Conference Proceeding Issue Published in International Journal of Trend in Research and Development (IJTRD), ISSN: 2394-9333, www.ijtrd.com

A Novel Machine Learning Algorithm for Spammer Identification in Industrial Mobile Cloud Computing

[1]T. Revathi, (M.Sc) and [2]Prof J. Joselyn M.C.A, M.Phil,
[1,2]Department of Computer Science, Kamban College of Arts and Science, For Women, Tiruvannamalai, India

Abstract: An industrial mobile network is crucial for industrial production in the Internet of Things. It guarantees the normal function of machines and the normalization of industrial production. However, this characteristic can be utilized by spammers to attack others and influence industrial production. Users who only share spam's, such as links to viruses and advertisements, are called spammers. With the growth of mobile network membership, spammers have organized into groups for the purpose of benefit maximization, which has caused confusion and heavy losses to industrial production. It is difficult to distinguish spammers from normal users owing to the characteristics of multidimensional data. To address this problem, this paper proposes a Spammer Identification scheme based on Gaussian Mixture Model (SIGMM) that utilizes machine learning for industrial mobile networks. It provides intelligent identification of spammers without relying on flexible and unreliable relationships. SIGMM combines the presentation of data, where each user node is classified into one class in the construction process of the model. We validate SIGMM by comparing it with the reality mining algorithm and hybrid FCM clustering algorithm using a mobile network dataset from a cloud server. Simulation results show that SIGMM outperforms these previous schemes in terms of recall, precision, and time complexity

I. INTRODUCTION

The Internet of Things (IoT) [1] is an important component of the new generation of information technology. It is widely used in many fields such as industrial control, cyber-physical systems, and military investigation through the techniques of intelligent perception, identification technology, and pervasive computing [2]. To understand and measure the environment through objects' inter-connections around people is the basic idea of IoT [3], its foundation is the internet and terminals to provide communication between objects [4]. It connects humans and objects, objects with objects, provides remote control, and controls intelligent networks in new ways through enabling technologies [5]. Simulations are performed to present SIGMM's performance in identifying spammers and we compare SIGMM with the reality mining algorithm (RMA) and hybrid FCM clustering algorithm (HFCM) in Section V. We implement spammer identification on the industrial mobile data to verify our proposed algorithm. Finally, the conclusions and future work are presented in Section VI.

II. RELATED WORK

For existing algorithms, there are three types of machine learning: supervised learning, unsupervised learning, and reinforcement learning.

1) Supervised Learning*:* The main goal of supervised learning is to learn a model from labeled training data that allows us to make predictions about unseen or future data [9]. Supervised refers to a set of samples where the desired output labels are already known [10]. In the Spammer Setection algorithm based on Logistic Regression [11], a spammer classifier is built for an online network with some features as inputs, and the algorithm output is 1 if a spammer is suspected. The model is trained on a large training set, however, collection of labeled data is rather difficult because of the recent emphasis on the secrecy of user data.

2) Unsupervised Learning: Using unsupervised learning techniques, we are able to explore the structure of our data to extract meaningful information without the guidance of a known outcome variable or reward function [12]. A clustering algorithm is the main algorithm for unsupervised learning [13]. Clustering is a technique that allows us to find groups of similar members [14]. In the RMA based on K-means in [15], the algorithm proposes a silhouette function which accepts the number of clusters as a parameter to judge the accuracy of clustering. Then it uses a matrix of means to record the mean silhouette values for each value of *k* and finally determines the best value of *k*. But the clustering result depends on the *k* centroids. Therefore it must consume extra time to determine the value of *k*. Furthermore, experimental results are unstable, with the same *k* used in several experiments, producing different results. In [16], a prediction model based on Big Data analysis using a hybrid FCM clustering algorithm (HFCM) is proposed. It works by repeating arithmetic operations to minimize the objective function and updating membership function, which is very time-consuming. The experimental performance depends on the volume of the dataset being small or its accuracy will be sharply reduced.

3) Reinforcement learning*:* Reinforcement learning algorithms are very suitable for learning to control an agent by allowing it to interact with an environment [17]. The goal is to choose an action to maximize an expected long-term reward. In [18], a recursive least squares (RLS) algorithm based on the reinforcement algorithm is proposed. It applies Q-Learning by choosing a policy which is the best selection for a specific user. But it starts from a random user and does the exploration within the network by friendship relationships, which restricts the scope of the exploration, and leads to decreased detection efficiency.

III. PRELIMINARIES

In order to learn the data construction and rules, and owing to the limited access to original data, we preprocess the data after extracting any available original data in an industrial mobile network.

A. Data description

Our data contain the following contents, user's ID, the relationship with other users, the time-stamped post record, and the activity in the past three months. From the post record, we calculate the frequency of posting and proportion of posts with URL or @, and the average similarity among the user's posts. The activity reflects whether the account is normal or not. It indicates the frequency of following others, which is necessary because spammers tend to follow others all the time.

