RSC: mining and modeling temporal activity in social media

ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 21th, 2015, Sydney.
http://www.producao.usp.br/handle/BDPI/49186

RSC: Mining and Modeling Temporal Activity in Social Media

Alceu Ferraz Costa[1], Yuto Yamaguchi[2], Agma Juci Machado Traina[1], Caetano Traina Jr.[1], Christos Faloutsos[3]
[1]Department of Computer Science, University of São Paulo
[2]University of Tsukuba
[3]Department of Computer Science, Carnegie Mellon University
alceufc@icmc.usp.br, yuto_ymgc@kde.cs.tsukuba.ac.jp, agma@icmc.usp.br, caetano@icmc.usp.br, christos@cs.cmu.edu

ABSTRACT

Can we identify patterns of temporal activities caused by human communications in social media? Is it possible to model these patterns and tell if a user is a human or a bot based only on the timing of their postings? Social media services allow users to make postings, generating large datasets of human activity time-stamps. In this paper we analyze time-stamp data from social media services and find that the distribution of postings inter-arrival times (IAT) is characterized by four patterns: (i) positive correlation between consecutive IATs, (ii) heavy tails, (iii) periodic spikes and (iv) bimodal distribution. Based on our findings, we propose Rest-Sleep-and-Comment (RSC), a generative model that is able to match all four discovered patterns. We demonstrate the utility of RSC by showing that it can accurately fit real time-stamp data from Reddit and Twitter. We also show that RSC can be used to spot outliers and detect users with non-human behavior, such as bots. We validate RSC using real data consisting of over 35 million postings from Twitter and Reddit. RSC consistently provides a better fit to real data and clearly outperform existing models for human dynamics. RSC was also able to detect bots with a precision higher than 94%.

Categories and Subject Descriptors

H.2.8 [**Database management**]: Database applications—*Data mining*

General Terms

Algorithms, Experimentation

Keywords

Social Media, Time-Series, User Behavior, Generative Model

1. INTRODUCTION

Given time-stamp data from social media services, can we identify patterns generated by human communication? Is it possible to model these patterns? Can we tell if a user is a human or a bot just by analyzing the timing of their postings? Understanding the dynamics of human activity has attracted the attention of the research community [15, 22, 10, 19, 14, 26] as it has implications that range from efficient resource management [12], human event recognition [16] and clustering [18, 8] to anomaly detection [17, 24, 9]. Previous attempts on modeling human communication have shown that the distribution of inter-arrival times (IAT) often follows heavy-tailed distributions [3]. However, are there other patterns in the IAT distribution that existing models fail to explain? In this paper we aim at answering the following research questions:

- **Q1: Patterns.** What patterns can be discovered from the temporal activities of social media users?
- **Q2: Model.** Can these patterns be modeled?
- **Q3: Usefulness.** Is it possible to use a model of human dynamics to spot anomalies based solely on temporal activity?

To answer these questions, we analyzed posting time-stamp data from two social media services: Reddit and Twitter. Figure 1(a) shows the distribution of consecutive IAT of over 2,000 Reddit users. Figures 1(c) and 1(d) depict the IAT inverse complementary distribution function (CCDF) and log-binned histogram for the same data. The distribution of IAT is characterized by four activity patterns:

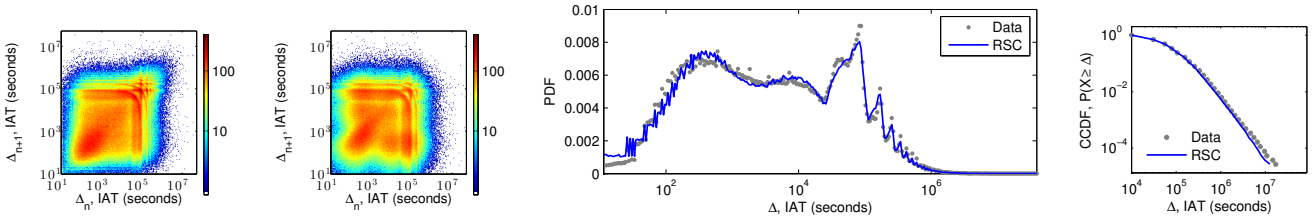
1. **Positive Correlation:** The IAT Δ_i between two postings depends on the previous IAT Δ_{i-1}. This is indicated by a concentration of points in the along the diagonal of Figure 1(a).
2. **Periodic Spikes:** The IAT distribution, shown in the log-binned histogram from Figure 1(c), has spikes at every 24 hours.
3. **Bimodal Distribution:** The IAT distribution has two "humps", the first occurring near 100s and the second occurring near 10,000s.
4. **Heavy-Tailed Distribution:** This pattern is depicted in IAT CCDF plotted in log-log scale in Figure 1(d).

In order to explain all the four discovered patterns we propose RSC, a model that is able to generate synthetic time-stamps that mimics human user activity in social media. In Figures 1(c) and 1(d), RSC is indicated by a solid blue line which accurately matches the real data. We also demonstrate the *usefulness* of RSC by using it to detect bots based only on the time-stamp data from users. The contributions of this paper are:

1. **Pattern Discovery:** We analyze data from social media services and discovered four patterns of user activity: (i) posi-

KDD'15, August 10-13, 2015, Sydney, NSW, Australia.

DOI: http://dx.doi.org/10.1145/2783258.2783294.



(a) Real Data (b) RSC Model (c) IAT Histogram (d) IAT CCDF

Figure 1: Accuracy of our RSC model. (a) Positive correlation between consecutive postings IAT for more than 2,000 Reddit users. (b) Synthetic time-stamps generated by our proposed RSC model. (c) Log-binned histogram showing periodic spikes and bimodal IAT distribution. (d) Heavy-tailed IAT distribution. Our proposed RSC model, indicated by a solid blue line in (b) and (c), is able to match all patterns from the real data.

tive correlation, (ii) heavy-tails, (iii) periodic spikes and (iv) bimodal distribution;

2. **Generative Model:** We propose RSC, a generative model that is able to match all four discovered patterns in user activity;
3. **Bot Detection:** We use RSC to spot outliers and detect users with automated behavior such as bots based only on users' time-stamp data.

We validate RSC using data from over 35 million postings from more than 25,000 Reddit and Twitter users. We show that RSC consistently provided a better match to real data than existing models for human dynamics. We also show that RSC can be used to detect bots with a precision higher than 94%.

In order to allow *reproducibility* of our experiments, we make our RSC code and the datasets used in the experiments available[1]. The outline of the paper starts with the problem definition and goes on to background, pattern mining, model description, experiments, and conclusions.

2. PROBLEM DEFINITION

We are given postings time-stamp data from a set of users $\{\mathcal{U}_1, \mathcal{U}_2, \cdots\}$. Each user $\mathcal{U}_i$ has a sequence of postings time-stamps $T_i = (t_1, t_2, \ldots)$ where $t_i \geq t_{i-1}$. A posting represents an event in which a user submits to the social media service a comment or a tweet. Each posting has a time-stamp t which indicates the time at which the comment or tweet was submitted. From each sequence of time-stamps $T_i = (t_1, t_2, \ldots)$ we are able to compute the postings inter-arrival times (IAT) that we define as follows:

DEFINITION 1 (INTER-ARRIVAL TIME (IAT)). *An IAT, denoted by Δ_i, corresponds to the time interval between two consecutive postings $t_i - t_{i-1}$ from the same user.*

A sequence T_i of time-stamps from a given user $\mathcal{U}_i$ yields a corresponding sequence of IAT $\boldsymbol{\Delta}_i = (\Delta_1, \Delta_2, \Delta_3, \ldots)$. In this paper we are also interested in analyzing *consecutive IAT*, which are defined as:

DEFINITION 2 (CONSECUTIVE IAT). *A consecutive IAT is a pair of two IATs (Δ_i, Δ_{i+1}) from the same user.*

Table 1 lists the symbols and definitions used throughout the paper. As discussed in the Introduction, we want to solve three problems in this paper, which can be stated as follows:

[1]Available at: http://github.com/alceufc/rsc_model

Table 1: Symbols and definitions.

