# Red Light/Green Light: A Lightweight Algorithm for, Possibly, Fraudulent Online Behavior Change Detection

Vitali Herrera-Semenets[1], Raudel Hernández-León[1], Lázaro Bustio-Martínez[2(✉)], and Jan van den Berg[3]

[1] Advanced Technologies Application Center (CENATAV). 7a ♯ 21406, Playa, C.P. 12200, Havana, Cuba
{herrera,rhernandez}@cenatav.co.cu

[2] Universidad Iberoamericana, DEII, Prolongación Paseo de Reforma 880, 01219 CDMX, Mexico
lazaro.bustio@ibero.mx

[3] Intelligent Systems Department, Delft University of Technology, Mekelweg 4, 2628, CD Delft, The Netherlands
j.vandenberg@tudelft.nl

**Abstract.** Telecommunications services have become a constant in people's lives. This has inspired fraudsters to carry out malicious activities causing economic losses to people and companies. Early detection of signs that suggest the possible occurrence of malicious activity would allow analysts to act in time and avoid unintended consequences. Modeling the behavior of users could identify when a significant change takes place. Following this idea, an algorithm for online behavior change detection in telecommunication services is proposed in this paper. The experimental results show that the new algorithm can identify behavioral changes related to unforeseen events.

**Keywords:** Online data processing · Behavior changes · Anomaly detection · Concept drift · Cybersecurity · Multimodal data analysis

## 1 Introduction

Today telecommunications are essential for people and companies: in other words, the more users are connected to telecommunication services, the greater the communication possibilities and needs. Telephony-related telecommunications services carry a large volume of call, message and data traffic every day. Such services can be used to monetize third party services, also in unintended ways [7]. In this sense, telephony can become a very profitable environment for fraud schemes.

There are common techniques that fraudsters often use, such as: malicious software (malware) and the phone call scams [7]. Malware that infects mobile


O. Pichardo Lagunas et al. (Eds.): MICAI 2022, LNAI 13613, pp. 316–327, 2022.
https://doi.org/10.1007/978-3-031-19496-2_24

phones may initiate phone calls or send short messages stealthily. Although many of such techniques are known, the number and diversity of these continues growing. Fraudulent techniques are becoming increasingly difficult to track and investigate due to their frequency, their layers of anonymity, and their global nature.

To deal with the latter shortcoming, a lightweight algorithm for online behavior change detection in telecommunication services is proposed in this paper. The algorithm is designed to process large data streams, while updating the behavior of each user in real-time without requiring large computing resources. The experimental results, show that the new algorithm can identify behavioral changes related to unforeseen events (such as losing the phone, texts or calls not made by the user, among others) that, in some cases, can be linked to malicious activities. In addition, a case study on SMS messaging is presented, which shows the feasibility of using the proposed algorithm in real-world scenarios.

The remainder of this paper is structured as follows. Related works are described in Sect. 2. The proposed algorithm is introduced in Sect. 3. In Sect. 4, the experimental results are discussed. Then, in Sect. 5, a case study on real SMS messages dataset is presented. Finally, the obtained conclusions are outlined in Sect. 6.

## 2 Related Work

To date, there is a wide variety of classification algorithms proposed for the detection of malicious activities in telecommunications services. Specifically, this work focuses on anomaly-based algorithms.

Anomaly-based algorithms first identify normal behavior and look for variations in behavior that represent an anomaly. Generally, the normal behavior is identified using an unlabeled training collection consisting of historical information. Then, the normal behavior defined can be compared to the current behavior in order to determine if significant changes occur that may indicate an anomaly. These algorithms have some advantages: (1) subtle changes in the subscribers behavior can be detected, and (2) a prior domain knowledge is not required, which allows to identify new unknown malicious activities. On the other hand, a common drawback is: (1) certain anomalies are associated with normal behaviors (increase of false positives).

