DeepScan: Exploiting Deep Learning for Malicious Account Detection in Location-Based Social Networks

Qingyuan Gong, Yang Chen, Xinlei He, Zhou Zhuang, Tianyi Wang, Hong Huang, Xin Wang and Xiaoming Fu

***Abstract*—The widespread location-based social networks (LBSNs) have immersed into our daily life. As an open platform, LBSNs typically allow all kinds of users to register accounts. Malicious attackers can easily join and post misleading information, often with the intention of influencing the users' decision in urban computing environments. To provide reliable information and improve the experience for legitimate users, we design and implement *DeepScan*, a malicious account detection system for LBSNs. Different from existing approaches, DeepScan leverages emerging deep learning technologies to learn users' dynamic behavior. In particular, we introduce the long short-term memory (LSTM) neural network to conduct time series analysis of user activities. DeepScan combines newly introduced time series features and a set of conventional features extracted from user activities, and exploits a supervised machine learning-based model for detection. Using the real traces collected from Dianping, a representative LBSN, we demonstrate that DeepScan can achieve an excellent prediction performance with an F1-score of 0.964. We also find that the time series features play a critical role in the detection system.**

***Index Terms*—Location-Based Social Networks, Malicious Account Detection, Deep Learning, Long Short-Term Memory Network**

I. Introduction

Emerging location-based social networks (LBSNs), such as Foursquare, Yelp and Dianping, have become a widely-used tool for people's daily life. Different from traditional online social networks (OSNs), LBSNs have a special focus on location-centric activities. Thanks to the popularity of smartphones or tablets equipped with GPS, a user can show her presence at a certain place by conducting a check-in in LBSNs. She can also post reviews for a selected venue. Each review or check-in must be associated with a point-of-interest (POI). These massive user-generated contents (UGCs) can help users make different location- and context-aware decisions, e.g., dinning, traveling or purchasing. Meanwhile, these apps record rich spatio-temporal information of massive users, which can be used for understanding human behaviors in the context of urban computing. Prospective applications include life style modeling and analysis [13], city dynamics studying [8] and venue recommendation [15]. Researchers typically use the rich spatio-temporal data collected from different LBSN platforms such as Foursquare, Dianping and Jiepang, to investigate these urban computing applications.

Due to the open nature of the LBSN platforms, attackers can register accounts and perform malicious attacks for various purposes. For example, in Foursquare, users might conduct extraneous check-ins to win rewards, such as badges, mayorships and virtual coins [10]. Similarly, in Dianping [1], the most popular LBSN network in China, malicious users might publish unsolicited reviews to promote some selected restaurants, attracting legitimate users to visit [14]. On one hand, it is important to ensure that LBSN users are able to access reliable information via the platform, and enjoy their experiences in the urban life. On the other hand, there is an imperative need to prevent the negative impact of malicious users from providing unreliable input for urban computing applications. Therefore, it becomes a critical task to accurately detect malicious accounts in LBSNs, and to provide a reliable and trustworthy infrastructure for various urban computing applications. In this work, we investigate the malicious account detection problem in LBSNs with a case study of Dianping.

To detect malicious accounts in LBSNs, machine learning-based approaches have been introduced (e.g., [6, 14]). It is essential to select a number of distinct and informative features to empower the machine learning algorithms. In these works, most of the selected features provide an aggregate view of user activities, for example, the average travel speed of a user. However, such a view cannot reflect the evolution of the fine-grained activities, for example, the number of venues she has visited within each interval along the timeline. Without considering the dynamics of user activities, these existing approaches still suffer from a limited detection accuracy.

In order to better exploit the rich spatio-temporal information recorded by LBSN platforms, we propose to introduce time series analysis to better understand user behaviors from a dynamic perspective. We find that deep learning techniques are widely used to deal with the time series data, showing the potential to overcome the weaknesses of conventional statistical features. With deep learning, it is possible to effectively analyze the time series activities of users, and generate an output that represents the intrinsic characteristics of user

Q. Gong, Y. Chen, X. He, Z. Zhuang and X. Wang are with the School of Computer Science, Fudan University, China, and the Engineering Research Center of Cyber Security Auditing and Monitoring, Ministry of Education, China and the State Key Laboratory of Integrated Services Networks, Xidian University, China; T. Wang is with Beijing Bytedance Technology, China and Research Center of Precision Sening and Control, Institute of Automation, Chinese Academy of Sciences, China; H. Huang is with the School of Computer Science and Technology, Huazhong University of Science and Technology, China and the Institute of Computer Science, University of Göttingen, Germany; X. Fu is with the Institute of Computer Science, University of Göttingen, Germany.

