Predicting Malicious Users on Anonymous Chat Networks

Tsung-Chuan Chen, Chieh Ho and Augustus Hong, Stanford University

Abstract— **Malicious users on chat network systems would reduce the willingness of benevolent users to chat on the same network. Thus it is often desirable to classify malicious users based on their personal profile and chat contents. In this study, different algorithms including Naive Bayes, SVM, Decision Table, Multilayer Perceptron, and Logistic classification are applied on the dataset from an anonymous chat network Chatous. It is found that by using both empirical features and chat word features together with the SVM algorithm, the best F-score achieved is 84.9%. This result shows the possibility of classifying malicious users from benevolent users on a chat network and may lead to chat quality improvement such as by restricting malicious user from chatting with benevolent users.**

***Keywords*—Malicious users detection, Word feature extraction, Support Vector Machine, Machine learning, Chat Network.**

I. Introduction

Chatous is a text-based, 1-on-1 anonymous chat network that has seen 2.5 million unique visitors from over 180 different countries. Users can create a profile that contains a screen name, age, gender, location, and a short free-form "about me" field. Interactions on Chatous include exchanging messages, sending/accepting a friend request, reporting an abusive user, ending a conversation. Since the network is consisted of completely anonymous users, it is therefore important to be able to distinguish between different kinds of users.

Some users come to Chatous to find friends and have quality conversations, while others might try to use the platform in a way that violates its policies. Malicious users might verbally harass other users, or try to steal other users' personal information, and thus identifying such users can greatly improve the user experience of the chat network. Machine learning techniques will be used to classify benevolent and malicious users based on the data set that is provided by Chatous. And with the developed models, malicious users can be identified based on his/her chat contents, age, conversation length, and many other attributes.

II. Data Sets

The dataset provided comes in two parts. The first part is 9 million conversations and the second part is 1.2 million profiles. Preprocessing has been done on these input files to extract features. And the attributes of the datasets are listed in Table I.

But since there is no actual answer to who the malicious users are, different methods are used in labeling benevolent and malicious users. One point to note in this dataset is that there are two different base entities, one is the user ID and the other one is profile ID. Each user can have multiple profiles to chat with different people, while they can only have one user account, say, user ID. It is found that the accuracy of detecting against bad profiles is significantly higher than detecting against bad users.

The approaches taken to label malicious and benevolent users include: (1) Label all user ID which has been reported once as malicious users. (2) Label all profile ID which has been reported once as malicious profiles. (3) Label all profile ID which has been reported more that the average number of reports as malicious profiles.

TABLE I
DATASETS

Chat Data	User Profile Data
Chat ID	Profile ID
Profile ID	Location
Timestamp	Profile introduction
User ID	Age
Reported ID	Gender
Disconnected person	
Word Vector	

III. Approaches

A. Features Selection Methods

The provided data sets are as described in II. And two set of features will be selected using the empirical features selection method and the word vector processing features.

(1) Empirical feature selection

Several features are obtained from the two databases, profile and chat heuristically. And the selected features and how they are selected are both described in Table II.

(2) Word Vector Processing Features

To get a deeper understanding about the difference of specific words used between malicious and benevolent users, the word vectors of all chats are examined. In particular the word frequency method, which is similar to the document frequency (DF) method [3], is used to extract this feature.

Word frequency is the number of tokens that occurs in all chats. Specifically in this method, all stop words are excluded in the word vector since they won't provide meaningful information. Then it counts the frequency of occurrence of every word. And the basic assumption is that the fewer times

TABLE II
EMPIRICAL FEATURES

Features	Descriptions
Age	Raw data given from profile dataset
Gender	Raw data given from profile dataset
Profile length	The length of "about me" section
Number of chats	The total number of chats of the user
Total word vector length	The total words that the user has typed throughout all conversations
Total disconnected number	The total number that the user actively disconnected the chats
Total line number	The total lines that the user has typed throughout all chats
Total min	The total minutes of all chats of the user
Percentage of disconnected	The percentage of chat that is actively disconnected by the user
Average lines per chat	The average of lines of a chat of the user
Average words per minute	The average words per minute of the user
Average words per chat	The average of words of a chat of the user
Different sex percent	The percentage of chat with different gender user
Naive Bayes	Prediction by Naive Bayes
Word real length	The total distinct words that the user has typed throughout all conversations
Percentage of the most frequent word	The times the most frequent word occurs divided by total words
Difference of frequent word	The difference between the most frequent word and the second frequent word

the word occurs, the fewer information of classification the word could offer. In the subsequent experiments, the top 100, 300, 500 frequent words are selected as the features respectively.