The drawback mentioned above is usually related to the fact that a user changes his behavior and starts behaving differently than he did originally. In such cases, it cannot be determined if this behavior change is related to fraud, but at least it can be accepted as suspicious, which may result into false positives. Fraudulent user behavior changes are characterized by a potentially high number of user action changes related to sending SMS messages, calls, among others, within a short period of time [3].

Some algorithms based on profiling human social behaviors have been proposed to detect anomalies. Stolfo et al. [10] introduce an algorithm that models

behavioral profiles based on user cliques, Hellinger distance, and cumulative distributions for email users. SMSBotHunter is an anomaly detection approach that uses a one-class classification to detect SMS botnets on mobile devices [2].

Anomaly detection in mobile phone networks has been addressed in several works. For example, calling activities have been analyzed to detect fraud on mobile phones [6], as well as the mobility patterns of mobile devices have been profiled to detect cloning attacks [12]. With the rise of smartphones, a growing number of malware has been identified on these devices. This fact has led to the proposal of various approaches to detect mobile malware that work by profiling the behavior of normal applications [13].

A large portion of anomaly-based algorithms are designed to be used offline, since the high dimensionality and volume of the data negatively affects the efficiency of conventional approaches, such as algorithms based on distance, density and clustering [1]. Although there are proposals that allow a more efficient processing of large volumes of data, these are designed to be used in distributed environments that require large computing resources [11]. However, there are "lighter" proposals designed for the online detection of behavior changes such as the algorithm proposed by Shaeiri et al. [8]. This algorithm have a time complexity, in the worst case, of $\mathcal{O}(N * M)$, where $N$ is the number of users and $M$ is the amount of phone numbers that have interacted with the user, which makes it the most efficient of the algorithms analyzed in this section.

In general, from the algorithms described in this section, at least two issues can be identified. One of them is the generation of multiple alarms, where each alarm does not have the same level of importance for the analyst. The other issue is the benefit of timely alarms. Fraud must be detected as soon as possible and the algorithm used must reflect it. The benefit of a timely alarm can often be quantified. In telecommunications services fraud detection scenarios, the cost of delaying an alarm can have a negative effect on the economy of a company or a user.

The algorithm proposed in this work addresses both issues. In addition, it is designed to be applied in scenarios that do not have large computing resources, so its efficiency, in terms of time complexity, is a fundamental aspect to be considered.

## 3 Red Light/Green Light Algorithm

The Red Light/Green Light (RLGL) algorithm proposed in this paper has its name based on the popular children's game of the same name [5].

The RLGL algorithm analyzes the daily behavior of each user and checks for any significant change that suggest anomalous behavior. To do this, it is analyzed how the user's behavior has been at different times of the day and compares it with the historical profile of the user. If the probability that the user performs an action, be it sending an SMS message or making a call at some point in the day, increases above a defined threshold, it is considered a change in behavior and is reported.

**Algorithm 1:** RLGL($r, th, detect, minNumRec$)

**Input**: $r$: new record, $th$: threshold increase, *detect*: anomaly detection enabled, $minNumRec$: minimum number of records
**Output**: *changeAlert*: alert of detected changes

```
1  currentday = -1
2  activeUsers = [ ]
3  usersHistory, tmpUsersHistory = Hash_Table()
4  if detect == True then
5  |  usersHistory, tmpUsersHistory = Load_Profiles()
6  while !stopSignal do
7  |  if detect == True then
8  |  |  if r.Date.weekDay != currentDay then
9  |  |  |  currentday = r.Date.weekDay
10 |  |  |  foreach userNumber in activeUsers do
11 |  |  |  |  changeList = Analyze_Behavior(usersHistory[userNumber],
   |  |  |  |  tmpUsersHistory[userNumber], th, minNumRec)
12 |  |  |  |  if changeList != Empty then
13 |  |  |  |  |  changeAlert = ChangeAlert(userNumber,anomaliesList,Date())
14 |  |  |  |  |  ThrowAlert(changeAlert)
15 |  |  |  activeUsers = [ ]
16 |  |  |  tmpUsersHistory = usersHistory
17 |  |  activeUsers.Add(r.userNnumber)
18 |  if usersHistory.hasKey(r.userNnumber) then
19 |  |  usersHistory[r.userNnumber].Update_Statistics(r)
20 |  else
21 |  |  usersHistory[r.userNnumber] = NewUser().Update_Statistics(r)
22 Save_Profiles(usersHistory)
```

