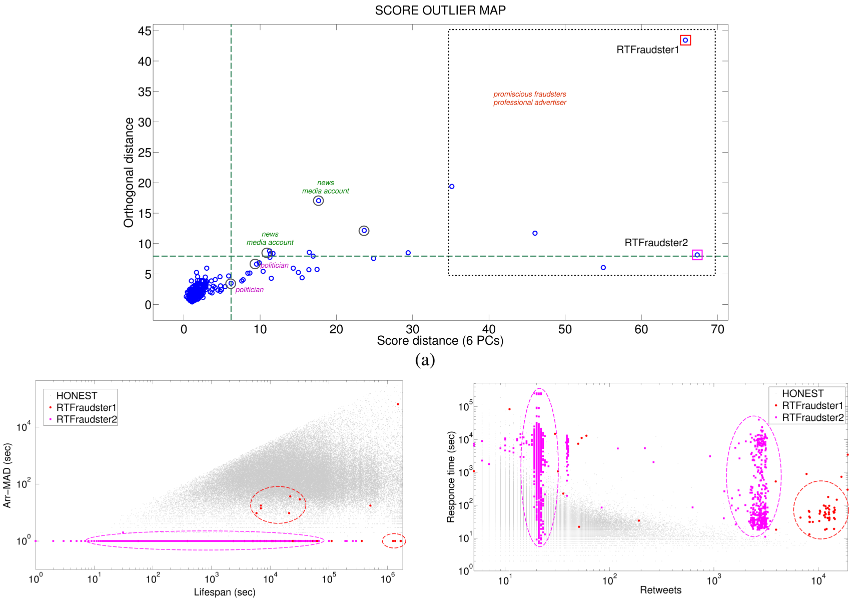
1 Introduction

Suppose that Twitter user "John" posts a tweet, and in 10 minutes it is retweeted by 3,000 people. Is this suspicious? Not necessarily. Suppose that this happens for the next 5 tweets of John: roughly the same 3,000 people all retweet his post in a few minutes.

Now, *that* is suspicious because such synchronized behavior among so many users is not natural. This is exactly the main intuition behind our work: events (like retweet threads) that belong to the same group are suspicious if they are highly synchronized; that is, if all of the retweet threads for John's tweets are synchronized. The challenge therefore is, given many events belonging to many groups (retweet threads for user "Jane", etc.), find the suspicious groups. We assume that organic behavior is the norm, and deviations from it constitute suspicious (anomalous, fraudulent) behavior.

Anomaly and outlier detection has attracted a lot of interest, both at the individual level (e.g. a single retweet thread in our example) [20, 3, 12] as well as *group* [25] or *collective anomalies* [6]. While most anomaly detection focused on a cloud of p-dimensional points (entities), extensions to more complex data have been proposed [6], such as for rare subsequences, subgraphs, and image subregions. Here, we propose a novel, general approach to collective anomaly detection, informally defined as follows:

Informal Problem 1 (Synchronicity Fraud Detection)
Given*: N groups of entities; a representation for each entity in a p-dimensional space;*
Identify *groups of entities abnormally synchronized in some feature subspaces.*



(a)

(b) Lifespan vs. Arr-MAD

(c) Retweets vs. Responce time

Fig. 1: **ND-SYNC detects outliers clearly separated from the majority of users.** In (1a), points right of the vertical dashed line are spotted as RT fraudsters. (1b), (1c) reveal high synchronicity in the retweet threads of users RTFraudster1 and RTFraudster2, for different feature combinations.

At a high level, our proposed ND-SYNC methodology is as follows:

1. Extract p features from each entity (i.e. retweet thread), such as the average inter-arrival time of retweets, variance of it, and number of retweets.
2. Analyze the collective behavior of each group (the set of retweet threads for posts from a given user), and compare it to the behavior of the rest of the threads. Using the concepts of "intra-synchronicity" and "inter-synchronicity" (see Section 4.2), assign a suspiciousness-score to each user in all 2^p available subspaces.
3. Combine the scores for each user and report the most suspicious such groups.

Figure 1 illustrates the effectiveness of ND-SYNC in detecting Twitter accounts whose posts trigger fake retweet threads. In Figure 1a each point is a user plotted in terms of two dimensions that reflect deviation from normal users' behavior. The vertical dashed line clearly separates fraudulent from normal users. The most "anomalous" users (at the rightmost part of the figure) correspond to bot accounts acting as professional promoters of content or other users. Figures 1b and 1c depict the retweet threads of normal users (grey points) along with the retweet threads of the "significantly outlying" caught users RTFraudster1 and RTFraudster2 projected in 2-D subspaces with respect to two pairs of our proposed features. Compared to honest retweet threads, we can clearly observe fraudulent users' retweet threads are abnormaly clustered together.