Symbol	Definition
$T_i = (t_1, t_2, \cdots)$	Sequence of time-stamps from user $\mathcal{U}_i$.
$\boldsymbol{\Delta}_i = (\Delta_1, \Delta_2, \cdots)$	Sequence of inter-arrival times (IAT) from user $\mathcal{U}_i$.
θ	Set of RSC parameters.
p_a	Probability of becoming active if resting.
p_r	Probability of resting if active.
p_{post}	Probablity of posting.
t_{wake} and t_{sleep}	Wake-up and sleep time.
f_{sleep}	Fraction of day that user is sleeping.
λ	Base rate of the SCorr.
ρ	Correlation parameter of the SCorr.

PROBLEM 1 (PATTERN-FINDING). *Given the time-stamps data from different social media services, analyze the IAT distribution and find patterns that are common to all services.*

PROBLEM 2 (TIME-STAMP GENERATION). *Design a model that is able to generate synthetic time-stamps whose IAT fits the real data distribution and matches all the patterns found in Problem 1.*

PROBLEM 3 (BOT-DETECTION). *Given time-stamp data from a set of users $\{\mathcal{U}_1, \mathcal{U}_2, \mathcal{U}_3, \cdots\}$ where each user $\mathcal{U}_i$ has a sequence of postings time-stamps $T_i = (t_1, t_2, t_3, \ldots)$ and the corresponding sequence of postings IAT $\boldsymbol{\Delta}_i$, decide if user $\mathcal{U}_i$ is a* human *or a* bot.

3. RELATED WORK

A classical model for human dynamics assumes that arrival times Δ follow a Poisson-Process (PP) [11, 5, 23]. In this case, the IAT are distributed following a memoryless exponential distribution with density function $f(x) = \beta \cdot e^{-x\beta}$, where β corresponds to the mean IAT. Previous works, however, have shown that human communication often has long periods of inactivity followed by bursts of activity, which a Poisson Process is not able to explain [15, 8, 21].

In [3], Barabási proposes a model for human activity that assumes that individuals decide when to perform a task based on a

priority-queue. This model generates an IAT distribution that follows a power law $f(x) = k \cdot x^{-\alpha}$. Other similar approaches that have been shown to generate power-law distributions include [10]. While these models are able to match heavy-tailed IAT distributions, they fail to account for how the daily cycle of user activity affects the distribution of IAT. For example, we show in this paper that the distribution of social media postings IAT is bimodal, which could be indicated bursts of activity followed by resting intervals.

Another approach for modeling human activity consists in using non-homogeneous Poisson Processes, in which the rate $\lambda(t)$ at which the events are generated changes over time [15]. Malmgren et al. propose in [20, 18] a Cascading Non-homogeneous Poisson Process (CNPP) model, based on modeling users' state. The rate changes according to two mechanisms: (i) the model state, (which can be either passive or active) and (ii) time of day and week. We show in Section 6.1 that, the non-homogeneous Poisson Processes from CNPP fails to match the heavy tailed IAT distribution from social media data. We also show that, even though CNPP models daily cycles of activity, the model fails to match daily spikes caused by periods of inactivity, such as sleep time (Figure 9).

The Poisson-Process as well as the power-law and CNPP models assume that consecutive IAT are independent and identically distributed (i.i.d.). However, recent works [14, 26, 13] as well as our collected social media data indicate that consecutive IAT are often correlated. In [26], Vaz de Melo et al. show that human communication violates i.i.d. assumption and propose a model named Self-Feeding Process (SFP) in which a synthetic IAT Δ_i is generated as a function of the previous IAT.

Table 2 shows the four activity patterns that we discovered from social media time-stamp data and compares the capabilities of existing models as well as the RSC model that we propose in this paper. Only RSC is able to match all the patterns.

Table 2: Communication patterns matched by different models. Only RSC is able to describe all patterns.

Pattern	Poisson-Process	Log-Logistic	Power-Law	CNPP	SFP	**RSC**
Heavy Tails		✓	✓		✓	✓
Bimodal				✓		✓
Spikes						✓
IAT Correlation					✓	✓

The study of outlier detection is also relevant to our work, since we use our proposed RSC model to detect bots. Many outlier detection methods have been proposed in the literature [1, 4, 25]. In the context of using time-stamp data to spot bots, in [28] a method is proposed that consists in constructing a scatter-plot of the minute vs. the second for all comment time-stamps of a user. The plot is then used to visually assess whether a user is a bot or not. However the plot is not used to automatically test whether users are bot or not.

In [6] the authors use the entropy of the histogram of IAT as a feature to detect Twitter bots. Their method differs from ours because we show that it is possible to detect bots using only time-stamp data while [6] combine time-stamp features with textual features as well as account properties such as number of friends.

4. USER ACTIVITY: PATTERNS

In this section we analyze time-stamp data generated by user communication on Twitter and Reddit. We report our findings and show that existing models are not able to match all the users postings patterns.

4.1 Datasets

We collected postings time-stamp data from Reddit and Twitter users. For the Twitter dataset we collected data from over 9,000 verified accounts. Verified accounts are manually verified to be authentic by Twitter. For each user we collected the 3,000 most recent postings. Users with less than 800 postings were discarded, resulting in 6,790 users.

For Reddit, we randomly selected a set of over 200,000 users that commented on Reddit at least once between December 6th and December 29th 2013. Due to Reddit API restrictions, we collected the 1,000 most recent comments for each user. Similarly to the Twitter data, users with less than 800 comments were discarded, resulting in set of 21,198 users.

In Section 6.2 we test our proposed method for the task of bot detection. For that purpose, we inserted bot users in both the Twitter and Reddit dataset. For Reddit, we searched the Web using queries such as "bots users reddit" to collect a list of suspicious accounts. We inspected each suspicious account by checking the content of the messages and manually selected 32 bots. For the Twitter dataset, we used the lists features in which individuals can create lists of Twitter accounts. Using the Twitter API, we searched for suspected users that were in lists that contained the term "bot" in its description or title. We manually inspected the suspected accounts recent tweets and manually selected 64 bots.

Table 3 summarizes the collected data. For both datasets, the comments and tweets time-stamps have a resolution of one second.

Table 3: Summary of the datasets.

Dataset	# Users	# Bots	# Time-stamps
Reddit	21,198	32	20 Million
Twitter	6,790	64	16 Million

4.2 Temporal Patterns in Social Media Communication

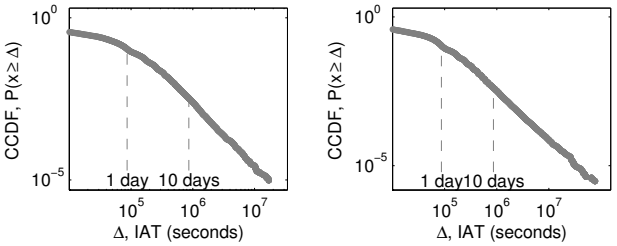
In this Section we analyze the postings IAT using data from Reddit and Twitter. Our goal is to find patterns that are common to both services and that can be used to model user activity in social media. We start our analysis plotting in Figure 2 the complementary cumulative distribution function (CCDF) of the users' postings IAT.