B. Classification Methods

(1) Support Vector Machine

Support Vector Machine (SVM) is known as a non-linear classification method by using kernels to map data with finite dimension of features to higher-dimensional feature space. And the primal optimization problem of an SVM can be modeled as:

$$min_{\gamma,w,b}\ \frac{1}{2}\left|\left|w\right|\right|^2 + C\sum_{i=1}^{m}\xi_i \qquad (1)$$

$$s.t.\ y^{(i)}\ (w^T\ x^{(i)} + b)\ \geq\ 1\ -\ \xi_i\ ,$$

, where C is the soft margin parameter. The parameter C can also be a regularization term, which provides a way to control over-fitting: as C becomes large, it must respect the data so as to reduce the cost of reducing the geometric margin; when it becomes small, it is easy to account for some data points with the use of slack variables and to have a fat margin placed so it models the bulk of the data.

There are generally three types of kernels, which are the linear kernels, polynomial kernels and the Gaussian kernels. In this study, the Gaussian kernels are chosen for the malicious users classification problem. The Gaussian kernel is defined as:

$$K(x,z) = exp(-\gamma|x-z|_2^2) \qquad (2)$$

,where γ is the bandwidth of the kernel. By using the Gaussian kernel, the features are mapped to infinite feature dimensions space. The value of gamma should be carefully tuned. If overestimated, the exponential will behave almost linearly and the higher-dimensional projection will start to lose its non-linear power. On the other hand, if underestimated, the function will lack regularization and the decision boundary will be highly sensitive to noise in training data. And the LIBSVM library [1] is used for implementing SVM in this study.

(2) Naïve Bayes

In addition to SVM, malicious and benevolent profiles/users are also classified with the Naive Bayes event model with Laplace Smoothing. With the Naive Bayes model as the baseline, one will be able to see how well other algorithms are doing. The word vectors in the conversations are used as classification features. And the models are trained with our own implementation of Naive Bayes algorithm in Python, as well as using the off-the-shelf implementation in Weka [2]. Both models have converged to similar result.

(3) Multilayer Perceptron [4]

A multilayer perceptron (MLP) is a feed forward artificial neural network model that maps sets of input data onto a set of appropriate outputs.

(4) Decision Table [5]

Decision tables, like decision trees or neural nets, are classification models used for prediction. They are induced by machine learning algorithms. A decision table consists of a hierarchical table in which each entry in a higher level table gets broken down by the values of a pair of additional attributes to form another table.

(5) Logistic Regression

Logistic Regression is an algorithm that utilizes the logistic function and a Bernoulli random variable to classify input samples.

C. Experimental Setup

The system block diagram of this study is shown in Fig. 1. After labelling the data, the performance of different feature selection methods and learning algorithms will be evaluated.

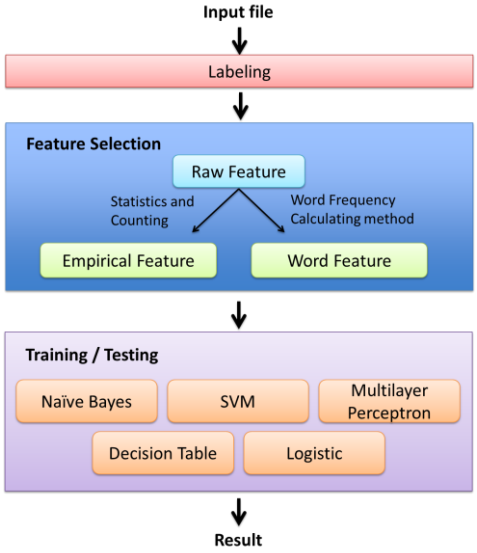

Fig. 1 The System Block Diagram

IV. Results

For all experiments, F-score is chosen to assess the performance of different learning algorithms and feature sets, where F-score is defined as:

$$Fscore = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (3)$$

F-score is preferred to the accuracy since it is equally important in this problem to have high recall and precision rates. Also, the F-score are examined using cross-validation method with the fold number being 10 in all experiments.