The contributions of this work are the following:

- **Methodology**: ND-SYNC is a general, effective pipeline that automatically detects group anomalies.
- **Feature engineering**: we customize ND-SYNC for the case of retweet fraud, using a carefully selected set of features.

Reproducibility: We share our (anonymized) data at: `http://oswinds.csd.auth.gr/project/NDSYNC`.

2 Related Work

We first discuss approaches addressing fraud detection in Twitter, and then review efforts on collective anomaly detection. Table 1 compares ND-SYNC to the methods that are most relevant to our problem setting. ND-SYNC does not require textual content, graph structure, or user attributes (such as account creation date) to detect fraudsters.

Table 1: ND-SYNC comparison against alternatives. (*: in multiple feature subspaces, **: searches the best, ***: in a single 2D feature space)

		Textual content-agnostic?	Graph structure-agnostic?	User attributes-agnostic?	Parameter free?	Unsupervised?	Designed for RT-FRAUD?	Detects synchronicity fraud?
ND-SYNC		✓	✓	✓	✓	✓	✓	✓*
Twitter Fraud Detection	[23]	✓	✓	✓	✓			
	[27]			✓		✓		
	[7]				✓			
	[11]	✓	✓	✓	✓**	✓	✓	
Collective Anomaly Detection	[25]	✓	✓	✓				
	[28]	✓		✓				
	[16]	✓		✓	✓	✓		✓***

Fraud on Twitter: *Retweet Fraud Detection* identifies Twitter users that obtain fake retweets for their posted content (RT fraudsters). Fraud is a serious problem on Twitter [24], with fraudsters exhibiting several classes of strange behaviors [8], and varying degrees of automation. [9] lists several such attempts of fraudsters to mimick organic behavior, for example by re-broadcasting others' posts. Earlier work focuses on account tweeting activity and/or social connectivity. [27] analyzes the relationships of fraudsters inside and outside of criminal communities to infer types of accounts serving as criminal supporters. [7, 8] leverage tweeting behavior, tweet content and account properties to compute the likelihood of an unknown user being a human, bot or cyborg. In general, such feature-based methods (e.g. username pattern, age) have been shown to fail to catch more sophisticated fraud schemes that exploit real users' accounts, such as the *pyramid follower markets* [22] and *account compromization* [1]. [23] shows the effectiveness of temporal features for distinguishing between account types. [11] addresses a problem similar to ours but uses the URLs found in tweets instead of retweet threads in conjunction with a time and user-based entropy to classify posting activity and content.

Collective anomaly detection: The goal of collective anomaly detection is to find groups of entities that jointly exhibit abnormal behavior. Variants exist for sequential [5], spatial [13] and graph [19] data, while other approaches are more general, simply assuming p-dimensional points [28, 26, 25]. Synchronized ("lockstep") behavior is often an indication of fraud, like e.g. users Liking the same Facebook Pages at the same time [2] or following the same accounts [21, 17, 16].

3 Background

This section provides the necessary background for ND-SYNC: a formula to measure the suspiciousness of a group of 2-D points, given a large set of 2-D points (Section 3.1) and a robust multivariate outlier detection method (Section 3.2).

3.1 Measuring group strangeness

Given a large cloud of 2-D points $\mathcal{E}$, and a set of points $\mathcal{E}' \subset \mathcal{E}$, how unusual is $\mathcal{E}'$ with respect to $\mathcal{E}$? [16] gave such a score, namely, the *residual score* (rs_score, see Eq. 4 below). The definition needs some auxiliary concepts: *synchronicity* (how coherent/lockstep is the subset $\mathcal{E}'$) and *normality* (how similar it is to the presumed-normal cloud $\mathcal{E}$). In the next 3 paragraphs, we give their mathematical definitions, as well as the equation for a crucial lower-bound.

Synchronicity: For a group $\mathcal{E}'$, it is the average closeness between all pairs of its members

$$sync(\mathcal{E}') = \bar{c}(e, e') \qquad e, e' \in \mathcal{E}' \tag{1}$$

The closeness $c(e, e')$ of two entities e and e', is a similarity function - in [16], it was binary: after dividing the address space into grid-cells, the closeness is 1, if the elements are in the same grid-cell, and zero otherwise.