OBSERVATION 1. *The distribution of the postings' IAT is heavy-tailed.*

This heavy-tail pattern agrees with previous studies in human activity. This also shows that classical Poisson statistics are not adequate to model the interval between users' postings. A consequence of the heavy-tail pattern is that a regular user can be inactive for long periods of time.

In this paper we show that it is possible to detect whether a user is a human or a bot based on time-stamp data (Section 6.2). A way that a bot could use to evade a bot-detection scheme would be

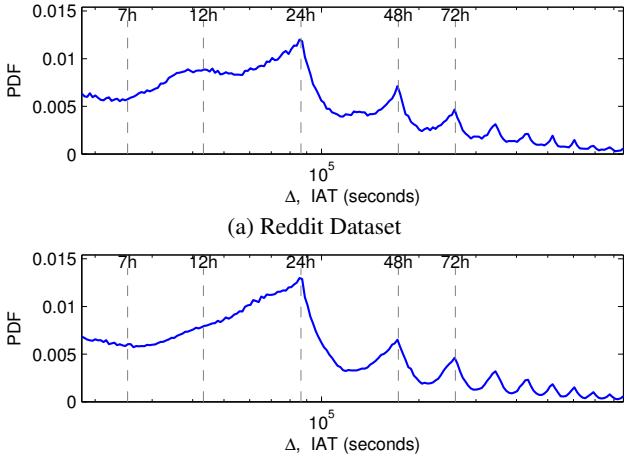
to mimic human behavior. However, as consequence of the heavy tail pattern, in order to mimic human behavior a bot would need to reduce its posting rate, potentially reducing volume of content generated by a single bot.



(a) Reddit (b) Twitter

Figure 2: Heavy-tailed distribution of postings IAT for the (a) Reddit and (b) Twitter datasets.

The circadian rhythm also affects the users' communication patterns. Figure 3 also shows the histogram of the postings IAT by zooming into the interval from five hours to ten days. The histogram for both Reddit and Twitter has periodic peaks at every 24 hour intervals.



(a) Reddit Dataset

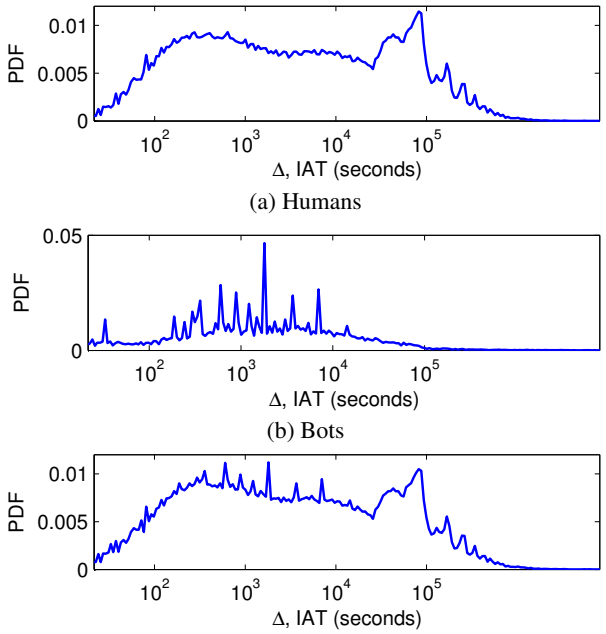
(b) Twitter Dataset

Figure 3: Periodic peak pattern: the log-scale histogram of postings' IAT has peaks at 24 hour intervals.

OBSERVATION 2. *The distribution of IAT has periodic spikes at every 24 hours.*

To the best of our knowledge, this is the first work to find this pattern of human activity. Also, existing models for human communication dynamics are not able to explain the emergence of daily peaks in the distribution of postings IAT. In Section 5, by using our proposed RSC model, we show that daily peaks in the distribution of IAT can be attributed to daily sleeping intervals in which users stop communicating.

Non-organic behavior often generates peaks in the distribution of IAT. The periodic spikes pattern is important, as differentiating between which peaks are caused by human behavior and which caused by automated behavior has implications in bot-detection. We illustrate this point by showing in Figure 4(a) the IAT distribution for 200 Reddit human users and in Figure 4(b) the IAT distribution for 28 Reddit bots. Figure 4(c) shows the combined IAT distribution with data from humans and bots. Notice that only humans have the periodic spikes, while bots have many spikes for IAT smaller than 10,000s.



(a) Humans

(b) Bots

(c) Humans and Bots

Figure 4: Peaks caused by human and bot behavior are different. (a) Periodic peak pattern caused by humans. (b) Peaks caused by bots. (c) IAT histogram with data from (a) and (b).

We also analyze the distribution function of the postings IAT. Figure 3 shows the histogram of the postings' IAT for all users in the Reddit and Twitter datasets in logarithmic scale. For both datasets, the histogram has two modes.

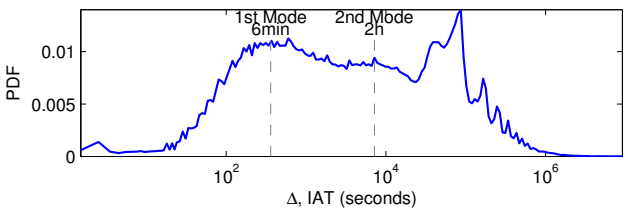
OBSERVATION 3. *Excluding the daily periodicities, the distribution of postings' IAT is bimodal.*

We show in Section 5 that the bimodal distribution can be explained by users having highly active sessions separated by rest intervals. Evidence of bimodal IAT distribution in human activity has been reported before in vehicle traffic [12] as well as in users exchanging SMS [27].

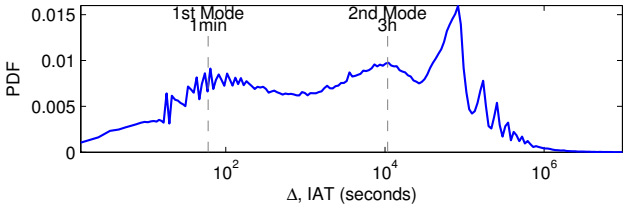
We also show that the analysis of the how consecutive postings' IAT are correlated can provide information about users' communication patterns. For that purpose we propose DELAY-MAP, a visualization that consists in plotting pairs of consecutive IAT (Δ_n, Δ_{n+1}). To avoid occlusion, DELAY-MAP divides the Δ_n vs. Δ_{n+1} space in a log spaced grid and count the pairs of consecutive IAT in each cell. Finally, DELAY-MAP uses color coding to represent the grid cell counts, resulting in a heat-map visualization. Figure 6 shows the DELAY-MAP for the Reddit and Twitter datasets.

OBSERVATION 4. *Consecutive IATs are positively correlated.*

There is a concentration of consecutive IAT along the diagonal of the DELAY-MAP for both the Reddit and Twitter datasets. This indicates that there is a positive correlation between consecutive IAT

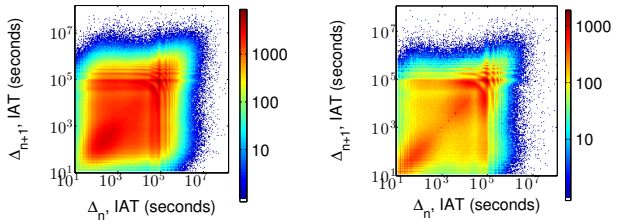


(a) Reddit Dataset



(b) Twitter Dataset

Figure 5: Bimodal IAT distribution: the log-scale histogram of postings IAT has two modes.