The RLGL algorithm requires some input parameters to be defined. As it is shown in Algorithm 1, the input parameter *detect* will indicate if the algorithm is going to be executed in anomaly detection mode ($detect = True$), or if it is only going to model the behaviors of the users ($detect = False$). The latter is recommended for the initial execution of the algorithm, since by having previous users profiles, it is possible to reduce false positives in the anomaly detection mode. In this way, the algorithm can model the behaviors of the users, by updating their statistics, for a suitable period of time (see lines 18–21 in Algorithm 1). The time defined for modeling the behavior of the users must be in correspondence with the context where the algorithm is applied. When it is decided to finish the previous process, a stop signal is sent (see line 6 in Algorithm 1). Next, it saves the modeled profiles of each user and ends its execution (see line 22 in Algorithm 1).

The *Update_Statistics* method consists of updating the count of actions performed by a user based on the time range of the day in which the new record $r$ originated (see lines 19 and 21 in Algorithm 1). For this, a full day is divided into four 6-h time ranges. Each time range is associated with a different part of the day: (0:00–5:59) early morning, (6:00–11:59) morning, (12: 00–17:59) afternoon and (18:00–23:59) evening. Thus, if a user sends an SMS at 6:30, the algorithm updates the statistics of said user by increasing the number of SMS sent in the time range (6:00–11:59) and the total number of SMS sent.

When the anomaly detection mode is activated, the algorithm proceeds to load the previously modeled user profiles in two hash tables: *usersHistory* and

**Algorithm 2:** Analyze_Behavior($currentP, historicalP, pDay, th, minNumRec$)

**Input**: $currentP$: current user profile, $historicalP$: historical user profile, $th$: threshold increase, $minNumRec$: minimum number of records
**Output**: $changeList$: list of detected changes

```
1 anomalies = [ ]
2 if currentP.totalRecords > minNumRec then
3     foreach timeRange in range(0,4) do
4         P(current) = currentP.timeRange[timeRange] / currentP.totalRecords
5         P(historical) = historicalP.timeRange[timeRange] / historicalP.totalRecords
6         changeTh = P(historical) + th
7         if P(current) > changeTh then
8             changeList.Add(Change(timeRange, P(current), changeTh))
9 return changeList
```

$tmpusersHistory$ (the latter is a temporary copy of the first (see lines 4–5 in Algorithm 1)). The hash table $usersHistory$ will continue to be updated during the day, while the temporary copy $tmpusersHistory$ remains unchanged and will serve as a historical profile for behavioral analysis. However, as the algorithm is running online, new user records could arrive that have not been considered during the behavior modeling process. In this case, the input parameter $minNumRec$ is included, which defines the minimum number of records that a user must generate so that the algorithm can perform the behavior analysis. A record stores information (source number, destination number, date, etc.) about an action performed by the user, be it making a call, sending an SMS message, among others.

Since the RLGL algorithm analyzes the daily behavior of each user, it must recognize when it is time to analyze users behaviors. To do this, it checks if the day of the week present in the date of the new record is different from the one stored in the $currentDay$ variable. If yes, it suggests that the new record already belongs to a new day. Therefore, the $currentDay$ variable is updated (see line 9 in Algorithm 1) and the behavior analysis of each user is performed (see lines 10–14 in Algorithm 1).