Normality: The *normality*, of a group $\mathcal{E}'$ with respect to a (superset) group $\mathcal{E}$, is the average closeness of the members of $\mathcal{E}'$ to the members of $\mathcal{E}$.

$$norm(\mathcal{E}') = \bar{c}(e, e') \qquad e \in \mathcal{E}, e' \in \mathcal{E}' \tag{2}$$

Lower bound: Given a group $\mathcal{E}'$, with normality value n with respect to a (superset) group $\mathcal{E}$, the synchronicity $sync(\mathcal{E}')$ is lower-bounded by $sync_{min}(n)$:

$$sync_{min}(n) = (-Mn^2 + 2n - s_b)/(1 - Ms_b) \tag{3}$$

where M is the count of non-empty grid-cells for $\mathcal{E}$, and s_b is the synchronicity of $\mathcal{E}$.

Residual score: For group $\mathcal{E}'$ with synchronicity $sync(\mathcal{E}')$ and normality $norm(\mathcal{E}')$ wrt $\mathcal{E}$, the **residual score** is given by:

$$rs\_score(\mathcal{E}') = sync(\mathcal{E}') - sync_{min}(norm(\mathcal{E}')) \tag{4}$$

3.2 Robust outlier detection

ROBPCA-AO [15] is a robust *Principal Component Analysis* (PCA) and outlier detection method that is suitable for multivariate, high-dimensional data and independent of their features' distribution. Proposed as a robust alternative to classic PCA, ROBPCA-AO identifies Principal Components (PCs) that best describe the uncontaminated data, while at the process, it detects outliers.

Initially, the method applies SVD on the set of observations to project them into the (restricted) space they span. Then, the dimensionality of data is reduced by keeping only the first k PCs of the data's covariance matrix. Taking into account the possibility of skewed data, ROBPCA-AO computes a robust k-subspace V_r that fits the majority of observations and projects them on it. Outliers are detected on subspace V_r based on two distance-based scores: (a) the *orthogonal distance* (*od*) of each observation from its projection on V_r, and (b) *robust score distance* (*sd*), which is taken as the *adjusted outlyingness* of the observation. *Adjusted outlyingness* is a measure suitable for multivariate, asymmetric data that estimates the distance of a given observation from the bulk of observations as its maximum robust distance (outlyingness) over B directions. Each such direction is perpendicular to the subspace spanned by k randomly sampled observations. Observations whose od or sd score surpasses a data-dependent cutoff threshold (defined as the upper whisker of the *adjusted boxplot* [15]) are characterized as outliers.

4 Proposed method: ND-SYNC

This section first outlines *Retweet Fraud Detection* as a special case of the *Synchronicity Fraud* problem, and then presents ND-SYNC, an effective and robust solution.

4.1 Problem Defintition

Retweet fraud detection (RTFRAUD) is a problem of various dimensions since: (a) it can be practiced by different types of user accounts (automated or bot orchestrated, semi-automated, human managed), (b) the inflation of content's popularity can be the sole purpose of the suspected user account or an occasional tactic hidden (*camouflaged*) in organic activities, (c) the promoted content can attract both fake and honest retweets. Thus, it is important to find features able to separate such diverse fraudulent activities from honest user behavior, and to design a method that can effectively leverage their variety. To study the retweeting activity in terms of time and retweeting users, given a user u_m (*author*) we represent the i^{th} tweet posted by u_m with $tw_{m,i}$ as a tuple $(u_m, t_{m,i})$, where $t_{m,i}$ is the tweet's creation time. A retweet thread is defined as follows:

Definition 1 (Retweet thread). *Given an author u_m and a tweet $tw_{m,i}$, a retweet thread $R_{m,i}$ is defined as the set of all tweets that retweeted $tw_{m,i}$.*

Here, we formulate RTFRAUD as an instance of the *Synchronicity fraud* (SYNCFRAUD) problem, which is defined at two levels (group and entity) below:

*Problem 1 (*SYNCFRAUD*).*

Given: N groups of entities G, where each group $g_m \in G$ comprises a variable number of entities $e_{m,i} \in E_m$, and a set of p features for the entities' representation,
Extract: a set of features at the group-level, and
Identify: suspicious groups S that exhibit highly *synchronized* characteristics.