(a) Reddit (b) Twitter

Figure 6: Consecutive IAT correlation pattern. The DELAY-MAP of postings IAT is concentrated along the diagonal $\Delta_n = \Delta_{n+1}$.

and that the distribution of IAT is not independent and identically distributed (i.i.d). Another pattern presented by the DELAY-MAP are crests for $\Delta_n \approx 1$ day or $\Delta_{n+1} \approx 1$ day. The crests match the periodic spikes at 24 hour intervals (Observation 2) from the PDF of postings' IAT (Figure 3).

5. Rest-Sleep-and-Comment

Can we describe how long it will take for a user to make a posting on a social media service such as Twitter and Reddit? Can we model all the patterns we found in real data in Section 4? Based on observations from real data, we propose Rest-Sleep-and-Comment (RSC), a generative model that is able to explain the following user activity patterns:

1. Heavy tailed IAT distribution (Observation 1);
2. Bimodal IAT distribution (Observation 3);
3. Periodic Spikes in the IAT distribution, centered at 24 hour intervals (Observation 2);
4. Correlation between consecutive IAT (Observation 4);

We start in Section 5.1 by proposing the Self-Correlated Process (SCorr), a stochastic process that is able to generate events whose IAT are correlated. Section 5.2 describes the algorithm used by RSC to generate synthetic time-stamps and IAT. Finally, in Section 5.3, we describe the RSC parameter estimation algorithm.

5.1 The Self-Correlated Process

In this Section we propose the Self-Correlated Process (SCorr), a stochastic process that generates a sequence of synthetic IAT. Differently from a Poisson Process or models based on priority queues, consecutive IAT generated by SCorr are correlated. The motivation is to match the correlation pattern from real data (Observation 4). SCorr is defined as follows:

DEFINITION 3. *Let δ_i be the inter-arrival time between the events i and $i-1$. A stochastic process is a Self-Correlated Process, with base rate λ and correlation ρ if:*

$$\delta_1 \sim Exp\left(\frac{1}{\lambda}\right) \tag{1}$$

$$\delta_i \sim Exp\left(\rho \cdot \delta_{i-1} + \frac{1}{\lambda}\right) \tag{2}$$

where $X \sim Exp(1/\lambda)$ denotes an exponentially distributed random variable with rate λ.

In SCorr, the duration δ_i between two events is sampled from an exponential distribution with rate λ depending on the previous inter-event time δ_{i-1}. SCorr uses the *correlation parameter* ρ to control the dependency between consecutive inter-event times. The Poisson-Process and the Self-Feeding Process are special cases of the SCorr:

THEOREM 1 (SCORR EQUIVALENCE). *When $\rho \to 0$, SCorr reduces to a Poisson Process with rate λ. When $\rho = 1$, SCorr reduces to a Self-Feeding Process.*

PROOF. For $\rho \to 0$, Equation 2 yields $\delta_i \sim \text{Exp}\left(\frac{1}{\lambda}\right)$ which corresponds to the IAT distribution of a Poisson-Process. For $\rho = 1$, Equation 2 results in $\delta_i \sim \text{Exp}\left(\delta_{i-1} + \frac{1}{\lambda}\right)$, which corresponds to the definition of the Self-Feeding Process. □

5.2 Time-stamp Generation

In order to model the bimodal IAT distribution from real data (Observation 3) and the periodic spikes (Observation 2), the RSC algorithm can be in one of the following states:

1. **Active:** While RSC remains in the active state, it generates postings events with a probability p_{post} or null events with probability $1 - p_{\text{post}}$ at every time interval δ_i^A. The intervals δ_i^A and δ_{i+1}^A are correlated and generated using SCorr.
2. **Rest:** While RSC is in the rest state, it generates null events at every time interval δ^R. As result, the RSC rest state only contributes to increment the postings inter-arrival times.
3. **Sleep:** When RSC enters the sleep state, it generates a single null event after a time interval δ^S. The interval δ^S corresponds to the time necessary to advance the clock-time t_{clock} until the next wake-up time t_{wake}.

In RSC, postings events are only generated during the active state. In order to match the correlation between consecutive postings' IAT (Observation 4), the interval δ_i^A between active state events is dependent on the previous interval δ_{i-1}^A. In order to model this property we use our proposed SCorr to generate the active state intervals:

$$\delta_i^A \sim \text{Exp}\left(\frac{1}{\lambda} = \left[\rho_A \cdot \delta_{i-1}^A + \frac{1}{\lambda_A}\right]\right) \tag{3}$$

where λ_A and ρ_A control the mean active state events intervals and correlation between consecutive state intervals for the active state.

The intervals δ^R between consecutive rest state null events is also generated by a Self-Correlated Process:

$$\delta_i^R \sim \text{Exp}\left(\frac{1}{\lambda} = \left[\rho_R \cdot \delta_{i-1}^R + \frac{1}{\lambda_R}\right]\right) \quad (4)$$

RSC assumes that the mean interval in the rest state is larger than the mean interval of the active state, that is $\lambda_R < \lambda_A$. As we show in Section 6.1, each SCorr generates a hump in the IAT distribution. By using a mixture of SCorr, RSC is able to match the bimodal pattern.

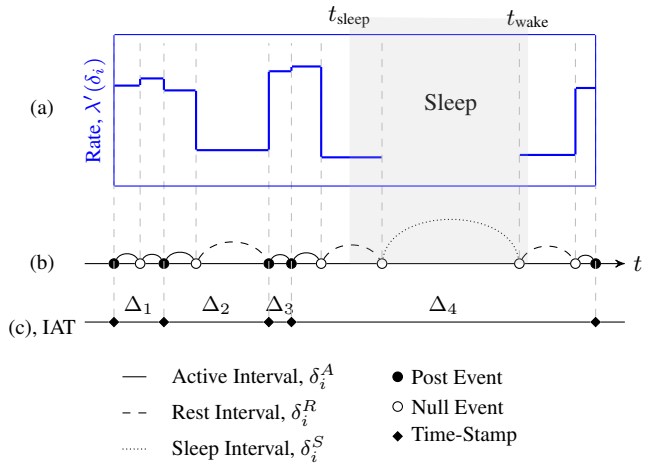


Figure 7: The RSC model. (a) The rate at which RSC generates events changes over time as a function of the previous inter-event time. (b) RSC generates null and postings events depending on its current state: active, rest or sleep. (c) Synthetic IAT Δ_i are generated at each posting event.

In order to compute the time-interval δ^S for the sleep state, RSC keeps track of the current time of day by using a clock t_{clock} variable, where 0:00h $< t_{\text{clock}} <$ 23:59h. The clock variable t_{clock} is advanced after each generated state time-interval. For example, when δ^R or δ^A intervals are generated in the rest or active states, the clock variable is advanced, respectively, by δ^R or δ^A. The interval duration δ^S for the sleep state is generated in order to advance the internal clock until the next wake-up time t_{wake}:

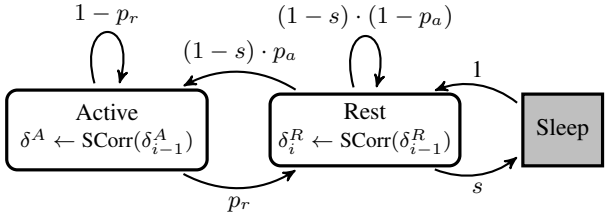
$$\delta^S = \begin{cases} t_{\text{wake}} - t_{\text{clock}} & \text{if } t_{\text{clock}} < t_{\text{wake}}, \\ t_{\text{wake}} + (24\text{h} - t_{\text{clock}}) & \text{otherwise.} \end{cases} \quad (5)$$

By assuming that $t_{\text{wake}} = 0$, we can replace the parameters t_{wake} and t_{sleep} by a single parameter f_{sleep} that corresponds to the fraction of the day that is considered sleep-time by RSC. In this case, $t_{\text{sleep}} = f_{\text{sleep}} \cdot 24\text{h}$ when $t_{\text{wake}} = 0$.