[1] http://www.dianping.com/, accessed on October 15, 2017

mobility. We incorporate deep learning algorithms in our model to improve the detection performance.

In this work, we propose *DeepScan*, a solution to precisely identify malicious accounts in LBSNs. In DeepScan, we make use of time series analysis to obtain features representing a user's dynamic behavior. We introduce a deep learning algorithm based on long short-term memory (LSTM) neural network [4], a representative architecture widely applied to analyze time series activities. With LSTM, the fine-grained activity sequences of a user can be extracted as time series features, depicting the user activity history from an evolutionary view. We also extract conventional features, incorporating their contributions to analyze the difference of legitimate and malicious users. Putting these features together, we introduce a machine learning-based decision maker to distinguish between malicious and legitimate accounts. To implement the classification, we use XGBoost [2], a scalable machine learning library widely used for machine learning competitions. As DeepScan relies on publicly-accessible information entirely, it can be leveraged by third-party vendors conveniently. Based on the real user data collected from Dianping, we have conducted comprehensive evaluations to demonstrate the effectiveness of DeepScan. Our results show that DeepScan can achieve an F1-score of 0.964, indicating an excellent performance in malicious account detection.

II. Background and Data Collection

In this section, we illustrate the background of our study. We first give an overview of the Dianping service, a representative LBSN used for our case study. In addition, we show the presence of malicious accounts in Dianping. Finally, we describe how we collect the dataset for our study.

The Dianping Service: Dianping is a dominant LBSN in China, offering functionalities similar to Foursquare and Yelp. Dianping serves more than 200 million users [2]. In Dianping, a point of interest (POI) is defined as a venue, for example, a restaurant or a train station. There are more than 20 million venues in Dianping today. A Dianping user is allowed to conduct check-ins or post reviews for different venues, forming the two primary forms of UGCs. On one hand, a user can perform a check-in by using the mobile app of Dianping. By clicking the check-in button, she can publish her real-time location. On the other hand, a user can post a review for a selected venue at any time. Each Dianping user has a unique numeric UID. By visiting http://www.dianping.com/member/UID, we can access the corresponding Dianping user's personal page of her demographic information, and all her published check-ins and reviews. Each check-in entry contains a timestamp and the ID of the visited venue. The movement trajectory of the corresponding user can be constructed when putting her check-in entries together. Each review entry contains the text of the review, the ID of the corresponding venue, and the published time.

Malicious accounts in Dianping: Due to the openness of LBSNs, malicious accounts widely exist. Given the context of LBSNs, these accounts can publish misleading information to promote or disrupt the reputation of a certain venue [14]. In addition, malicious accounts might perform fake check-ins to get rewards from the LBSN platform [10], or, to increase the popularity of a venue in a manipulative way. All these activities will make the information of the platform incorrect. In order to improve user experience, detecting and blocking these accounts are extremely important. In particular, Dianping itself operates a fake review detection/filtering system to remove fake reviews [6]. Unfortunately, according to our measurement, still a substantial portion of accounts are malicious. In this work, we aim to build a scheme to detect malicious accounts in an accurate way, making the datasets from LBSN applications more reliable for urban computing.

Data Collection and Account Annotation: We conduct a data-driven study by using the demographic and behavior data of a large sample of Dianping users. By using the Selenium webdriver [3], we implement a Python-based crawler to crawl user profiles. During Mar. 11, 2017 - Apr. 16, 2017, we have crawled the profile pages of 107,616 Dianping users. The data entry of each user includes her demographic information (user ID, gender, registration date, birthday, number of followings, number of followers), her check-in records, her posted reviews, and her favorited venues. In each user's profile page, there are a time sequence of check-ins and a time sequence of reviews. In each of these two time sequences, the activities (check-ins or reviews) are listed in a reverse chronological order. These activities are considered as the user's time series activities. Following the practice in [12], we do not consider inactive accounts, as there is no enough data could be used for judgement. Among the crawled users, we focus on those who have conducted at least 5 check-ins and published at least 5 reviews.