B. Feature scaling

The data we obtained have the following two constraints. First, the labeled data are far fewer than the unlabeled data which severely decreases the precision of training. Second, there is large data noise that may cause incorrect factors in the parameters of the model. Data points that do not belong to any class are defined as data noise. The values of some data may greatly differ from the mean of samples. SIGMM reduces data noise by calculating the similarity among users to increase the precision of training.

We arbitrarily extend the modulus of vectors, Euclidean distance is sensitive to differences, while cosine distance does not detect any change. Therefore, the method based on Euclidean distance is more appropriate for our purpose.

C. Feature grouping

By utilizing standardization and the Pearson correlation coefficient, the features remain simple. The multidimensional feature is divided into three parts to indicate three user perspectives, which are basic features, content features and the basic features include the number of fans, following users, posts, and the frequency of following. Previous studies show that spammers tend to follow a large number of users, and their fans are rare. The proportion of fans to following is particularly low.

- These characteristics reflect whether the user is normal or not. Spammers' frequency in following others is further higher than that of normal users.
- The content features mainly reflect the characteristics of the information sent by a user in the most recent three months. The user's activity can be analyzed by the content features.
- The network feature mainly describes user characteristics in an industrial mobile network. The number and proportion of following each other represent the degree of intimacy between users. Spammers usually follow a large number of normal users to attack. Therefore, their proportion of following each other is lower than that of normal users.

IV. SIGMM MECHANISM

The SIGMM mechanism fits the behavior data of normal users and spammers, where the behavior data of normal users and spammers are mixed random sampling. The SIGMM mechanism learns the parameters of the two distributions (normal users and spammers) to obtain the classification model.

A. Parameters estimation based on Expectation-maximization

The data are approximately subject to the Gaussian distribution. The mean and variance must be estimated for initializing the model. According to the probability density $p(x|\theta)$, we independently extract some samples to constitute the training sample set X. Parameter θ represents the mean and variance of the dataset, and is estimated through the sample set X. Consider $X = \{x1,x2,...,xn\}$ as a set of extracted samples, xi represents the i th user data, and n represents the number of samples. Because they are independent, the probability that xi and xj are extracted simultaneously is $p(xi|\theta)* p(xj|\theta)$. Similarly, the probability that n samples are extracted simultaneously is the product of their respective probabilities, as shown in Eq. $L(\theta)$ is called the likelihood function related to the sample set X and parameter θ.

B. Analysis of Gaussian Mixture Model in dataset

There is another unknown variable z belonging to the class xi in the likelihood function. Our goal is to find the proper θ and z that maximize the value of $L(\theta)$. Eq. (10) defines the likelihood function with the variable z.

The parameters can be estimated according to the data with the same Gaussian distribution. The description of each sample is represented by a triple yi. $yi = \{xi,zi1,zi2\}$, where xi is the i th sample, and $zi1$ and $zi2$ indicate which Gaussian distribution produces xi. They indicate whether the user is normal or not. For example, if xi is equal to 1.8, and he belongs to the Gaussian distribution of a normal user, then we can describe the sample as $\{1.8,1,0\}$. If the values of $zi1$ and $zi2$ are known, the parameters can then be estimated by the maximum likelihood algorithm. The process below describes how to calculate $zi1$ and $zi2$ based on labeled samples.

1. Initialize distribution parameter θ and repeat the following steps until θ converges.
2. Calculate the posterior probability of the variable z according to the initial parameters or the parameters from the last iteration. $Qi(zi)$ stands for the expectation of the implicit variable z.
3. Implementation on Semi-supervised learning

Semi-supervised learning [22] is suitable for data with few labels. Therefore, how to use a large number of unlabeled samples to improve learning performance has become one of the most important issues in current machine learning research. The semi-supervised learning algorithm takes full advantage of labeled data [23, 24]. It is a training process with an initial model constructed from a small group of labeled data. It continues by predicting unlabeled data first, and then transfers the data into the labeled dataset. The parameters of the model are updated and optimized until the model reaches a stable and optimal state.

According to the judgment standard based on the EM algorithm, the proportion of normal users to spammers is required as a prior knowledge. It is defined by Eq. (12).The proportion cannot be estimated due to the large number of unlabeled data. πk is an uncertain variable, therefore the value of $r(i,k)$ cannot be calculated. To solve the problem, two solutions are proposed. First, roughly calculate πk according to the samples to obtain the probability of each class. Second, through data visualization we note that the unlabeled data points are distributed in the medial and lateral of two ellipsoids. Each ellipsoid is a distribution. Thus, the determination of class depends on the distance from a data point to the ellipsoid surface. Compared with the first method, the second method is more convenient to calculate. We conducted the following experiment to identify which one performs better. Fig.4 illustrates the precision of the two methods with a random selection of 1,000 labeled data. The X-axis represents the number of iterations, and the Y-axis is the precision of the two methods.

