# A Real-Time Fraud Detection Algorithm Based on Usage Amount Forecast

Kun Niu[1], Zhipeng Gao[2(✉)], Kaile Xiao[1], Nanjie Deng[2], and Haizhen Jiao[1]

[1] School of Software Engineering,
Beijing University of Posts and Telecommunications, Beijing 100876, China
[2] State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China
gaozhipeng@bupt.edu.cn

**Abstract.** Real-time Fraud Detection has always been a challenging task, especially in financial, insurance, and telecom industries. There are mainly three methods, which are rule set, outlier detection and classification to solve the problem. But those methods have some drawbacks respectively. To overcome these limitations, we propose a new algorithm UAF (Usage Amount Forecast). Firstly, Manhattan distance is used to measure the similarity between fraudulent instances and normal ones. Secondly, UAF gives real-time score which detects the fraud early and reduces as much economic loss as possible. Experiments on various real-world datasets demonstrate the high potential of UAF for processing real-time data and predicting fraudulent users.

**Keywords:** Real-time Fraud Detection · Usage Amount Forecast · Telecom industry

## 1 Introduction

With development of society and evolution of technology, economic fraud which is less in the past has gradually risen [1, 2], resulting in heavy loss of many enterprises and organizations. Therefore, from theoretical research to practical application, identification and monitoring fraud [3, 4] have caught more attention than before.

### 1.1 Related Work

Sieve method based on rule set used historical data related to fraud users' behavior feature to define a series of rules [4–6]. If users break pre-defined rules, system will warn administrators by reporting an emergency. For example, a mobile phone user is presumed to be fraud if his 'monthly cumulative charge exceeds 1,000 USD.

Outlier detection uses intelligent model to detect special samples in total, then system submits the outliers to administrators [7]. For example, by using density-based algorithm DBOM [8], abnormal degree of each instance in feature space is measured by LOF (local outlier factor).

W. Che et al. (Eds.): ICYCSEE 2016, Part I, CCIS 623, pp. 87–95, 2016.
DOI: 10.1007/978-981-10-2053-7_9

Another solution is category discrimination [9]. It uses classification methods in data mining, such as decision tree [10], support vector machine [11], neural network [12–14], to classify and evaluate new samples. According to such IF-THEN rules, a person whose monthly outbound times are more than 6,000 may be regarded as a fraud user.

However, those methods are not good at processing stream data. Among those methods, some are not easy to set up parameters, and some others cannot teach themselves to fit variable data. In addition, those methods limit the capacity of application system for their high calculation complexity [15].

### 1.2 Our Contributions

To overcome these limitations, we present a new algorithm UAF (Usage Amount Forecast). We analyze variables independent of total amount to predict whether a user is fraudulent. The experiment shows that UAF is superior over existing relative methods in terms of runtime, accuracy, and robustness.

Overall, the contributions of our work on real-time fraud detection are as follows:

1. UAF does not need cumulative variables, which makes it has low computational cost.
2. UAF only computes variables which are independent of total amount, so it is able to catch fraud timely.
3. UAF can be used on real-time scenarios. The scores update synchronously while bills are inputted continuously.

The rest parts are organized as follows. In Sect. 2, we demonstrate the main idea of UAF, and give notions and definitions. The complete process of UAF with pseudo-code is showed in Sect. 3. Experiment results are presented in Sect. 4 and we conclude our work in Sect. 5.

## 2 Preparation of Your Paper

### 2.1 Notions

Assume that dataset D is the feature space to be studied. It contains n instances and m attributes. It is represented as $D = \{x_1, \ldots, x_n\}$ and the matrix form is $D = \left\{x_1^T, \ldots, x_n^T\right\} \in Z^{n*m}$. For any instance $x_i$ of D, we have $x_i = \{x_{i1}, \ldots, x_{im}\}^T$. Here $x_{ik}$ is the discretized result of the $i^{th}$ instance on $k^{th}$ attribute. For each fraudulent sample, we also have $y_j = \left\{y_{j1}, \ldots, y_{jm}\right\}^T$. Here $y_{jk}$ is the discretized result of the $j^{th}$ sample on $k^{th}$ attribute.

### 2.2 Real-Time Fraud Detection

To find out whether a user is fraudulent, we have to know how to accurately divide the target user sets into two subsets, fraud and normal.

### A. Usage Amount Forecast

In telecom industry, users pay bills periodically. The billing cycle is usually a month, users randomly generate call records, surf the internet and purchase value-added services. Data scientists working for operators collect and analyze these consuming data with big data techniques. The attributes used to describe users can be divided into two types, cumulative attributes and feature ones.