Figure 7(b) illustrates the null events and postings events generated by RSC over time, which are indicated by white and black dots, respectively. The arcs between events indicate the current RSC state: active, rest or sleep. Figure 7(c) shows the generated synthetic time-stamps and IAT Δ_i, which occur at each posting event when RSC is in the active state. Figure 7 shows how the rate at which RSC generates posting and null events changes over time as a function of the previous inter-event interval, as described by Equations 3 and 4.

Figure 8 shows the state diagram for the RSC model. If the current clock time t_{clock} falls within the sleep time, that is, $t_{\text{wake}} > t_{\text{clock}}$ and $t_{\text{sleep}} < t_{\text{clock}}$, then the user will always transition to sleep state if the current state is rest. If the current clock time does not fall within the sleep time, then the transition is given by the following probabilities:

- If a user is active, there is a probability p_r that she will rest and a probability $1 - p_r$ that she will remain active.
- If a user is resting, there is a probability p_a that she will become active and a probability $1 - p_a$ to remain resting.
- After the sleep state ends, the user will always transition to rest state.



$$s = \begin{cases} 1, \text{ if } t_{\text{wake}} < t_{\text{clock}} < t_{\text{sleep}}, \\ 0, \text{ otherwise.} \end{cases}$$

Figure 8: State diagram for the RSC model. Each edge indicates the probability of going from one state to another. The duration δ^R and δ^A of the rest and active states are generated using the Self-Correlated Process. The transition from the rest to the sleep state (indicated by a clock) occurs based on the current clock time.

When users transition between rest and active states, RSC sets the previous state duration variables to $\delta_{i-1}^A = 0$ and $\delta_{i-1}^R = 0$. Users submit postings with a probability p_{post} at the end of each active state. As result, there can be more than one state transition between two comments.

Algorithm 1 describes the procedure to generate time-stamps using the RSC model. The procedure RAND generates a uniformly distributed number in the interval $[0, 1]$. The procedure TIMEUNTILWAKEUP generates the sleep state event interval and corresponds to Equation 5. Finally, the procedure ISSLEEPING computes the current t_{clock} value and checks whether RSC should transition from the rest state to the sleep state.

5.3 Parameter Estimation

In order to estimate the parameters of RSC we propose an algorithm based on fitting the histogram of the observed data IAT. The algorithm starts by generating synthetic time-stamps using the RSC model. The next step consists in computing a log-binned histogram of IAT for real and synthetic data. The width w_i of the i-th bin is wider than the previous $(i-1)$-th bin by a fixed factor k. That is, $w_i = w_{i-1} \cdot k$. We denote the counts of IAT in each i-th bin for the real and synthetic data as c_i and $\hat{c}_i$, respectively.

Using the *Levenberg-Marquardt* algorithm, we find the parameter values θ that minimize the squared difference between synthetic and real data bin counts:

$$\min_{\theta} \sum_i \left(c_i - \hat{c}_i(\theta)\right)^2 \quad (6)$$

Using logarithm binning when approximating the PDF allows the parameter estimation method to match the heavy tail pattern as well as the daily spikes found in real data.

Algorithm 1: Algorithm to generate time-stamps using the RSC model.

Input : Parameters $\theta = \{p_r, p_a, p_{\text{post}}, \lambda_A, \rho_A, \lambda_R, \rho_R, f_{\text{sleep}}\}$ and size N of the time-stamp sequence.
Output: Sequence of time-stamps $T = [t_1, t_2, \ldots]$.

currentState $\leftarrow$ Active, $\Delta \leftarrow 0$, $i \leftarrow 1$, $T \leftarrow [0]$;
$t_{\text{wake}} \leftarrow 0$, $t_{\text{sleep}} \leftarrow 24\text{h} \cdot (1 - f_{\text{sleep}})$;
while $i \leq N$ **do**
  if currentState = *Active* **then**
    $\Delta \leftarrow \Delta + \text{SCorr}(\lambda_A, \rho_A, \delta^A_{i-1})$;
    if $p_{post} >$ RAND() **then**
      $i \leftarrow i + 1$, $T[i] \leftarrow T[i-1] + \Delta$, $\Delta \leftarrow 0$;
    if RAND() $< p_r$ **then**
      currentState $\leftarrow$ Rest, $\delta^R_{i-1} \leftarrow 0$;
    else
      currentState $\leftarrow$ Active;
  else if currentState = *Rest* **then**
    $\Delta \leftarrow \Delta + \text{SCorr}(\lambda_R, \rho_R, \delta^R_{i-1})$;
    if IsSleeping(T, Δ, t_{wake}, t_{sleep}) **then**
      currentState $\leftarrow$ Sleep;
    else if RAND() $< p_a$ **then**
      currentState $\leftarrow$ Active, $\delta^A_{i-1} \leftarrow 0$;
    else
      currentState $\leftarrow$ Rest;
  else if currentState = *Sleep* **then**
    $\Delta \leftarrow \Delta +$ TimeUntilWakeUp(T, t_{wake} , t_{sleep});
    currentState $\leftarrow$ Rest;

6. RSC AT WORK

In this Section we show that RSC is able to accurately fit real data and demonstrate its usefulness by using it to detect bots based solely on time-stamp data from users.

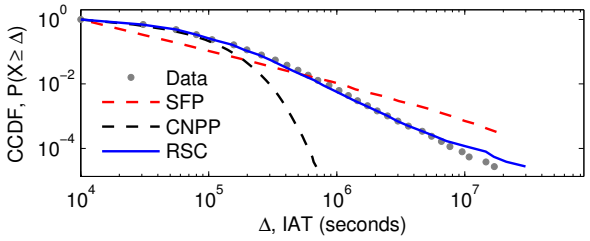
6.1 Simulations

In this Section we analyze how well RSC is able to match real data from social media services. We compare RSC against two other models: the cascading non-homogeneous Poisson Process (CNPP) proposed in [18] and the Self-Feeding Process proposed in [26]. We estimate the parameters of all models using the algorithm described in Section 5.3.

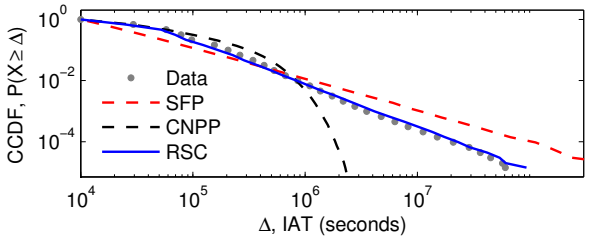
We start by comparing the IAT distribution of real and synthetic data. Figure 9 shows the log-scale histogram of IAT for all users in the Reddit and Twitter datasets. We compare the real data distribution (gray dots) to IAT generated by RSC and competitors: SFP (Self-Feeding Process) and CNPP (Cascading Non-homogeneous Poisson-Process). Our RSC model, indicated by a solid blue line, accurately matches the real data. By modeling the sleep vs. awake cycle, RSC is able to explain the spikes centered at every one day interval which the other models fail to explain. Moreover, RSC is able to explain the bimodal distribution of IAT.