Algorithm 2 Predicting process for unlabeled data

It can be observed from Fig. 4, that the pink line which represents the distance method fluctuates because of the unstable feature at the beginning of the iteration. Although the probability method is slightly higher than the distance method it exhibits little change in the later iterations, and is lower than the distance method at the end of the iterations. When the probability $r(i,k)$ is calculated, the roughly estimated πk does not represent the proportion of each distribution in the whole dataset owing to the large amount of unlabeled data, which decreases precision.

We choose the distance solution which calculates the distance between data points and the two ellipsoids as our improved judgment standard. This prediction process is detailed in Algorithm 2.

struct two ellipsoids (Lines 2 and 3). Then we define two perpendicular lines to the two ellipsoids from the λ point, and construct tangent planes on the ellipsoids (Lines 4 and 5). Lines 6 to 8 describe the relationships of points and tangent planes. Next, we calculate the Euclidean distance between each point and ellipsoid with the *linalg.norm* () function in the *Scipy* package (Lines 9 and 10). Next, we choose the closer distance to the model as the classification result (Line 11). Finally, the prediction of result is returned. The complexity of Algorithm 2 is O(1).

If a node has been classified as one of the two classes, it will be removed from the unlabeled dataset and joins the training set. The node that joined most recently might cause the model to adjust parameters. Through the entire iteration process, the parameters of the two models are gradually adjusted to the optimal result. The process of training and prediction.

The above training process avoids relying solely on the labeled dataset. When new data are added to the training set, the parameters are adjusted at each iteration. The final parameter will gradually converge to a fixed range. Take one group of features as an example, where the changing trend of parameters can be observed in Fig. 6. The mean of this one-dimensional feature is a component of the ellipsoidal coordinates. The variance is the radius of the ellipsoid. The figure shows the variation of the parameter over 100 iterations.

It can be seen from Fig. 6 that both the center coordinate and the radius change significantly at the beginning of the iterations. With new data joining, the change gradually decreases and eventually converges to a fixed range.

D. Classification of SIGMM Model

A Gaussian mixture model is a probabilistic model for statistical learning [25]. Through the estimation of the probability density distribution of samples, each Gaussian model represents a class. By matching samples with several Gaussian models to obtain probabilities, the class with the largest probability is chosen as the classification result.

With continuous iteration, the ellipsoid’s center as well as radius change slowly as shown in Fig. 7(a) and converge to the stable position. The following conclusions can be obtained from the figure. (1) The data of the two classifications are clearly separated. (2) The radius of the green one representing normal users is larger than that of the red one, because the number ofnormal users is large. Their behavior data are not similar to each other and deviate from the center. The radius of the red one is smaller than that of the green one because spammer behaviors for attacking others are similar. (3) The radius and location of the two ellipsoids vary only slightly.

V. SIMULATION RESULTS

In this section, we perform simulations using SIGMM. The stable parameters are confirmed through simulation experiments. Taking the accuracy, recall, and precision as metrics, we compare SIGMM with RMA and HFCM. In addition, the performances of the three schemes are compared in terms of recall, which represents the percentage of real spammers identified.

A. Simulation Setup

The number of iterations is set to 100. The comparison experiment is to divide the labeled data into two parts, one part is to generate the initial values, and the other is the test set. All the data are preprocessed identically.

Precision, accuracy, and recall are the main basis for judging performance. SIGMM focuses on identifying spammers from normal users, and the accuracy represents the proportion of all the spammers predicted by the model. Recall represents the proportion of real spammers that have been correctly identified. Precision is the proportion of correct numbers in synthetic forecasts. We first compare the three solutions from these three aspects, then replace the semi-supervised training process with supervised training, and compare the two schemes in terms of the three measures.

B. Time complexity

We can see that the recall of the SIGMM model is slightly higher compared with the other two schemes. Moreover, HFCM and RMA are influenced because of the initial values of the algorithms and the two poly lines have a certain degree of fluctuation.

When the size of the data is very large, the complexity of an algorithm is also an important metric to judge its performance. Thus, the three schemes are compared in terms of time complexity. The SIGMM model is established by the process of prediction and updating. The number of user samples is n, and the labeled training set sample is approximately $0.4n$. The labeled set is randomly divided into two parts. Each is approximately $0.2n$.