As shown in Fig. 1, the cumulative attributes are increasing monotonously when consuming records generate, but the feature attributes are stable throughout the whole billing cycle. The feature attributes are independent of usage amount and almost constant for a single user. That is why they are called feature attributes.

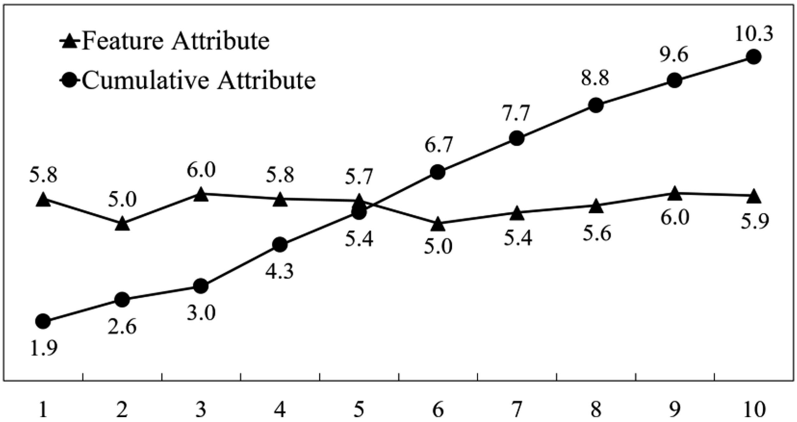

**Fig. 1.** Cumulative attribute and feature attribute

With large sample analysis, we find out that the two types have some specific correlations. The cumulative attributes can be predicted by feature attributes. When we detect fraud, the cumulative attributes are useless because they need long enough time to increase and warn, which is really belated. So when we use feature attributes only, as shown in Fig. 2, we may estimate the potential risk of total usage and locate fraud timely.

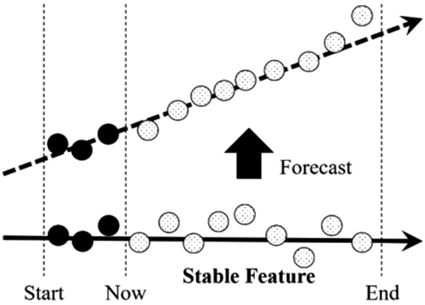

**Fig. 2.** The time window of usage amount forecast

**B. Similarity Evaluation**

Although we know feature attributes are more useful in fraud detection, a mechanism of scoring is still needed. Generally speaking, close objects have similar patterns, such as K-NN (K-Nearest Neighbors) algorithm [17]. The user who shows similar features to given fraudulent samples has a higher risk of fraud.

Therefore, we give the definition of Similarity Score (SS).

$$\text{Definition}: \forall i = 1, \cdots, n; j = 1, \cdots, n', SS(x_i) = min_j\left(\sum_{k=1}^{m} \left|x_{ik} - y_{jk}\right|\right) \quad (1)$$

$\sum_{k=1}^{m} \left|x_{ik} - y_{jk}\right|$ is the Manhattan Distance between user $x_i$ and fraudulent user $y_j$. Manhattan Distance not only reduces the impact of correlation between attributes, but also greatly reduces computational complexity than commonly used Euclidean Distance.

## 3 Algorithm Description

The whole process of UAF is shown in Fig. 3, which includes 2 main phases, data prepare and SS calculation.

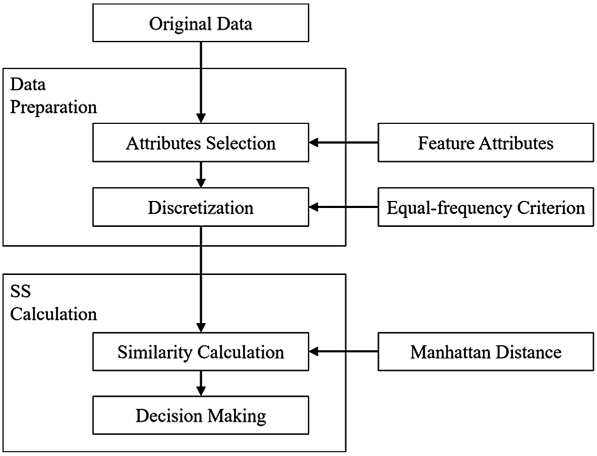

**Fig. 3.** Process of UAF

### 3.1 Data Prepare

First of all, we do data cleaning, e.g., Missing value interpolation and outlier detection. Then the predefined feature attributes are generated automatically. Some basic attributes are obtained directly from the original datasets such as calling duration and times. The other feature attributes are obtained by transforming, for example, the average duration of single call is defined as cumulative duration divided by cumulative times.