Even though essentially, the RTFRAUD problem involves three levels instead of two – lower (individual retweets), middle (retweet threads) and upper (users) – we simplify it by collapsing a post's retweets into a single retweet thread and by defining features for its characterization. Then, RTFRAUD can be directly mapped to the SYNCFRAUD problem where each user u_m has a group g_m containing all retweet threads $R_{m,i}$ for that user, and the suspicious groups S are the detected fraudsters (RT fraudsters).

4.2 Proposed Approach

In this section we provide the pipeline of ND-SYNC, our approach to the SYNCFRAUD problem, and describe its steps. Then, we propose a 7-dimensional feature space for representing the retweet threads in the RTFRAUD problem.

ND-SYNC pipeline. ND-SYNC comprises three main steps: (1) *Feature subspace sweeping*, which generates and bins all entity-wise feature subspaces and projects entities in them; (2) *User scoring*, which calculates the group-based suspiciousness score vectors; (3) *Multivariate outlier detection*, which identifies suspicious groups based on their deviation from normal behavior. ND-SYNC's pipeline is outlined in Algorithm 1.

```
Data: $E = \{e_i\}$ : Set of $p$-dimensional entities, $G = \{g_m\}$ : Set of $N$ groups where
      $g_m = \{e_{m,i}\}$, $I$ : number of iterations
Result: $S$ : suspicious groups
generate all $2^p$ subspaces $F$ and project $e_i$ in them ;
logarithmically segment all subspaces and assign $e_i$ to bins ;
for each group $g_m$ in $G$ do
    for each feature subspace $f_i$ in $F$ do
        calculate suspiciousness $susp(g_m, f_i)$ (Def. 2);
    combine $susp$ over all subspaces in vector $SU(g_m)$ (Def. 3);
robustly scale and center $SU(g_i)$ vectors, $i \in [1, N]$ ;
for $iter = 1 : I$ do
    extract set of outliers $S_{iter}$ from $SU(g_i)$ applying multivariate outlier detection ;
extract final $S$ set by applying majority voting on all $S_{iter}$ ;
```

Algorithm 1: Pseudocode for ND-SYNC

Feature subspace sweeping. How can we detect microclusters in a p-dimensional space? Given N groups of entities represented in a $p-$feature space, ND-SYNC first: (a) projects all entities into the desired feature subspaces, and then (b) reduces the statistical noise in each subspace to prepare the data for synchronicity detection. Given a p-feature space, we take all possible q-feature combinations for $q \in [1, p]$; this produces 2^p possible subspaces to analyze. In anomaly detection, it is difficult to estimate apriori the most effective feature combinations for discriminating suspicious groups from normal ones. Thus, a straightforward approach is to generate and apply ND-SYNC's next steps on all 2^p feature subspaces. In cases when p is too high, though, practitioners can select a subset of subspaces for extra efficiency. As we show in Section 5.2, considering only the 3-D and 2-D subspaces is relatively effective for RTFRAUD.

To compute the suspiciousness score, we need to bin the feature space (see subsection 3.1); we choose logarithmic binning, in powers of 2, for each dimension/feature.

User scoring. At this point, we need to combine the entity-level features in a way that reflects the synchronicity of the group. These group features should allow us to identify the suspicious set S from the normal groups. Here, on each entity-feature subspace f_i, where $i \in [1, 2^p]$, we calculate the *intra-synchronicity* $intra\_sync(g_m, f_i)$ and *inter-synchronicity* $inter\_sync(g_m, f_i)$ measures for each group of entities g_m, based on Eq. 1 and Eq. 2, respectively.

We expect that suspicious groups will have significantly higher intra-synchronicity compared to normal groups. However, depending on the feature subspace, the deviation between the intra-synchronicity of suspicious and normal groups may vary (due to differences in the features' discriminative power and distribution). Thus, given $F = 2^p$ feature subspaces, we generate an F-dimensional feature vector for each group g_m, i.e. the *suspiciousness score* vector, SV, where each dimension represents the group's *suspiciousness*, $susp$, in the corresponding subspace (given in Definition 2). The $susp$ scores correspond to our user-level features.

Definition 2 (Suspiciousness). *For subspace f_l and group g_m with intra-synchronicity $intra\_sync(g_m, f_l)$ and inter-synchronicity $inter\_sync(g_m, f_l)$, the group's suspiciousness is given by $\mathbf{susp(g_m, f_l)} = rs\_score(projection(g_m, f_l))$.*

Definition 3 (Suspiciousness score vector). *For group g_m and $i \in [1, F]$, the suspiciousness vector is given by $\mathbf{SV(g_m)} = [susp(g_m, f_i)]_{i=1}^{F}$.*

Unlike previous approaches, here we do not assume that suspicious groups are characterized by low inter-synchronicity. We claim that inter-synchronicity is more difficult to interpret as an indication of normality vs. suspiciousness, since its value depends on the selected features and their distribution over all entities in normal and suspicious groups. For example, our experiments on RTFRAUD indicated that normal users are approximately at the same scale of inter-synchronicity, whereas suspicious users can have either very low values (retweet threads have rare feature values) or very high (retweet threads of several suspicious users have the same feature values, e.g. zero inter-arrival time of retweets; for normal users the corresponding values are more diverse).