To determine the ground truth of whether a user is legitimate or malicious, we build an online annotation platform using the Tornado web server. We have recruited 14 undergraduate students from Fudan University as volunteers to perform the annotation. A volunteer is expected to annotate about 2,000 accounts using our platform, and she will receive $30 after finishing the job. On our annotation platform, the volunteers can view each users' profile page, including the demographic information, check-ins and reviews. A volunteer can give her opinion about whether an account is legitimate or malicious. After a judgement submitted, the annotation platform will guide the volunteer to the profile page of the next Dianping user. To ensure the quality of annotation, each user's profile is examined by at least two independent volunteers. If the two volunteers have different opinions, we ask the third volunteer to break the tie. Finally, we have annotated 13,858 users, which form our Dianping dataset. Among all users within the dataset, 3,930 (28.36%) of them are concluded as malicious, while the rest 9,928 (71.64%) users are regarded as legitimate. We use this Dianping dataset to evaluate the performance of our malicious account detection approach.

[2] http://www.dianping.com/aboutus, accessed on October 15, 2017

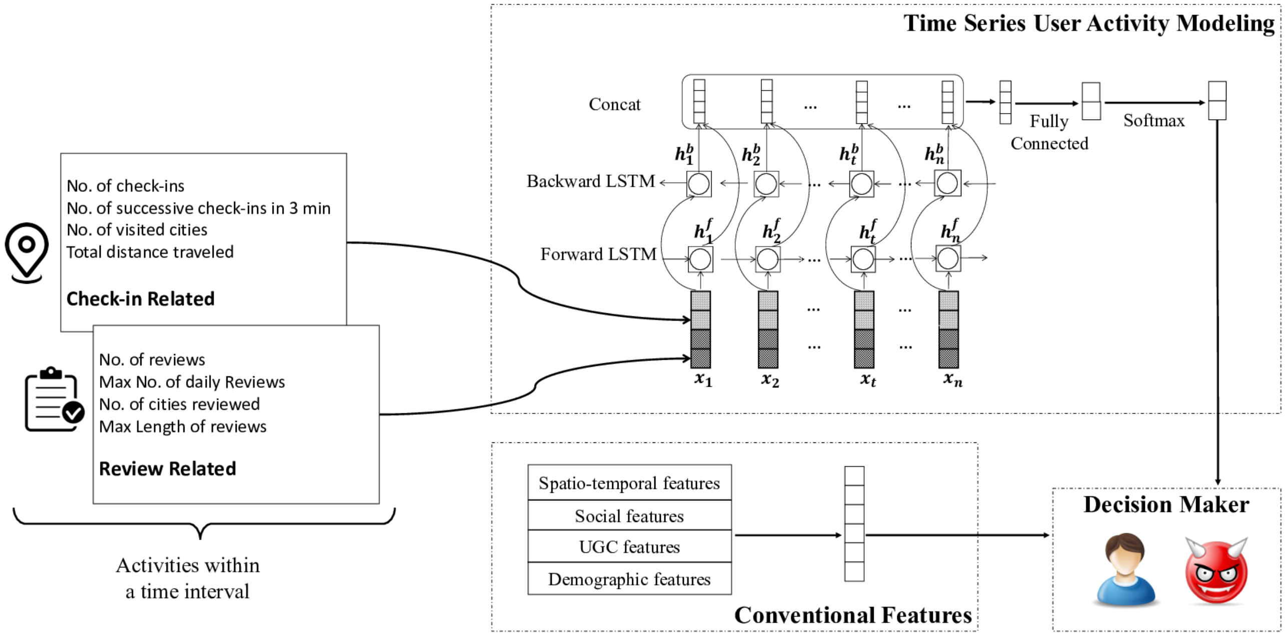

Fig. 1: System Overview of DeepScan

III. System Design

In this section, we present the architecture of DeepScan, a machine learning-based approach incorporating fine-grained user activities. DeepScan distinguishes itself from existing malicious account detection methods [5, 12, 14] in that it involves an LSTM network [4] to model user activities in time series.