The third step is discretization. Because of frequent left avertence of normal distribution in telecom industry, equal-frequency criterion is more suitable than the common equal-width criterion. For example, assuming L is range of an attribute, K is number of segments, N is number of instances, the critical values of equal-width method are $\left\{0, \frac{L}{K}, \frac{2*L}{K}, \ldots, \frac{K*L}{K}\right\}$, the critical values of equal-frequency method are $\left\{x_1, x_{\left[\frac{N}{K}\right]}, x_{\left[\frac{2*N}{K}\right]}, \ldots, x_{\left[\frac{K*N}{K}\right]}\right\}$.

### 3.2 SS Calculation

Firstly, SS is calculated by (1). After that, SS has to be normalized and reversed for displaying. Scoring range is from 0 to 100. So we have (2).

$$SS(x_i) = 100 - \frac{100 \times (SS(x_i) - SS_{min})}{SS_{max} - SS_{min}} \tag{2}$$

The last step is decision process. When SS is higher than decision threshold, the user will be assumed as fraud, and the system triggers alarm to administrators, otherwise updates the user SS score. The decision threshold is an important parameter which can adjust and optimize by actual results.

## 4 Experiments and Results

### 4.1 Empirical Evaluation

**A. Datasets**

In this work, we use nine datasets to evaluate the performance of UAF. Description of the datasets is given in Table 1, for example, the date set A-1 means the data is from city A, which has 1,715,459 bills and 177,761 users in the first month. Additionally, a library which includes 6 international roaming fraudulent users is used as reference.

**Table 1.** Description of used datasets

| Datasets | #Bills | #Users |
|---|---|---|
| A-1 | 1,715,459 | 177,761 |
| B-1 | 224,697 | 43,461 |
| C-1 | 72,191 | 7,246 |
| B-2 | 465,266 | 41,720 |
| B-3 | 199,459 | 38,708 |
| B-4 | 578,252 | 51,631 |
| B-5 | 292,244 | 56,165 |
| B-6 | 468,491 | 45,382 |
| B-7 | 463,587 | 46,854 |

All feature attributes are divided into boxes of total number n with equal-frequency discretization. Improper n may result in failure or over-fitting. The following results are the best performance of different n.

### B. Attributes

Considering usage amount forecast mechanism, we select attributes which are dependent of total amount, such as average call duration and average times of each number. Description of attributes is showed in Table 2.

**Table 2.** Description of used attributes

| Name | Method of Calculation |
|---|---|
| Avg_call_dur | $= \frac{Total\ call\ duration}{Total\ call\ numbers}$ |
| Fluc_call_dur | $= \frac{\text{Stdev(total call duration)}}{\text{avg}_{\text{call\_dur}}}$ |
| Avg_call_invl | $= \frac{\text{First call start time–last call finish time}}{\text{total call numbers}}$ |
| Cnt_call_num | $= \text{Count(non} - \text{repeated call number)}$ |
| Avg_times_num | $= \frac{\text{Total call times}}{\text{total call numbers}}$ |
| Cnt_ctry | $= \text{Count(non} - \text{repeated roaming countries)}$ |
| High_fee_share | $= \frac{\text{Total high settlement call fee}}{\text{total call fee}}$ |
| 3rd_ctry_share | $= \frac{\text{Total third country call fee}}{\text{total call fee}}$ |

### C. Decision Threshold

After finishing tests and adjustments, decision threshold may be 90 % of minimum score of all fraudulent users in the last month. If the system has a higher false rate compared with missing rate, the decision threshold should be increased, otherwise, it should be reduced.

### D. Evaluation Criteria

We designed two ways to evaluate the effectiveness and robustness of UAF: post-testing, and pre-testing.

*Post-Testing:* Examine whether the given fraudulent users get a higher score than normal users. Conduct Experiments on different cities and different months to ensure that UAF is applicable for different situations.

*Pre-Testing:* With continuous input of bills, users' real-time scores can be calculated simultaneously. Pre-testing focuses on the proportion of bills occupied when a fraud user is caught. The lower the rate is, the more effective UAF is.

## 4.2 Results and Analysis

### A. Post-Testing

To illustrate the performance of UAF, both normal and fraudulent users' scores of nine datasets are calculated, as shown in Table 3.