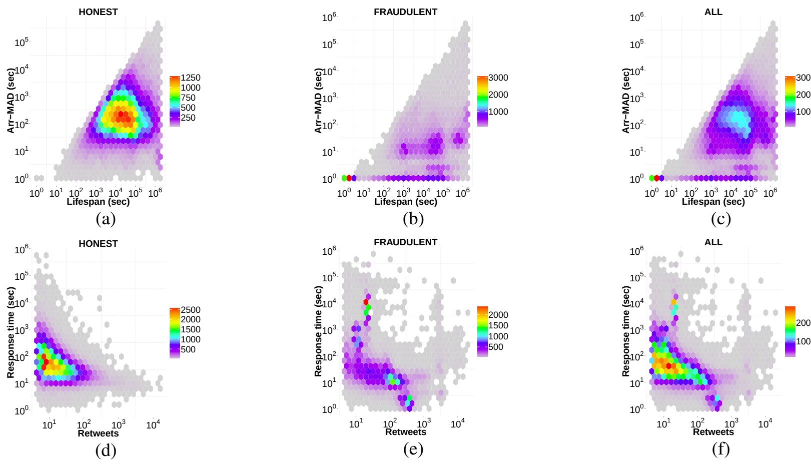
Multivariate outlier detection. The last step of ND-SYNC takes as input the set of groups G with their F-dimensional suspiciousness score vectors, and proceeds to the identification of the suspicious groups S. First, we standardize the vectors using their *median* and *mean average deviation* (MAD), which are considered as robust estimators of center and scale. ND-SYNC then spots as suspicious the groups that largely deviate from the majority of groups in G, considering their standardized scores in all feature subspaces, based on the outlier detection approach described in Section 3.2.

To address the non-deterministic nature of this outlier detection method, mainly in terms of entities positioned close to the distance cutoffs, ND-SYNC applies it iteratively, maintains a list of the identified outliers in each iteration, and classifies a group as suspicious based on the majority vote over all runs. Our experiments on RTFRAUD revealed that even a small number of iterations (e.g. 10) is enough to estimate the suspicious groups. Moreover, to eliminate the need for selecting parameters in ND-SYNC:

1. We propose automatic selection of the k principal components via a heuristic technique such as the *95% cumulative variance explained* criterion [18]. According to this criterion, during dimentionality reduction, the first k components that together explain more than 95% of the data's variance are maintained;
2. We use all entities, instead of a subset of them (as in the approach of Section 3.2), to estimate the robust feature subspaces. This is reasonable since typically the percentage of outliers in a dataset is small, and difficult to estimate apriori.

Feature Engineering for Retweet Threads. Earlier works have associated bot activity with temporal activity anomalies, such as low entropy in the time intervals between posts. Here, we also expect that the retweet threads of RT fraudsters will exhibit different inter-synchronicity and high intra-synchronicity compared to honest users, with respect to their temporal characteristics. In addition, due to automation tools, as well as due to the way retweet markets operate, we expect RT fraudsters to be synchronized in terms of the number of retweets their posts receive. Based on the above, after experimenting with several features which are omitted here for brevity, we ended up with the following features for a retweet thread's representation:

- **Retweets**: number of retweets
- **Response time**: time elapsed between the tweet's posting time and its first retweet
- **Lifespan**: time elapsed between the first and last (observed) retweet, constrained to 3 weeks to remove bias with respect to later tweets
- **RT-Q3 response time**: time elapsed after the tweet's posting time to generate the first 3 quarters of the (observed) retweets
- **RT-Q2 response time**: time elapsed after the tweet's posting time to generate the first half of the (observed) retweets
- **Arr-MAD**: mean absolute deviation of inter-arrival times for retweets
- **Arr-IQR**: inter-quantile range of inter-arrival times for retweets



(a) (b) (c)

(d) (e) (f)

Fig. 2: **Synchronicity patterns are revealed as microclusters in RT fraudsters behavior**. a, b, c (d, e, f) correspond to the Lifespan vs. Arr-MAD (Retweets vs. Response time) scatter plots for "honest", "fraudulent", and both types of users. The majority of organic retweet threads are clustered around a limited range of values, clearly separated from the fraudsters' microclusters.