A small number of records may not be sufficient to define the historical behavior of a user. Therefore, the first step of the behavioral analysis algorithm is to check if the number of records generated by the user exceeds the $minNumRec$ variable (see line 2 in Algorithm 2). This step ensures that each user has at least a minimum number of records defining the historical profile, which guarantees a better behavioral analysis.

If the above condition is satisfied, for each time range of the day, the probability that a user generates a record is computed (see lines 3–8 in Algorithm 2). Note that the current probability and the historical probability are computed (see lines 4 and 5 in Algorithm 2). The probability is given by the number of records generated in a time range over the total number of records generated. The threshold $changeTh$, which will indicate when a change in behavior occurs,

is given by the sum of the historical probability and a coefficient $th$ defined by the analyst (see line 6 in Algorithm 2).

To determine if there is a behavior change in the evaluated time range, it is checked whether the current probability exceeds the computed threshold (see lines 7–8 in Algorithm 2). If this condition is satisfied, it means that there has been an increase in the number of user-generated records indicating a change in user behavior. If so, a behavior change alert is created and added to the list of detected changes to be reported (see line 8 in Algorithm 2). After analyzing each time range, the list of detected changes $changeList$ is returned (see line 9 in Algorithm 2).

If any change in the analyzed user behavior is detected, an alert with the necessary information is created and reported (see lines 12–14 in Algorithm 1).

After analyzing each user that has been active in the previous day, some variables are reset (see lines 15 and 16 in Algorithm 1) and the statistics of the users during the current day begin to be updated.

The algorithm proposed in this work is considered "lightweight", not only because of its heuristics, but also by its time complexity. The proposed algorithm iterates over each active user, with a time complexity of $\mathcal{O}(N)$, and gets the user profile from a Hash Table, which has a computational complexity of $\mathcal{O}(1)$. Therefore, when applying the sum rule, the time complexity that defines the RLGL algorithm would be $\mathcal{O}(N)$, which makes it more efficient than the proposals analyzed in the previous section.

## 4 Experimental Results

To evaluate the proposed algorithm, an unlabeled phone calls data sets (13035 records) were used. Such data set is the result of the mobile phones usage, collected through monitoring devices of 27 users during a 5-month study [4]. The first three months were used to model user behavior and the last two months were used to detect behavioral changes. The experiments were conducted on a PC equipped with a 3.2 GHz Intel Quad-Core processor, 8 GB of RAM memory running Ubuntu 18.04 OS.

### 4.1 Experiments on Phone Calls

In the phone call data set, during the initial three months, there were some users that only made between 10 and 30 calls, which is a low number of calls, compared to those performed by the rest of the users. This fact can cause false positives, since if the user makes two or three calls during a day, the probability of said user can increase considerably and can be reported as a change in behavior. To avoid this type of situation, the RLGL algorithm allows defining a parameter for a minimum number of records to be considered to analyze a user ($minNumRec = 90$).

The experimental results are shown only for the threshold increase parameter $th = 0.02$. This threshold is enough to analyze the behavior of the proposed algorithm during the experiments. Using a smaller threshold would increase the

number of behavioral changes detected, an opposite effect would be obtained by increasing the threshold.

After applying the RLGL algorithm on the phone call data set, using the threshold increase parameter $th = 0.02$, 6 users with behavioral changes were detected. The Table 1 shows the users with identified behavior changes, as well as the time range, the date and the day that it represents in the five months processed (being day 0 on 2/9/2010 and day 145 on 30/1/2011).