**Table 3.** Post-testing results

| Datasets | | A-1 | B-1 | C-1 | B-2 | B-3 | B-4 | B-5 | B-6 | B-7 |
|---|---|---|---|---|---|---|---|---|---|---|
| Best n | | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 |
| Ordinaries | max | 93 | 98 | 94 | 95 | 89 | 92 | 99 | 96 | 94 |
| Fraudulent Samples | No.1 | 152 | 121 | 154 | 196 | 189 | 186 | 176 | 169 | 153 |
| | No.2 | 148 | **93** | 153 | 169 | 152 | 143 | 151 | 141 | 127 |
| | No.3 | 113 | 116 | 146 | 139 | 121 | 124 | 136 | 119 | 136 |
| | No.4 | 123 | 129 | 148 | 141 | 135 | 133 | 140 | 131 | 143 |
| | No.5 | 112 | 112 | 122 | 129 | 103 | 122 | 126 | 110 | 129 |
| | No.6 | 116 | 118 | 149 | 140 | 129 | 101 | 137 | 121 | 139 |

(i) Different Cities in the Same Month

Obviously, comparing the result of A-1, B-1, and C-1, all fraudulent users get scores higher than 100 except No.2 in B-2. Since normalizing is based on normal users, the fraudulent users have specific features. That is why the fraudulent users get much higher scores.

However, there is a score lower than 100 in dataset B-2, and by sorting all scores and studying the bills, we are convinced that there is really a fraud user in B-2.

(ii) Different Months at the Same City

Analyzing 7 months' results of B city, the fraudulent users' scores are always higher than the normal users' scores, which proves that UAF works steadily through a long time.

**B. Pre-Testing**

In this part, the program of UAF reads in bills continuously simulating a data stream. Then, it calculates the usage rate of bills until fraud is detected, as show in Table 4.

**Table 4.** Pre-testing results

| Datasets | A-1 | B-1 | C-1 | B-2 | B-3 | B-4 | B-5 | B-6 | B-7 |
|---|---|---|---|---|---|---|---|---|---|
| Best n | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 |
| No.1 | 1.74 % | 2.61 % | 1.94 % | 2.63 % | 2.58 % | 2.34 % | 2.43 % | 2.21 % | 2.69 % |
| No.2 | 63.19 % | 59.36 % | 52.12 % | 58.26 % | 26.38 % | 31.64 % | 28.65 % | 20.87 % | 26.31 % |
| No.3 | 1.82 % | 1.82 % | 1.98 % | 1.32 % | 1.74 % | 1.95 % | 1.65 % | 1.98 % | 1.52 % |
| No.4 | 13.02 % | 12.50 % | 13.26 % | 10.89 % | 11.32 % | 10.98 % | 11.32 % | 11.65 % | 10.63 % |
| No.5 | 5.54 % | 8.86 % | 7.65 % | 8.12 % | 8.09 % | 8.35 % | 7.86 % | 8.32 % | 8.09 % |
| No.6 | 0.01 % | 0.01 % | 0.01 % | 0.01 % | 0.01 % | 0.01 % | 0.01 % | 0.01 % | 0.01 % |

(i) Different Cities in the Same Month

Obviously, the fraudulent users can be detected when only 0.01 % bills produced under the best condition, and for the worst, it needs 63.19 %. In this table, the usage rate will be 10.75 % on average, which means the model is robust and performs steadily for each dataset.

(ii) Different Months at the Same City

From Table 4, each fraudulent user in the 7 datasets of city B can be detected timely.

### C. Parameter n

Due to different sizes of datasets, the parameter n may affect the performance remarkably. For example, datasets A-1 has 177,761 users, where n = 10 is not big enough for distinguishing each attribute. As shown in Table 5, the No.3 fraudulent user only gets 98. When n increases to 20, the scores become more reasonable.

**Table 5.** Contrast experiment on A-1

| A-1 | | n = 10 | n = 20 |
|---|---|---|---|
| Ordinaries | max | 93 | 98 |
| Fraudulent Samples | No. 1 | 101 | 152 |
| | No. 2 | 112 | 168 |
| | No. 3 | **98** | 113 |
| | No. 4 | 103 | 123 |
| | No. 5 | 101 | 112 |
| | No. 6 | 101 | 116 |

But n is not the larger the better. To illustrate this puzzle, experiment results are shown in Table 6. There are 3 different n on B-2: 10, 20 and 30. When n is10, the minimum is 110. It rises to 131when n increases to 20, while it drops to 126 when n is 30. That is a typical example of overfitting.