To estimate the suitability of the proposed features for revealing retweet fraud, we examined the projections of the retweet threads of the Twitter dataset described in Section 5.1 in all 2-D feature subspaces derived by the proposed feature set. To assist visualization, we binned all feature subspaces and generated 2-D heatmaps in logarithmic scales.

Figure 2 indicatively depicts the scatter plots of *Lifespan vs. Arr-MAD* and *Retweets vs. Response time* for all dataset's users (Fig. 2c and 2f), and only for those annotated as "honest" (Fig. 2a and 2d) and "fraudulent" (Fig. 2b and 2e) .

The *Lifespan vs. Arr-MAD* plots reveal microclusters of fraudulent retweet threads (at very low values of Arr-MAD and at high Lifespan values), whereas the majority of honest users' retweet threads seem to be concentrated around a certain area of the feature subspace, clearly separated from RT fraudsters. Similar observations are made from the *Retweets vs. Response time* plots where we observe a microcluster at abnormaly low values for both features, and another one for high Response time. Some of these results were anticipated based on our intuition, e.g. bots of the same network may retweet all at once, having on average a zero Arr-MAD, but it seems that our proposed features can reveal more complex retweet fraud practices. For example, promoted posts may continue to receive retweets for a prolonged period of time, which explains the microcluster of long Lifespan, whereas certain RT fraudsters may wait for some time, after posting their tweets, before applying to some retweet market for their promotion.

5 Experiments

In this section, we evaluate ND-SYNC by conducting a series of experiments on a dataset crawled from Twitter which is comprised of over *130K retweet threads* characterizing more than *11M retweets* to posts of several hundred active Twitter users. We detail our data collection approach, describe the settings of our numerous experiments, and finally present the performance of our ND-SYNC.

5.1 Twitter Dataset

The evaluation of ND-SYNC requires a dataset of several, *complete* retweet threads of honest and fraudulent users, i.e. with no gaps in the tuples representing a given post's retweets. Due to the Twitter Streaming API's constraint of allowing access to only a sample of published tweets, our requirement for complete retweet threads, and the lack of a relevant (labeled) dataset, we manually selected a set of target users for whom we could track all tweets and retweets in a given time period.

Target user accounts were selected in several fashions. Firstly, we examined a recent, 2-day sample of the global Twitter timeline and identified the users who posted the most retweeted tweets and those who posted tweets containing keywords heavily used in spam campaigns (*casino*,*followback*, etc). Our next approach involved selecting users based on "Twitter Counter"[3], a web application that publishes lists ranking Twitter users based on criteria including follower count and number of tweets. We chose users based on their posting frequency and influence – specifically, we kept only users who tweeted several times per week and received more than 100 retweets on their recent posts. Lastly, we collected users active in specific topics (European affairs and Automobile), given that they were added in such topic-related lists by other Twitter users.

We manually labeled target users as "fraudulent" in cases where (a) inspection of their tweets' content led to the discovery of spammy links to external web pages, spam-related keywords, and multiple posts with similar promotions or vacuous content (e.g. quotes), and (b) profile information was clearly fabricated. The rest of the target users were labeled as "honest". We monitored the set of target users for time periods spanning from 2 to 6 months and eliminated those who had less than 20 retweet threads or a

[3] http://twittercounter.com/

maximum-length retweet thread of less than 50 retweets. This process left us with a total of 298 users in the dataset, of which 270 were labeled honest and 28 labeled fraudulent. For each user, we extracted the retweet threads and mapped them to our proposed 7-D feature space. The dataset includes **134,022 retweet threads** (83,587 with respect to honest and 50,435 of fraudulent users) which in total comprise **11,727,258 retweets** (2,939,455 to posts of honest and 8,787,803 to posts of fraudulent users).

5.2 Results

Next, we present the experimental results of ND-SYNC's application on the Twitter dataset described above. We discuss its effectivess when ND-SYNC was applied on (a) all available 2^7 feature subspaces, (b) a restricted subset of the feature subspaces.