**Table 1.** Behavioral changes detected in the phone calls data set.

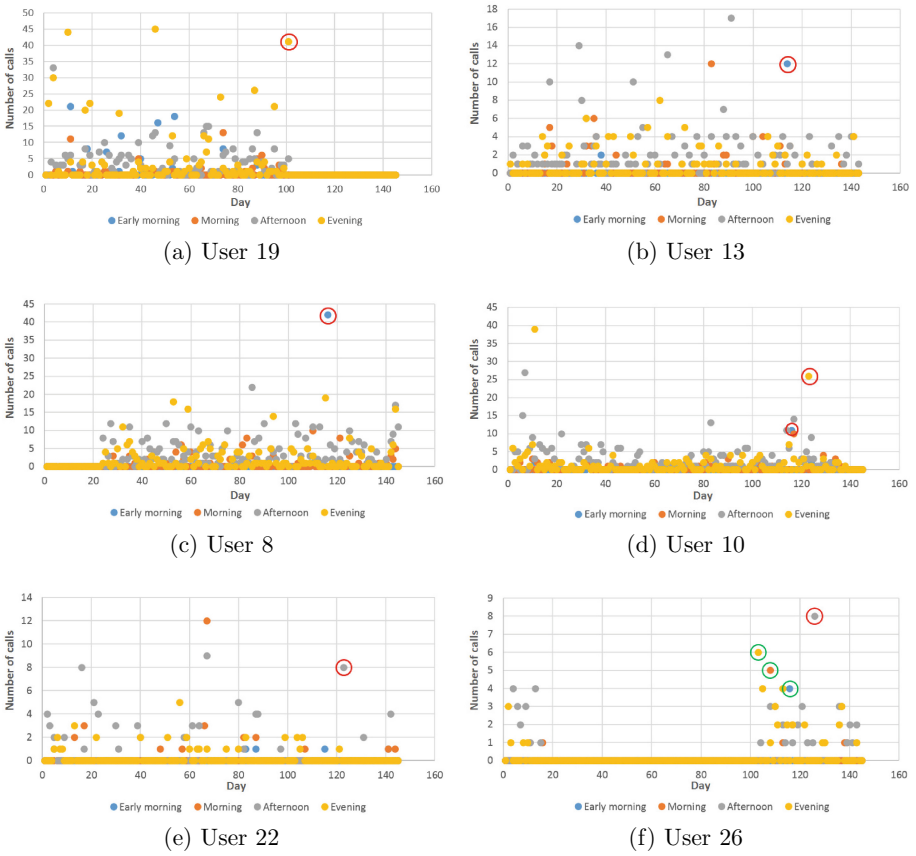
| Time range | Date | Day | User |
|---|---|---|---|
| Evening | 17/12/2010 | 101 | 19 |
| Early morning | 1/1/2011 | 116 | 13 |
| Early morning | 1/1/2011 | 116 | 8 |
| Early morning | 1/1/2011 | 116 | 10 |
| Evening | 8/1/2011 | 123 | |
| Afternoon | 8/1/2011 | 123 | 22 |
| Afternoon | 11/1/2011 | 126 | 26 |

Figure 1a shows the daily behavior of the user 19 during the five months of registered calls. Note that the first 90 days are associated with the 3 months used to model the user behavior. Although during that time, two days were recorded with more than 40 calls made at the evening, in general, this user does not make more than 20 calls in that time range. Therefore, the behavior change highlighted with a red circle in Fig. 1a is due to the user making 41 calls during the evening. In this sense, the probability of making calls for the user 19 at the evening increases above the threshold, which represents a behavior change.

The probability that user 13 will make a call in the early morning is very low. As shown in Fig. 1b, the highest number of calls that user 13 made in the early morning were 2. In this sense, the behavior change detected (highlighted in a red circle in Fig. 1b) is related to 12 calls made in the early morning, which increased the probability of making a call in that time range above the threshold. A similar case can be seen in Fig. 1c where user 8, with only 3 calls recorded in the early morning history, made 42 calls in a single early morning.

Figure 1d shows, highlighted with a red circle, the two behavioral changes that were detected for the user 10. The first one occurred when the user made 11 calls at the early morning. This figure represents more than 5 times the maximum number of calls made by user 10 in a single early morning. As can be seen in Fig. 1d, the second change in behavior occurred at the evening making 26 calls, when user 10 usually made no more than 5 calls at night. Both events represent a considerable increase in the probability of making calls at the corresponding times, which is why they are identified as behavioral changes.