**Table 6.** Contrast experiment on B-2

| B-2 | | t = 10 | t = 20 | t = 30 |
|---|---|---|---|---|
| Ordinaries | max | 100 | 95 | 93 |
| Fraudulent Samples | No. 1 | 132 | 196 | 181 |
| | No. 2 | 122 | 169 | 153 |
| | No. 3 | 118 | 139 | 128 |
| | No. 4 | 129 | 151 | 142 |
| | No. 5 | **110** | **131** | **126** |
| | No. 6 | 118 | 147 | 132 |

## 5 Conclusion

In this paper, we provide a new algorithm UAF to tackle the problem of real-time fraud detection. UAF selects feature attributes which are independent of total amount and uses equal-frequency criterion for discretization. After that, similarity calculation is proceeded by computing and comparing Manhattan distance between users. The experiments demonstrate that UAF is more precise than the state-of-the-art techniques in this domain and also has more effectiveness and scalability. In future studies, we will extend our algorithm to handle more complicated data types.

**Acknowledgements.** This paper is supported by the National Natural Science Foundation of China (61272515), and National Science & Technology Pillar Program (2015BAH03F02).

## References

1. Hoath, P.: What's new in telecoms fraud. Comput. Fraud Secur. **2**, 13–19 (1999)
2. Hoath, P.: Telecoms fraud, the gory details. Comput. Fraud Secur. **20**, 10–14 (1998)
3. Ghosh, M.: Telecoms fraud. Comput. Fraud Secur. **2010**, 14–17 (2010)
4. Rosset, S., Murad, U., Neumann, E., Idan, Y., Pinkas, G.: Discovery of fraud rules for telecommunications - challenges and solutions. In: International Conference on Management of Data and Symposium on Principles of Database Systems, pp. 409–413. ACM, New York (1999)
5. Estévez, P.A., Held, C.M., Perez, C.A.: Subscription fraud prevention in telecommunications using fuzzy rules and neural networks. Expert Syst. Appl. **31**, 337–344 (2006)
6. Panigrahi, S., Kundu, A., Sural, S., Majumdar, A.: Use of dempster-shafer theory and bayesian inferencing for fraud detection in mobile communication networks. In: Pieprzyk, J., Ghodosi, H., Dawson, E. (eds.) ACISP 2007. LNCS, vol. 4586, pp. 446–460. Springer, Heidelberg (2007)
7. Gupta, D., et al.: An analysis of telecommunication fraud using outlier detection model based on similar coefficient sum. Int. J. Soft Comput. Eng. (IJSCE) **4**, 2231–2307 (2014)
8. Cárdenas-Montes, M.: Depth-based outlier detection algorithm. In: Polycarpou, M., de Carvalho, A.C., Pan, J.-S., Woźniak, M., Quintian, H., Corchado, E. (eds.) HAIS 2014. LNCS, vol. 8480, pp. 122–132. Springer, Heidelberg (2014)
9. Hollmén, J.: User profiling and classification for fraud detection in mobile communications networks. Helsinki Univ. Technol. **29**, 31–42 (2000)
10. Zou, K., Sun, W., Hongzhi, Y.: ID3 decision tree in fraud detection application. In: 2012 International Conference on Computer Science and Electronics Engineering, pp. 399–402. IEEE Press, Hangzhou (2012)
11. Subudhia, S., Panigrahib, S.: Quarter-sphere support vector machine for fraud detection in mobile telecommunication networks. Procedia Comput. Sci. **48**, 353–359 (2015)
12. Moreau, Y., Verrelst, H., Vandewalle, J.: Detection of mobile phone fraud using supervised neural networks: a first prototype. In: Gerstner, W., Hasler, M., Germond, A., Nicoud, J.-D. (eds.) ICANN 1997. LNCS, vol. 1327, pp. 1065–1070. Springer, Heidelberg (1997)
13. Burge, P., Shawe-Taylor, J.: An unsupervised neural network approach to profiling the behavior of mobile phone users for use in fraud detection. J. Parallel Distrib. Comput. **61**, 915–925 (2001)
14. Hilas, C.S., Mastorocostas, P.A.: An application of supervised and unsupervised learning approaches to telecommunications fraud detection. Knowl. Based Syst. **21**, 721–726 (2008)
15. Yufeng, K., Lu, C.-T., Sirwongwattana, S., Huang, Y.-P.: Survey of fraud detection techniques. In: 2004 IEEE International Conference on Networking. Sensing and Control, pp. 749–754. IEEE Press, Taiwan (2004)