Figure 10 compares the IAT CCDF of real and synthetic time-stamps. Our RSC model accurately matches the heavy tailed IAT distribution. The CNPP model, based on a non-homogeneous Poisson-Process fails to match the heavy tail. Even though SFP is able to generate a heavy tailed distribution, it fails to fit the slope of the data.

We also evaluate how well RSC is able to match the correlation between consecutive IAT. Figure 11 uses the Delay-Map visualization to compare the distribution of consecutive IAT from real synthetic time-stamps. RSC provides the best match to real



(a) Reddit



(b) Twitter

Figure 10: RSC (solid blue line) matches the heavy tailed distribution of the real data (gray dots). CCDF of postings IAT for the (a) Reddit and (b) Twitter datasets.

data when compared to competitors. By using our proposed Self-Correlated Process, RSC is able to generate correlated consecutive IAT, that can be seen by a concentration of points along the diagonal of Figures 11(b) and 11(f). Additionally, RSC is able match the crests centered at $\Delta(n) \approx 1$ day or $\Delta(n+1) \approx 1$ day. The CNPP model, shown in Figures 11(c) and 11(g), fails to match both the crests and correlation of consecutive IAT. Finally, the SFP model, shown in Figures 11(d) and 11(h) generates a correlation between consecutive IAT that is too strong, significantly deviating from the real data.

6.2 Bot Detection

Is it possible to tell if users are bots or humans just by analyzing the time-stamps of their postings? In the problem that we want to solve, we are given time-stamp data from a set of users $\{\mathcal{U}_1, \mathcal{U}_2, \cdots\}$ where each user $\mathcal{U}_i$ has a sequence of postings time-stamps $T_i = (t_1, t_2, t_3, \ldots)$ and the corresponding sequence of comments inter-arrival times $\boldsymbol{\Delta}_i = (\Delta_1, \Delta_2, \Delta_3, \ldots)$. Our goal is to decide whether user $\mathcal{U}_i$ is a *human* or a *bot*.

In this Section we propose RSC-Spotter, a method for bot detection that uses RSC to solve the bot-detection problem.

6.2.1 RSC-Spotter

RSC-Spotter compares the distribution of IAT of each user to the aggregated distribution of IAT of all users in the dataset. Users whose IAT distribution are significantly different from the aggregated IAT distribution are flagged as outliers and potential bots. RSC-Spotter uses RSC to compare users' IAT distributions to the aggregated IAT distribution. First, RSC-Spotter estimates RSC parameters using the aggregated IAT data, and then, for each user, generates n_i time-stamps, where n_i is *exactly the number of time-stamps for user* $\mathcal{U}_i$. Finally, RSC-Spotter computes the dissimilarity between the synthetic time-stamps distribution the users' time-stamps. The RSC-Spotter algorithm can be summarized as follows:

1. Using the RSC parameter estimation algorithm proposed in

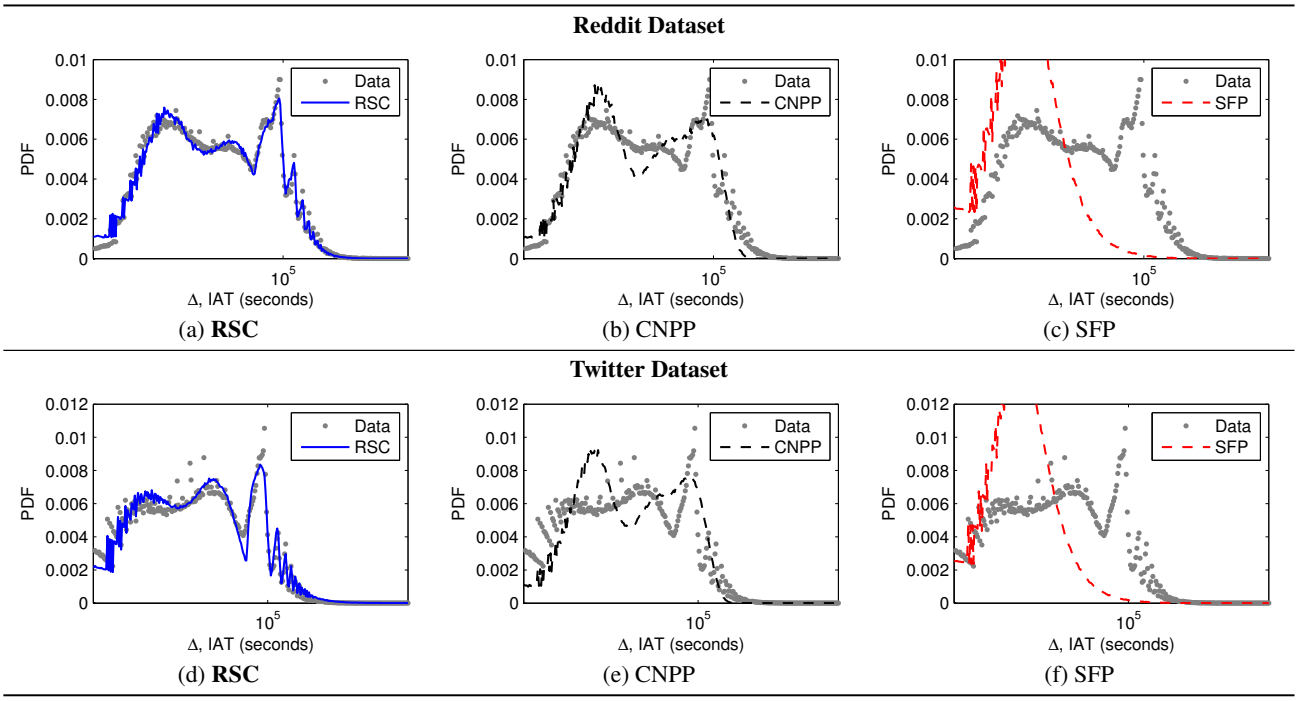


Figure 9: RSC (solid blue line) matches both the periodic spikes and bimodal IAT distribution of the real data.

Section 5.3, estimate the parameters $\theta_\cup$ for the set of IAT $\mathbf{\Delta}_\cup$, where $\mathbf{\Delta}_\cup = \bigcup_i \mathbf{\Delta}_i$ is the union of all users' sequences of IAT $\mathbf{\Delta}_i$.

2. Compute the counts c_j of the log-binned histogram of the user $\mathcal{U}_i$ postings IAT, where $\sum_j c_j = 1$.
3. Generate synthetic n_i synthetic IAT using RSC and the estimated parameters $\theta_\cup$, where n_i *is the number of IAT for user* $\mathcal{U}_i$.
4. Compute the counts $\hat{c}_j$ of the log-binned histogram of synthetic IAT, where $\sum_j \hat{c}_j = 1$.
5. Compute the dissimilarity $\mathcal{D}_i$ from user $\mathcal{U}_i$ to the RSC model as:

$$\mathcal{D}_i = \sum_j |c_j - \hat{c}_j| \tag{7}$$

The next step consists in deciding whether the dissimilarity value $\mathcal{D}_i$ indicates that user $\mathcal{U}_i$ is a human or a bot. Given a training set of users labeled either as bots (positive examples) or humans (negative examples), we train a Naive-Bayes classifier to estimate the posterior probability p_{bot} that a user is a bot. The dissimilarity values $\mathcal{D}_i$ are used as features for the classifier. If p_{bot} is higher than a decision threshold p_{thresh}, then the user is classified as a bot.