The user 22 had a behavior change similar to user 19 discussed above. As shown in Fig. 1e, during the behavioral modeling, two days were recorded with 8 or more calls in the afternoon. However, in general, the user does not make more




(a) User 19 (b) User 13

(c) User 8 (d) User 10

(e) User 22 (f) User 26

**Fig. 1.** Number of calls performed each day by users 19, 13, 8, 10, 22 and 26. (Color figure online)

than 4 calls in the afternoons. For this reason, a behavior change is identified when making 8 calls in a single afternoon (see Fig. 1e, red circle).

The user 26 is an example of users who made few calls during the three months of behavioral modeling. In Fig. 1f, it can be seen that in the first month the user 26 did not exceed 30 calls in total, and the remaining two months did not make any calls. As RLGL is an online algorithm, it is capable of constantly modeling user behavior and when the user exceeds the number of calls established in the *minNumRec* parameter, the user behavior is analyzed. That is why in some cases, such as those highlighted with a green circle in Fig. 1f which could turn out to be anomalies, a behavioral analysis is not performed, since at that time the number of calls does not exceed the value of the parameter *minNumRec*. On the other hand, when the user makes 8 calls in a single afternoon (see Fig. 1f, red circle), the *minNumRec* value is reached, which is why its behavior is analyzed detecting a change.

### 4.2 Results Discussion

As can be seen in Table 1 some behavior changes are associated with festive events such as new year celebrations. Following the heuristics implemented in the algorithm, which monitors different time ranges in a day, it could be interesting and positive to incorporate other features. For example, include monthly time ranges, since users may show variations in their behavior in certain months, say due to vacations, festive events among others.

User 19 is reported by the authors of the data set as having lost the phone on December 17, 2010, the same day that the proposed algorithm identified a change in user behavior. Perhaps it could be related to someone else finding the phone and starting to use it.

The authors of the data set also report that user 26's device apparently had a serious malfunction causing calls to be placed on its own. The RLGL algorithm identified a behavior change in this user. This unforeseen event could also be associated with some malicious program that causes such behavior.

User 10 is another one that was reported by the authors of the data set. In this case, the user 10 left the school, where the study was conducted, at the end of 2010. This user continued to use the phone, but the authors note that patterns in the data may have changed due to non-attendance of school. The proposed algorithm identified two behavioral changes in user 10 during January 2011.

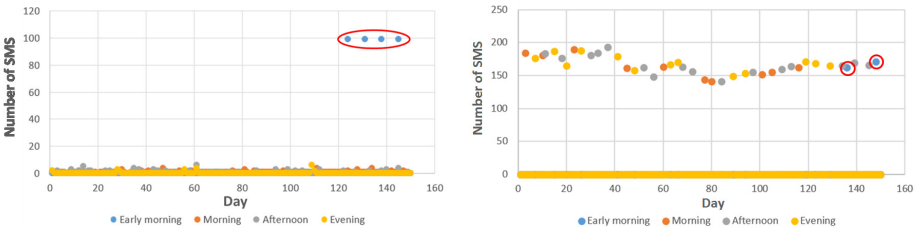
Commercially, it can be an interesting fact to evaluate the users detected with behavioral changes, since for some particular reason, if it is not associated with malicious activity, they decided to increase the use of the contracted service, which can help the company to reorient its commercial strategy or create new rate plans based on these situations.

As discussed previously, some fraudulent user behaviors are characterized by the unexpectedly high values of call counts or SMS messages within a period. As can be seen from the results achieved, the proposed algorithm can identify behaviors similar to those that describe a fraudulent behavior. At the fist stage, we cannot determine if these behavior changes were exactly fraud, but we can accept them as suspicious behaviors.