A. *Comparison of Different Labeling Approaches*

One of the ambiguities in this data set is that there is no gold standard for the labeling of data. Therefore a few labeling methodologies are proposed to address this issue. First, as suggested by the provider of this dataset, the report count could be an efficient indicator for labeling. Users who were reported often by other users are more likely to be malicious users. Second, on the Chatous Network, one user may own more than one profile and can use different profiles to chat with other users. And thus the reported times can be counted either with respect to the users or to the profiles. Therefore, the performances of setting different thresholds for the report count based on either users or profiles are evaluated. Specifically, there are three different labeling settings in this experiment, which are reported at least once based on user ID, reported at least once based on profile ID and reported more than the average times based on the profile ID, where the average reported times is approximately 1.87. And since this experiment focuses on the comparison of different labeling approaches, only the SVM algorithm with empirical features is used in this study.

The results are as shown in Fig.1. From the figure, the profile-based labeling is better than the user-based labeling by 1.3%. This improvement might result from that users may have different behaviors when chatting with different profiles. If the labeling method is user-based, a lot malicious profiles may be counted that don't perform malicious behavior but still be marked as malicious profile just because the user has one other profile classified to be malicious one. Hence, it may be more accurate if determining with respect to profiles. On the other hand, the F-score improves 9% when the threshold of reported time based on profiles is set to more than the average reported times (approximately 1.89). This significant improvement may due to the reason that when setting the threshold to be larger than the average reported count, the likelihood of a labeled user being an actual malicious user increase. Nevertheless, it should also be noted that if the threshold for reported times is set too high, some malicious users may not be identified correctly.

As a consequence of this result, the malicious users will be defined as profiles with reported times larger than the average reported times (1.87) in subsequent experiments.

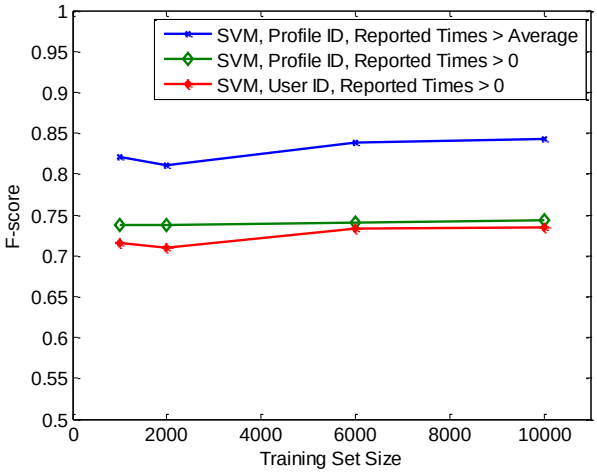

Fig. 2 The result of using different labeling approaches

B. *Naïve Bayes and SVM Classification using Word Features*

In this experiment, Naive Bayes and SVM classification are implemented to classify users by their chat contents. Nevertheless, since the total number of words in chats approaches several tens of thousands, learning with all the words as features would be inefficient. As a result, feature selection is needed to reduce the total feature number. For both the Naive Bayes and the SVM method, the word frequency (WF) method is implemented to select a subset of word vectors, which appear more frequently, as the features.

(1) Naive Bayes Chat classification

Naive Bayes text classification is implemented to classify the word features of chats in order to determine whether the user is malicious. The features number are set to be 100, 300 and 500 using WF method for feature selection. And the training set sizes for each feature numbers are 1000, 2000, 4000 and 6000 respectively.

The results of the Naive Bayes Chat classification are shown in Fig 2. It can be seen from the figure the best achieved f-score is 67.9% with feature number equal to 500 and with 6000 training sets. And the F-score is affected by changing the training set sizes. However, the total numbers of features have limited effects on the results from the figure.

(2) SVM Chat classification

SVM classification is implemented to classify the word features of chats in order to determine whether the user is malicious. The features number is set to be 100, 300 and 500 using the word frequency method. And the training set size for each feature numbers are 1000, 2000, 4000 and 6000 respectively.