Preliminary observations on the data. Before applying the final outlier detection step of ND-SYNC, we examine the distribution of standardized user-level scores for honest users with respect to each feature subspace. All score variables were found to be significantly asymmetric, having a *medcouple*[4] [4] between 0.16 and 0.46, and the corresponding p-values rejected the normality hypothesis at the 1% significance level. If honest users' scores had symmetric distributions, we could apply typical thresholded outlier detection techniques that assume normal distribution [10, 14]. However, in our case, the skewness of the "uncontaminated" data would likely result in many false positives. Conversely, ND-SYNC's outlier detection approach is rather suitable for the RT-FRAUD problem.

Detection effectiveness on real data. Figure 3 shows ND-SYNC's performance on detecting RT fraudsters in terms of *F1-score* and *accuracy*. To examine robustness of ND-SYNC to the number of dimensions maintained (k) in the beginning of the outlier detection step, we provide perfomance measures for k from 1 to 10 and all feature subspaces considered for the users' suspiciousness estimation. We observe that ND-SYNC is relatively robust with respect to k: we attain $[95\% - 97\%]$ accuracy and $[0.73 - 0.82]$ F1-score. For $k = 6$, based on the *95% cumulative variance explained criterion*, ND-SYNC has the best performance with respect to precision-recall balance and accuracy. Our experiments with ND-SYNC considering only the 3-D and the 2-D feature subpaces showed effectiveness in catching several cases of fraud. Specifically, the accuracy (F1-score) with respect to the best performing ND-SYNC run in all feature subspaces was reduced only by 4.5% (0.4%) and 10.6% (1%) for 3-D and 2-D subspaces, respectively.

Observations on the outlier map. Figure 1a illustrates the outlier map of sd and od scores generated for the best run of ND-SYNC, where red lines correspond to adaptive cutoff values for sd and od – the plot clearly discerns the outliers from the majority of users which lie in the bottom-left region. All discovered outliers have an abnormal sd score, whereas 36% also have outlying od scores (in the top-right quartile of the figure).

[4] Medcouple measures the skewness of a distribution in $[0, 1]$ range. Right and left skewed distributions have positive and negative medcouple respectively; symmetric distributions have zero medcouple.

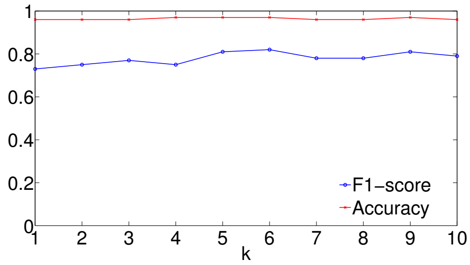


Fig. 3: **ND-SYNC is highly accurate and robust to the selection of k (number of dimensions maintained)**. The best performance is at $k = 6$ which explain approximately 95% of the data's cumulative variance.

A closer examination of RT fraudsters that were caught by ND-SYNC reveals that the ones who scored high in sd and od were exemplary bot accounts that are typically hired for promotion or advertisement. For example, RTFraudster1, enclosed within a rectangle in the top-right quartile of Figure 1a, is an example of such a *promiscuous* fraudster with 800 followers that had 65 retweet threads in a 4-month time period – 80% (60%) of these were comprised of more than 1k (10k) retweets and had almost 0 Arr-IQR[5]. The remainder of the "caught" RT fraudsters have a more *subtle* profile, resembling cyborg behavior: the accounts often create vacuous posts, but occasionaly interact genuinely with other users, thus indicating a human operator. We found that the five false positives detected by ND-SYNC (enclosed by a red circle in Figure 1a) belonged to media accounts and politicians. Three of these accounts have significantly abnormal sd and od scores, while the others are situated very close to RT fraudsters, suggesting that they may have tampered with the organic behavior of their retweet threads.

6 Conclusions

In this work, we broach the problem of discerning fraudulent, group-based activity from organic online behavior. The contributions of this work are the following:

- **Methodology**: we present ND-SYNC, a general, effective pipeline, which automatically detects group anomalies
- **Feature engineering**: a carefully designed set of features, that customize ND-SYNC for the case of retweet fraud.

We present experiments on real data from Twitter consisting of almost *12 million retweets*, where the proposed ND-SYNC achieved excellent classification accuracy of 97% in distinguishing fraudulent from honest users.

Reproducibility: For the reproducibility of our results we share an (anonymized) version of our data at: `http://oswinds.csd.auth.gr/project/NDSYNC`.

Acknowledgments. The authors would like to thank Dr Stephan Günnemann for his insightful comments and valuable discussions about this work.

References

1. A. Almaatouq, et al. Twitter: Who Gets Caught? Observed Trends in Social Micro-blogging Spam. In *WebSci*, 33–41. ACM, 2014.