## 5 A Case Study on Real SMS Messages Dataset

The performance and behavior of the RLGL algorithm have been verified and validated by a Mexican telecommunications company (named hereafter as "TC") in a real scenario[1]. Following the same experimental design described in Sect. 4, five months of SMS data were collected, hereby obtaining a dataset composed of 21772546 records. Each record contains the source and destination number, as well as the date that the SMS message was sent. From the dataset created, three

[1] Due to privacy and commercial policies of the telecommunication company, names, data, and other information that could lead to a personal or commercial information leakage are not offered. This was guaranteed by a Statement of Confidentiality signed between the research authors and TC.




(a) User A (b) User B

**Fig. 2.** Number of SMS messages sent each day by two users.

months (13963527 records) were used as baseline for modeling user's behavior. The other two months (7809019 records) were used to detect changes that deviate from the baseline. Also, the parameters $minNumRec$ and $th$ got the same values as used in the experiments reported in Sect. 4, 90 and 0.02 respectively.

In this case study, the RLGL algorithm detected a behavioral change by a user (referred here as "User A") on the $124^{th}$ day, where User A sent 100 SMS messages to several cellphone numbers in a range of $xxx - xxx - 0000$ to $xxx - xxx - 0099$. All these SMS messages were sent in the early morning (at 03:00 am), which contrasts with the historical behavior of this user. In day 131, User A sent others 100 SMS messages, but this time, the targets were moved to other 100 cellphones number in the range $xxx - xxx - 0100$ to $xxx - xxx0199$. This behavior was repeated four times every seven days after the $124^{th}$ day. Figure 2a shows the behavior of User A during the five months evaluated, and the behavior changes detected are highlighted with a red oval.

Considering the behavioral change detected by the RLGL algorithm, the security analysts conducted a detailed study of the behavior of User A. This study determined that User A was not a bot, but a legitimate user that kept a normal historical behavior. Apparently, before the $124^{th}$ day, User A's device was infected by malware that sent 100 SMS at $03:00$ am to 100 different users. This behavior is repeated periodically every seven days at the same hour. Furthermore, the SMS sent contain a fraudulent URL inviting the victims to access a supposed Facebook address to update their credentials. In this way, the fraudsters can obtain personal information from users who enter their credentials on the URL suggested in the SMS message. This can be seen as a typical case of scam or Phishing [9].

The RLGL algorithm identified 123 other users with behavior similar to that of User A. After examining each of the reported users, security analysts concluded that their devices were infested with the same malware as User A.

Another 18 users were also identified with specific behavior changes, that is, a notable increase in the probability of sending SMS messages at a certain time (mainly at the evening and at the early morning). These cases only occurred on a single day, which is why security analysts determined that they may be normal behaviors associated with some personal event.

Some false positives were also reported. Fortunately, these cases could have been ruled out by security analysts, since they were associated with SMS messages sent automatically to subscribers of a news agency (referred here as "User B"). This change in behavior was detected since, for four months, the news agency only sent SMS messages between the morning and evening hours. It was not until the $5^{th}$ month that a group of SMS messages was sent on two occasions at the early morning, which was detected as a change in behavior (highlighted in red circles in Fig. 2b). Another 7 news agencies were detected with a similar behavior to that of User B. These false positives do not represent a problem for security analysts, since they have identified these, and other, news agencies and can filter their SMS, so that they will not be processed by the RLGL algorithm.

The RLGL algorithm processed a data flow associated with 5 months of SMS messaging without showing any deterioration in its performance, in terms of efficiency. A large part of the users identified with a behavior change were verified by security analysts, who concluded that they were related to a malicious activity. In addition, another group of users with behavioral changes not associated with malicious activities, also was detected. However, this phenomenon is known to security analysts, so the related false positives can be filtered out to unnecessary detection by the RLGL algorithm.

## 6 Conclusions

A new algorithm for online behavior changes detection was presented in this work. Its linear time complexity makes it an efficient algorithm, with a feasible use in scenarios with low computing resources, and lighter than other proposals reported in the literature.