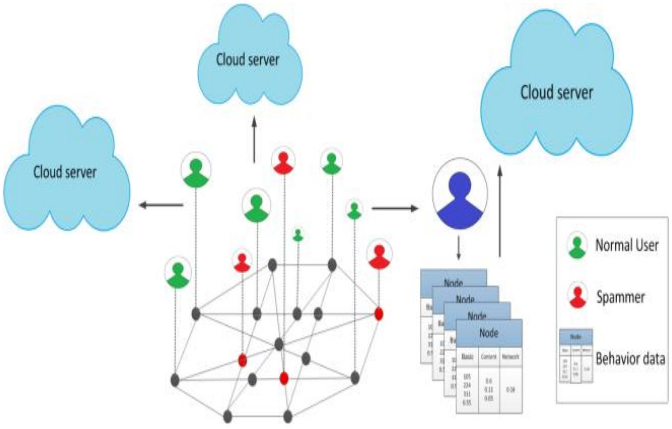
Therefore, the complexity of the algorithm is determined by the following factors: (1) it can be concluded from experiments that the initial values have significant influence on the accuracy of the model. We iterate k times to select the best model and calculate the initial value O(1). The complexity of testing is O(n). The total complexity is then approximately O(n). (2) The calculation of each sample’s distance to the two distributions is approximately O(1). Adjust the parameters, for a total $0.6n$ of data indicates a complexity of O(n). So theoretically, the overall complexity is approximately O(n).

The algorithm complexity of HFCM comes from the establishment of the fuzzy matrix and updating of membership. For n samples, d dimension features, k iterations, and C clusters, the complexity is O($n2k$).

Similarly, the complexity of RMA comes from computing the mean of all dimensions, cluster centers, and the samples in each cluster. The complexity is then approximately O(n). Fig. 8(c) shows a comparison of the time complexity for the three schemes with a dataset of 1,000 users.

The X-axis represents the amount of data in Fig. 8(c), and the Y-axis represents running time. With the increase of the data, the difference in running times among the three algorithms gradually increases, and the SIGMM model is significantly better than the other two for larger datasets in terms of running time.

VI. SYSTEM ARCHITECTURE



CONCLUSION

In order to solve the malicious attack problem in industrial mobile networks and reduce the computational complexity of using large cloud server datasets, this paper proposes SIGMM, a spammer identification model based on the Gaussian Mixture Model. We extract features related to labels from originally labeled data in a given dataset containing both labeled and unlabeled data, and visualize the data to add labels to the unlabeled data.

According to the characteristics of data presentation, each user data belongs to one distribution. Multidimensional features are divided into three groups, and SIGMM separates the two distributions based on these features. Finally, we performed simulations to evaluate the performance of SIGMM. The results show that even if the relationships among users are not taken into account, it can implement classification.

Our work is based on binary classification, whereas in large networks, the types of users are varied and complex. Our future work will extend the categories of users to multi-classifications such as celebrity, advertiser, hacker, etc.

References

[1] J. Miranda, N. Makitalo, J. Garcia-Alonso, J. Berrocal, T. Mikkonen, C. Canal, and J. M. Murillo, “From the internet of things to the internet of people,” *IEEE Internet Computing*, vol. 19, no. 2, pp. 40–47, 2015.

[2] T. Qiu, A. Zhao, F. Xia, W. Si, and D. O. Wu, “Rose: Robustness strategy for scale-free wireless sensor networks,” *IEEE/ACM Transactions on Networking*, vol. 25, no. 5, pp. 2944–2959, 2017.

[3] L. Yao, Q. Z. Sheng, and S. Dustdar, “Web-based management of the internet of things,” *IEEE Internet Computing*, vol. 19, no. 4, pp. 60–67, 2015.

[4] T. Qiu, R. Qiao, and D. O. Wu, “Eabs: An event-aware backpressure Scheduling scheme for emergency internet of things,” *IEEE Transactions on Mobile Computing*, vol. 17, no. 1, pp. 72–84, 2017.

[5] T. Qiu, K. Zheng, H. Song, M. Han, and B. Kantarci, “A local-optimization emergency scheduling scheme with self-recovery for smart grid,” *IEEE Transactions on Industrial Informatics*, vol. 13, no. 6, pp. 3195–3205, 2017.

[6] S. Lu, V. H. Nascimento, J. Sun, and Z. Wang, “Sparsityaware adaptive link combination approach over distributed networks,” *Electronics Letters*, vol. 50, no. 18, pp. 1285–1287, 2014.

[7] E. Tan, L. Guo, S. Chen, X. Zhang, and Y. Zhao, “Spammer behavior analysis and detection in user generated content on social networks,” in *IEEE International Conference on Distributed Computing Systems*, May. 1618, 2012, pp. 305–314.

[8] P. Heymann, G. Koutrika, and H. Garcia-Molina, “Fighting spam on social web sites,” *IEEE Internet Computing*, vol. 11, no. 6, pp. 36–45, 2007.

[9] M. Al Hasan, V. Chaoji, S. Salem, and M. Zaki, “Link prediction using supervised learning,” in *Proc of Sdm Workshop on Link Analysis Counterterrorism and Security*, Apr. 26-28, 2006, pp. 798–805.