Based on [7], we estimate the decision threshold p_{thresh} by assigning a cost c_{FN} to false negative (FN) errors and a cost c_{FP} to false positive (FP) errors. A false negative error occurs when bot is classified as a human. A false positive error occurs when a human is classified as a bot. We select the threshold p_{thresh} that minimizes the F_β-Measure on the training set:

$$F_\beta = \frac{(1+\beta^2)\cdot \text{TP}}{(1+\beta^2)\cdot \text{TP} + \beta^2 \cdot \text{FN} + \text{FP}} \tag{8}$$

where:

$$\beta = \sqrt{\frac{c_{\text{FN}}}{c_{\text{FP}}}}. \tag{9}$$

6.2.2 RSC-SPOTTER *Evaluation*

We evaluated RSC-SPOTTER using a sample of 2,000 Reddit users composed of 37 bots and 1,963 humans. We also selected 1,353 Twitter users that were identified by Twitter as humans (e.g. music celebrities and politicians) and also a set of 64 bots. We start by showing that the dissimilarity $\mathcal{D}_i$ can be used to separate humans and bots. Figure 12 shows the kernel smoothing function estimate of $\mathcal{D}_i$ values for bots and humans. The dissimilarity values for humans are significantly lower than the dissimilarity values for bots, generating two clusters.

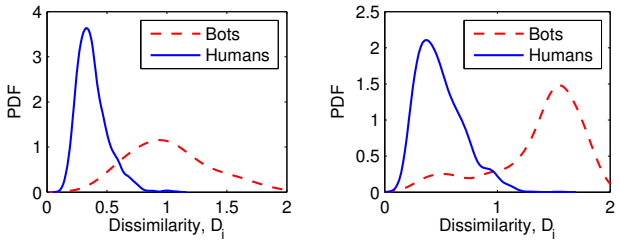
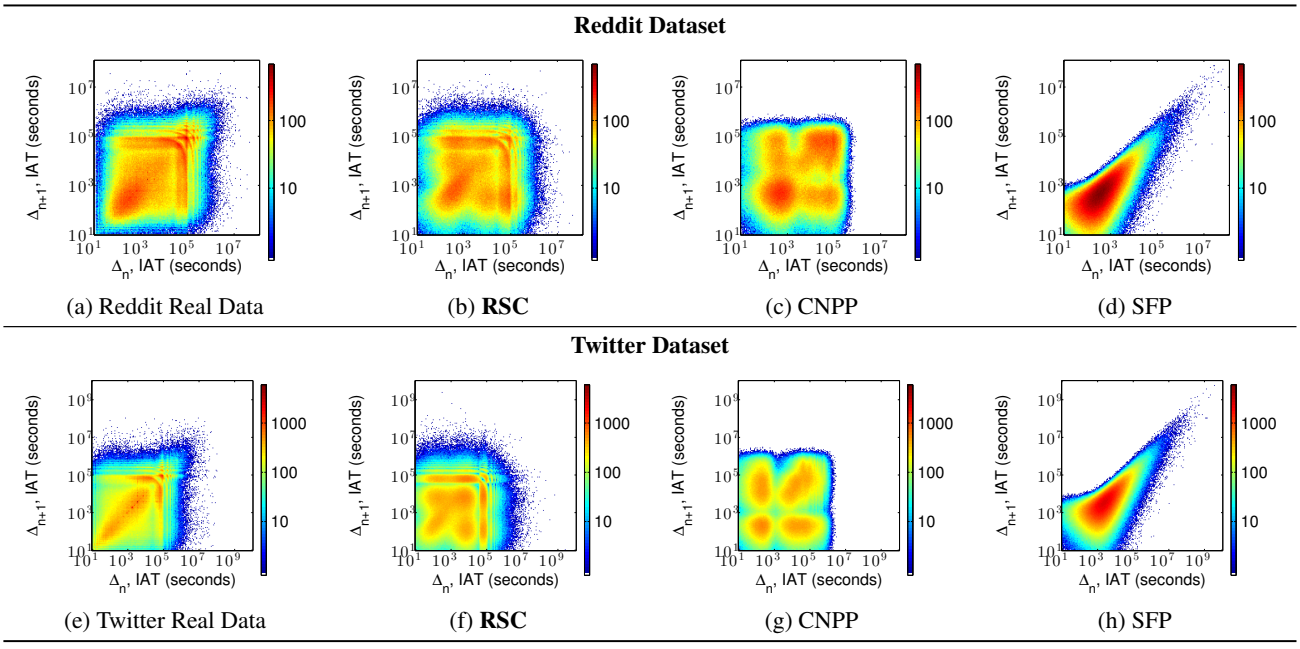


Figure 12: RSC-SPOTTER is able to separate bots and humans. Dissimilarity values $\mathcal{D}_i$ for users labeled as bots and humans.

To classify the users as humans or bots, we randomly split the datasets into train and test subsets of the same size while preserving the class distribution. The train subset is used to train the Naive-Bayes classifier, estimate p_{thresh} and the RSC parameters $\theta_\cup$. Each experiment was repeated ten times. We compare RSC-SPOTTER against the following features extracted from the sequence of timestamps of each user:



(a) Reddit Real Data (b) **RSC** (c) CNPP (d) SFP

(e) Twitter Real Data (f) **RSC** (g) CNPP (h) SFP

Figure 11: RSC– (b) and (f) – matches the IAT correlation pattern, providing the best match for the distribution of consecutive IAT. Each heat-map depicts the distribution of consecutive IAT.

1. *IAT Histogram:* Log-binned histogram of postings IAT;
2. *Entropy:* Entropy of the IAT histogram, proposed in [6];
3. *Weekday Histogram:* Histogram with seven bins, where each bin counts the number of postings for each day of the week;
4. *All Features:* Combination of all features from 1, 2, and 3.

Figure 13 compares the precision vs. sensitivity (recall) curve [2] obtained by RSC-SPOTTER and competitors. A good performance is indicated by a curve closer to the top part of the plot. For the Twitter dataset (Figure 13(b)), RSC-SPOTTER obtained the highest precision for all sensitivity values, indicating that for any configuration of FP and FN costs, it is better than the other features. For the Reddit dataset (Figure 13(a)), RSC-SPOTTER obtained considerably higher precision for sensitivity values smaller than 70%, and precision values closer to the other methods for sensitivity values larger than 70%.

Even though the datasets are strongly imbalanced, with significantly more humans than bots, RSC-SPOTTER obtained a precision of with 96.5% and 94.7% for sensitivity values of 47.9% and 70.3% for the Reddit and Twitter datasets, respectively.

7. CONCLUSIONS

In this paper we analyzed time-stamp of over 35 millions users postings form two social media services: Reddit and Twitter. We found that the IAT distribution has four activity patterns:

1. **Positive Correlation:** There is a dependency between consecutive IAT (Observation 4).
2. **Periodic Spikes:** The circadian rhythm affects the users' postings times, generating period peaks in the IAT distribution at every 24 hours (Observation 2).
3. **Bimodal Distribution:** The IAT distribution has two modes (Observation 3).
4. **Heavy-Tailed Distribution:** The distribution of IAT is heavy-tailed, indicating that users can be inactive for longs period of time before making a posting (Observation 1).

We also proposed RSC, a generative model that describes the IAT between users' postings in social media. We compared RSC against representative models for human activity from the literature. We show that only RSC was able to match all four activity patterns. Moreover, RSC was able to provide an accurate fit to the real data IAT distribution.