Although the detection of a behavioral change does not necessarily imply a malicious activity, the alarms provided by the RLGL algorithm facilitate the investigative work of security analysts. The experimental results show that RLGL detects behavioral changes related to festive events that, fortunately, can be ruled out by security analysts. In addition, RLGL also detected unforeseen events that, given their nature, could be linked to some malicious activity. A sample of them is the case study presented, where the RLGL algorithm identified 124 users who carried out a malicious activity associated with a scam or Phishing.

In future work, it is intended to incorporate other variables to the modeling of user behavior. The challenge is to do this without negatively affecting the efficiency of the algorithm, as well as its time complexity.

**Acknowledgement.** This research was supported by the Universidad Iberoamericana (Ibero) and the Institute of Applied Research and Technology (InIAT) by the project "Detection of phishing attacks in electronic messages using Artificial Intelligence techniques."

## References


1. Ahmed, M., Mahmood, A.N., Hu, J.: A survey of network anomaly detection techniques. J. Netw. Comput. Appl. **60**, 19–31 (2016)
2. Faghihi, F., Abadi, M., Tajoddin, A.: Smsbothunter: a novel anomaly detection technique to detect SMS botnets. In: 2018 15th International ISC (Iranian Society of Cryptology) Conference on Information Security and Cryptology (ISCISC), pp. 1–6. IEEE (2018)
3. Kilinc, H.H.: A case study on fraudulent user behaviors in the telecommunication network. Electrica **21**(1), 74–85 (2021)
4. McDiarmid, A., Bell, S., Irvine, J., Banford, J.: Nodobo: detailed mobile phone usage dataset (2013). Unpublished paper, accessed http://nodobo.com/papers/iet-el.pdf on pp. 9–21
5. Nakamura, T., Munekata, N., Nakamura, F., Ono, T., Matsubara, H.: Universal game based on traditional children's outdoor games. In: Anacleto, J.C., Fels, S., Graham, N., Kapralos, B., Saif El-Nasr, M., Stanley, K. (eds.) ICEC 2011. LNCS, vol. 6972, pp. 59–64. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-24500-8_7
6. Peng, L., Lin, R.: Fraud phone calls analysis based on label propagation community detection algorithm. In: 2018 IEEE World Congress on Services (SERVICES), pp. 23–24. IEEE (2018)
7. Sahin, M., Francillon, A.: Understanding and detecting international revenue share fraud. In: Proceeding of the Network and Distributed System Security Symposium (NDSS 2021), Reston, VA. The Internet Society (2021)
8. Shaeiri, Z., Kazemitabar, J., Bijani, S., Talebi, M.: Behavior-based online anomaly detection for a nationwide short message service. J. AI Data Mining **7**(2), 239–247 (2019)
9. Sonowal, G.: Introduction to Phishing. In: Phishing and Communication Channels, pp. 1–24. Apress, Berkeley, CA (2022). https://doi.org/10.1007/978-1-4842-7744-7_1
10. Stolfo, S.J., Hershkop, S., Hu, C.W., Li, W.J., Nimeskern, O., Wang, K.: Behavior-based modeling and its application to email analysis. ACM Trans. Internet Technol. (TOIT) **6**(2), 187–221 (2006)
11. Thudumu, S., Branch, P., Jin, J., Singh, J.J.: A comprehensive survey of anomaly detection techniques for high dimensional big data. J. Big Data **7**(1), 1–30 (2020). https://doi.org/10.1186/s40537-020-00320-x
12. Ullah, F., Naeem, M.R., Mostarda, L., Shah, S.A.: Clone detection in 5g-enabled social IoT system using graph semantics and deep learning model. Int J. Mach. Learn. Cybernet. **12**, 3115–3127 (2021)
13. Yu, B., Fang, Y., Yang, Q., Tang, Y., Liu, L.: A survey of malware behavior description and analysis. Front. Inf. Technol. Electron. Eng. **19**(5), 583–603 (2018). https://doi.org/10.1631/FITEE.1601745