[5] This account was later suspended by Twitter.

2. A. Beutel, et al. CopyCatch: Stopping Group Attacks by Spotting Lockstep Behavior in Social Networks. In *WWW*, 119–130. ACM, 2013.
3. M. Breunig, et al. LOF: Identifying Density-Based Local Outliers. In *Proc. ACM SIGMOD Conf. 2000*, 93–104, 2000.
4. G. Brys, et al. A Robust Measure of Skewness. *Journal of Computational and Graphical Statistics*, 13:996–1017, 2004.
5. P. K. Chan et al. Modeling Multiple Time Series for Anomaly Detection. In *ICDM*, 90–97. IEEE Computer Society, 2005.
6. V. Chandola, et al. Anomaly Detection: A Survey. *ACM Comput. Surv.*, 41(3):15:1–15:58, 2009.
7. Z. Chu, et al. Detecting Automation of Twitter Accounts: Are You a Human, Bot, or Cyborg? *IEEE Trans. Dependable Secur. Comput.*, 9(6):811–824, 2012.
8. D. Cook, et al. Twitter Deception and Influence: Issues of Identity, Slacktivism, and Puppetry. *Journal of Information Warfare*, 13(1):58–71, 2014.
9. C. A. Freitas, et al. Reverse Engineering Socialbot Infiltration Strategies in Twitter. *ArXiv e-prints*, 2014.
10. R. G. Garrett. The Chi-square Plot: a Tool for Multivariate Outlier Recognition. *Journal of Geochemical Exploration*, 32(13):319 – 341, 1989.
11. R. Ghosh, et al. Entropy-based Classification of 'Retweeting' Activity on Twitter. In *KDD workshop on Social Network Analysis (SNA-KDD)*, 2011.
12. A. Ghoting, et al. Fast mining of distance-based outliers in high-dimensional datasets. *Data Mining and Knowledge Discovery*, 16(3):349–364, 2008.
13. G. Hazel. Multivariate Gaussian MRF for Multispectral Scene Segmentation and Anomaly Detection. *IEEE Transactions on Geoscience and Remote Sensing*, 38(3):1199–1211, 2000.
14. M. Hubert, et al. ROBPCA: A New Approach to Robust Principal Component Analysis. *Technometrics*, 47:64–79, 2005.
15. M. Hubert, et al. Robust PCA for Skewed Data and its Outlier Map. *Computational Statistics & Data Analysis*, 53(6):2264–2274, 2009.
16. M. Jiang, et al. CatchSync: Catching Synchronized Behavior in Large Directed Graphs. In *KDD*, 941–950. ACM, 2014.
17. M. Jiang, et al. Inferring Strange Behavior from Connectivity Pattern in Social Networks. In *PAKDD*, Tainan, Taiwan, 2014.
18. I. T. Jolliffe. Discarding Variables in a Principal Component Analysis. II: Real Data. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 22(1):21–31, 1973.
19. C. C. Noble et al. Graph-based anomaly detection. In *KDD*, 2003.
20. S. Papadimitriou, et al. LOCI: Fast Outlier Detection Using the Local Correlation Integral. *ICDE 2003*, 2003.
21. N. Shah, et al. Spotting Suspicious Link Behavior with fBox: An Adversarial Perspective. In *ICDM*, 2014.
22. G. Stringhini, et al. Follow the Green: Growth and Dynamics in Twitter Follower Markets. In *IMC*, 163–176. ACM, 2013.
23. G. Tavares et al. Scaling-Laws of Human Broadcast Communication Enable Distinction between Human, Corporate and Robot Twitter Users. *PLoS ONE*, 8(7):e65774, 2013.
24. Twitter Inc. S-1 Filing, US Securities and Exchange Commission. `sec.gov/Archives/edgar/data/1418091/000119312513390321/d564001ds1.htm`, 2013.
25. L. Xiong, et al. Group Anomaly Detection using Flexible Genre Models. In *Advances in Neural Information Processing Systems 24*, 1071–1079. 2011.
26. L. Xiong, et al. Efficient Learning on Point Sets. In *ICDM*, 847–856, 2013.
27. C. Yang, et al. Analyzing Spammers' Social Networks for Fun and Profit: a Case Study of Cyber Criminal Ecosystem on Twitter. In *WWW*, 71–80, 2012.
28. R. Yu, et al. GLAD: Group Anomaly Detection in Social Media Analysis. In *KDD*, 372–381. ACM, 2014.