Finally, we also show that RSC can be used to spot bots with automated behavior in social media. We proposed RSC-SPOTTER, a method that uses RSC to tell if users are humans or bots based solely on the timing of their postings with a precision higher than 94%. The contributions of this paper can be summarized as follows:

1. **Pattern Discovery:** We analyzed the time-stamps from users' postings and discovered four activity patterns: positive correlation, heavy-tails, periodic spikes and bimodal distribution;
2. **Generative Model:** We proposed RSC, a model that is able accurately to match the distribution of postings IAT from social media services;
3. **Bot-Detection:** we proposed RSC-SPOTTER, a method that users RSC to detect bots with a precision higher than 94%.

Acknowledgments: This material is based upon work supported by FAPESP, CNPq, CAPES, STIC-AmSud, the RESCUER project funded by the European Commission (Grant: 614154) and by the CNPq/MCTI (Grant: 490084/2013-3), JSPS KAKENHI, Grant-in-Aid for JSPS Fellows #242322, the National Science Foundation under Grant No. CNS-1314632, IIS-1408924, ARO/DARPA under Contract Number W911NF-11-C-0088 and by the Army Research Laboratory under Cooperative Agreement Number W911NF-09-2-0053.

Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation, or other funding parties. The U.S. Government is authorized to reproduce

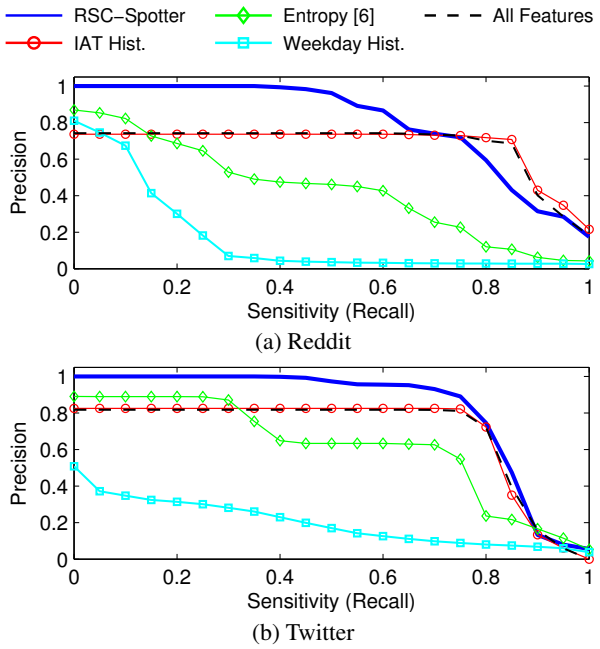


Figure 13: RSC-SPOTTER wins or ties with the best competitor. A good performance is indicated by a curve closer to the top part of the plot.

and distribute reprints for Government purposes notwithstanding any copyright notation here on.

8. REFERENCES

[1] C. C. Aggarwal and P. S. Yu. Outlier detection for high dimensional data. *SIGMOD*, pages 37–46, 2001.

[2] R. Baeza-Yates and B. Ribeiro-Neto. *Modern information retrieval*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2nd edition, 1999.

[3] A. L. Barabasi. The origin of bursts and heavy tails in human dynamics. *Nature*, 435(7039):207–211, 2005.

[4] V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection: a survey. *ACM Computing Surveys*, 43(3):1–72, 2009.

[5] J. Cho and H. Garcia-Molina. Estimating frequency of change. *ACM Transactions on Internet Technology*, 3:256–290, 2003.

[6] Z. Chu, S. Gianvecchio, and H. Wang. Who is Tweeting on Twitter: Human, Bot, or Cyborg? In *ACSAC*, pages 21–30, 2010.

[7] C. Drummond and R. C. Holte. Cost curves: An improved method for visualizing classifier performance. *Machine Learning*, 65:95–130, 2006.

[8] J.-P. Eckmann, E. Moses, and D. Sergi. Entropy of dialogues creates coherent structures in e-mail traffic. *PNAS*, 101(7):14333–14337, 2004.

[9] S. Günnemann, N. Günnemann, and C. Faloutsos. Detecting Anomalies in Dynamic Rating Data: a Robust Probabilistic Model for Rating Evolution. In *KDD*, pages 841–850, 2014.

[10] C. A. Hidalgo R. Conditions for the emergence of scaling in the inter-event time of uncorrelated and seasonal systems. *Physica A*, 369(2):877–883, Sept. 2006.

[11] P. G. Hoel, S. C. Port, and C. J. Stone. *Introduction to Stochastic Processes*. Waveland Pr. Inc., 1986.

[12] A. Ihler, J. Hutchins, and P. Smyth. Adaptive event detection with time-varying poisson processes. In *KDD*, pages 207–216, 2006.

[13] D. Juan, L. Li, H. Peng, D. Marculescu, and C. Faloutsos. Beyond Poisson : Modeling Inter-Arrival Time of Requests in a Datacenter. In *PAKDD*, volume 1, pages 198–209, 2014.

[14] M. Karsai, K. Kaski, A. Barabási, and J. Kertész. Universal features of correlated bursty behaviour. *Scientific Reports*, 2:1–7, 2012.

[15] J. Kleinberg. Bursty and Hierarchical Structure in Streams. In *KDD*, pages 373–397, 2003.

[16] N. C. Krishnan and D. J. Cook. Activity recognition on streaming sensor data. *Pervasive and Mobile Computing*, 10:138–154, 2014.

[17] T. Lappas, M. R. Vieira, D. Gunopulos, and V. J. Tsotras. On the spatiotemporal burstiness of terms. In *VLDB*, pages 836–847, 2012.

[18] R. D. Malmgren, J. M. Hofman, L. A. N. Amaral, and D. J. Watts. Characterizing Individual Communication Patterns. In *KDD*, pages 607–616, 2009.

[19] R. D. Malmgren, D. B. Stouffer, A. S. L. O. Campanharo, and L. A. N. Amaral. On universality in human correspondence activity. *Science*, 325:1696–1700, 2009.

[20] R. D. Malmgren, D. B. Stouffer, A. E. Motter, and L. A. N. Amaral. A Poissonian explanation for heavy tails in e-mail communication. *PNAS*, 105(47):18153–18158, 2008.

[21] J. G. Oliveira and A. Barabási. Human dynamics: Darwin and Einstein correspondence patterns. *Nature*, 437:1251, 2005.

[22] R. Ottoni, D. L. Casas, J. P. Pesce, W. Meira Jr., C. Wilson, A. Mislove, and V. Almeida. Of Pins and Tweets: Investigating How Users Behave Across Image-and Text-Based Social Networks, 2014.

[23] K. C. Sia, J. Cho, and H. K. Cho. Efficient monitoring algorithm for fast news alerts. *TKDE*, 19(7):950–961, 2007.

[24] M. Tsytsarau, T. Palpanas, and M. Castellanos. Dynamics of News Events and Social Media Reaction. In *KDD*, pages 901–910, 2014.

[25] P. O. S. Vaz de Melo, C. Faloutsos, R. Assunção, R. Alves, and A. A. F. Loureiro. Universal and Distinct Properties of Communication Dynamics: How to Generate Realistic Inter-event Times. *CoRR*, abs/1403.4, 2014.

[26] P. O. S. Vaz de Melo, C. Faloutsos, R. Assunção, and A. A. F. Loureiro. The Self-Feeding Process: A Unifying Model for Communication Dynamics in the Web. In *WWW*, pages 1319–1330, 2013.

[27] Y. Wu, C. Zhou, J. Xiao, J. Kurths, and H. J. Schellnhuber. Evidence for a bimodal distribution in human communication. *PNAS*, 107:18803–18808, 2010.

[28] C. M. Zhang and V. Paxson. Detecting and Analyzing Automated Activity on Twitter. *Lecture Notes in Computer Science*, 6579:102–111, 2011.