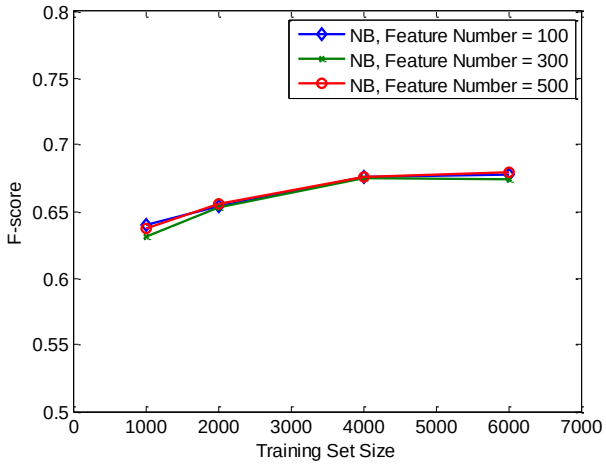

Fig. 3 Naive Bayes Classification by using Word Features

The result of SVM text classification is shown in Fig 3. The best f-score is 75.8% when the feature number is 500, which is 7.9% higher than the best result of Naive Bayes. It can be seen from Fig 3 that the F-score for all training set size ranges from 65% to 75.8%. As opposed to the results of using Naive Bayes, the F-score increases as the feature number becomes larger. This difference may be due to that the SVM has mapped the original feature space to higher dimension.

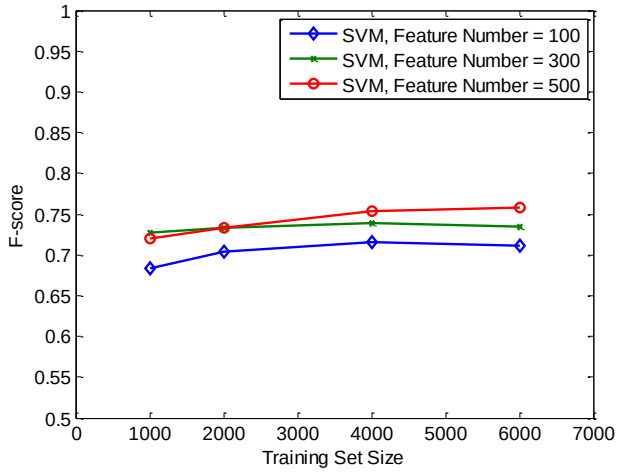

Fig. 4 SVM Chat Classification with Selected Features

C. Different Learning Algorithms on Empirical Features

Besides using the word vectors as features, empirical features are calculated as described in II.B. Additionally, different learning algorithms are implemented with these 17 empirical features. The learning algorithms include Naive Bayes, Decision Table, Multilayer Perceptron, and Logistic Classification. And it is found that the SVM algorithm has the best performance (84.6%) for this learning problem. The performance of the Decision Table algorithm, which is 84.1%, is also comparable to that of the SVM algorithm. Despite that the result of the Naive Bayes algorithm decreases to be lower than 70%, the SVM, Multilayer Perceptron, Decision Table and Logistic algorithms all have F-score larger than 80%, which are better than the result of classification using the chat contents as in IV.A.

D. Different Learning Algorithms on Empirical Features combined with Word Features

In this experiment, the empirical features and the word features are combined together for user classification. It is found that the performance for the combination of two kinds of features is better than using either of the two kinds of

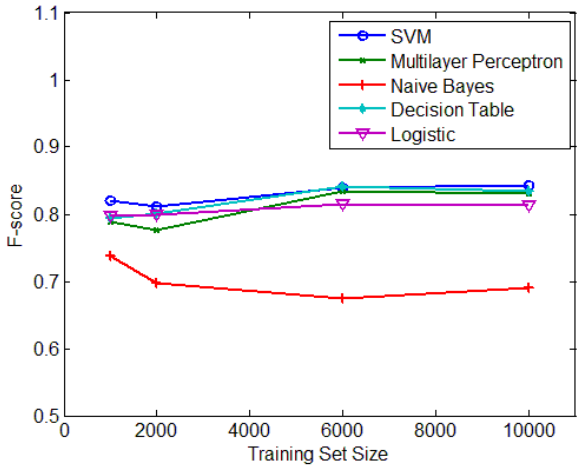

Fig. 5 Result of Different Learning Approaches

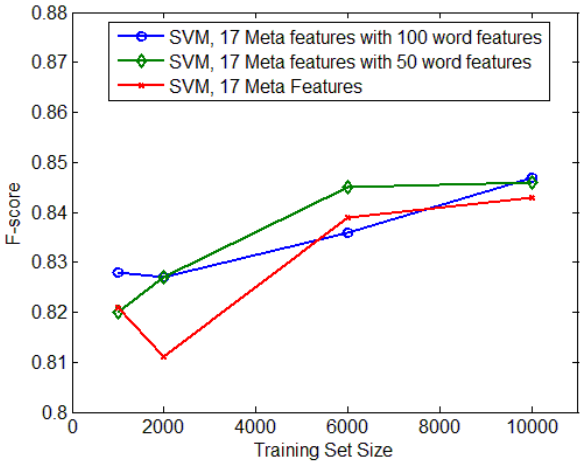

Fig. 6 Result of combination of Empirical Features and Word Features

features alone. The best performance it can achieve is 84.9% with training set size being 10,000. However, the difference of adding the word features to the empirical features is slight. One possible reason for this result is that the empirical features have already contained similar information in added word features.

The summaries of the all the experiment results are shown in Table 3.

TABLE III
SUMMARY OF RESULTS

Method	Feature Sets	F-score (%)
Naïve Bayes	500 word features	67.8
SVM	500 word features	75.8
Naïve Bayes	17 empirical features	69.1
Logistic	17 empirical features	81.3
Multilayer Perceptron	17 empirical features	83.1
Decision Table	17 empirical features	84.1
SVM	17 empirical features	84.6
SVM	17 empirical features and 100 word features	84.9

V. Discussion

The labeling method used for malicious users depend on the reported counts of users. The reported count is thought to be a good measure to separate different kinds of users because it is a measure of how malicious a user can be. Even if the reported count doesn't directly reflect who the malicious users are, it is still considered reasonable to separate those who are more prone to be reported with those who are not. The significant improvement in F-score when the labeling method is switched from one report count threshold to average report count, also shows that the selected features have a high correlation to the reported counts. One way to improve the accuracy of classifying malicious users is increase the report threshold. But if the report threshold is increased too much, actual malicious users may be omitted when during classification.

One possible reason why the chat content classification is not effective may result from that the word features are sparse. That is, not all the chosen word features would occur in the chats of users. Additionally, natural language processing algorithms are not yet used to investigate the relationship between word vectors. Only the statistics of word features is taken into consideration. Thus, the performance of chat content classification may be limited. Another reason that this approach is ineffective might be due to that a lot of profiles in the given dataset don't even have a chat with others. Thus when classifying those profiles word vectors wouldn't supply enough information.

In the study, it is found that the empirical features are very effective even without additional word features. This may due to that the empirical features selected have already contained the information of the word vectors. The other possible reason may be that it's already become a high variance problem, so that adding more features may not improve the result. One possible solution to this problem would be trying to increase the training set sizes.

There are many possible variations of this study that can be explored to seek performance improvement. Natural language processing algorithms could have been applied on the word vectors to extract more meaningful features. Also treating the profiles/users as nodes in a social graph might provide more insight to the relationship between benevolent and malicious users.

VI. Conclusion

In this study, different algorithms and features sets are evaluated for the classification problem of the users in the Chatous Network. It is found that the empirical features combined with word features selected by the WF method with SVM learning method can give the best prediction result for Chatous Network.

Since current Chatous Network system randomly pair the users for chat, one possible application of this result may be using the method to classify malicious users and benevolent users and pair only the benevolent users together to improve the chats qualities. And this approach may be used by other random chat systems as well.

VII. Acknowledgement

We want to thank Kevin Kuo from Chatous for providing us the dataset as well as giving us guidance on how to interpret the data.

VIII. Reference

[1] Chih-Chung Chang and Chih-Jen Lin, LIBSVM : a library for support vector machines. ACM Transactions on Intelligent Systems and Technology, 2:27:1--27:27, 2011.

[2] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, Ian H. Witten (2009); The WEKA Data Mining Software: An Update; SIGKDD Explorations, Volume 11, Issue 1.

[3] Yang, Yiming, and Jan O. Pedersen. "A comparative study on feature selection in text categorization." *ICML*. Vol. 97. 1997.

[4] Gardner, M. W., and S. R. Dorling. "Artificial neural networks (the multilayer perceptron)--a review of applications in the atmospheric sciences." *Atmospheric environment* 32.14-15 (1998): 2627-2636.

[5] Becker, B. "Research report: Visualizing decision table classifiers." *Information Visualization* 98 (1998).