A. System Overview

DeepScan includes three parts, i.e., time series user activity modeling, conventional feature extraction, and the decision maker module. The workflow of our system is shown in Fig. 1. For each user, we obtain all her activity histories and demographic information by referring to her personal page. We aim to extract the informative signals from this page to help determining whether she is malicious.

Statistically aggregating user behaviors might miss important information along the histories. Therefore, we introduce the LSTM network to model the dynamic user behaviors, which learns and builds the new time series features based on fine-grained activities of each user. The time series features can be used to effectively detect a malicious user. Putting the subsets of features shown in Table I together, the decision maker applies a supervised machine learning-based classifier to predict whether an account is malicious.

B. Time Series User Activity Modeling

Recurrent neural networks have been widely used in dealing with time series data. Such networks apply a chain-like architecture, i.e., an array of recurrently connected cells (neurons). Elements of a time sequence will be fed into this array sequentially. Each cell memorizes the information that has been calculated so far, in the form of a vector called hidden state. The hidden state at the previous time interval will be fed into the cell together with the current input element. Since the inputs are processed in temporal order, the current hidden state of the cell tends to be largely based on the latest inputs. As a result, standard recurrent neural networks are not able to capture time dependencies more than a few timesteps long, while the current hidden state might still be correlated to long-distance pieces of information in the inputs. To solve this challenge, we use LSTM, which is known for its ability to handle long-distance dependencies between elements in a time sequence [4]. Each cell in LSTM has three gates, i.e., the input, output and forget gates. With the help of these gates, the cell can control the fraction of information to get into or out of the cell, and generate a hidden state at each time interval accordingly. In particular, we apply bidirectional LSTM (Bi-LSTM), which comprises two parallel layers of LSTMs from both forward and backward directions. For each element in the time sequence, the Bi-LSTM network has the complete information before and after it. According to the study in [3], Bi-LSTMs outperform unidirectional LSTMs (Uni-LSTM) for classifying frames of acoustic data into phonemes.

We feed the LSTM network with a sequence of vectors, inputting one vector $\mathbf{x_t}$ at each step. Each vector $\mathbf{x_t}$ represents the user activities in the t-th time interval. Accordingly, the hidden state that memorizes the information until the time interval t can be represented as $\mathbf{h_t}$. The cell has two inputs at each step, i.e., vector $\mathbf{x_t}$ and the hidden state from the prior step $\mathbf{h_{t-1}}$. Bi-LSTM network comprises two parallel LSTM layers, i.e., the backward LSTM and the forward LSTM. Each layer deals with the input vector sequence independently. The

[3] http://www.seleniumhq.org/, accessed on October 15, 2017

TABLE I: Subsets of Features of DeepScan

Time Series Features	· Probability of Legitimate (Time Series Activities)
	· Probability of Malicious (Time Series Activities)
Spatio-temporal Features	· Number of check-ins
	· Average of check-in intervals
	· Variance of check-in intervals
	· Number of successive check-ins happened in less than 3 minutes
	· Average check-in speed
	· Number of visited cities
	· Fraction of check-ins in each of the 14 venue categories
	· Fraction of check-ins in each hour of a day
	· Fraction of check-ins in each day of a week
UGC Features	· Number of uploaded photos
	· Number of favorited venues
	· Number of reviews
	· Average of the review lengths
	· Variance of the review lengths
	· Maximum number of reviews per day
	· Maximum number of reviews to a single venue
	· Number of reviewed cities
	· Average of the ratings
	· Variance of the ratings
Social Features	· Number of followers
	· Number of followings
Demographic Features	· Gender
	· Age
	· Age of the account
	· Number of days since last login

hidden states $\mathbf{h_t^b}$ and $\mathbf{h_t^f}$ generated by the two LSTM layers at each time interval t will both infuse into the output layer, concatenating the output series of the Bi-LSTM. The last output, which is concatenated by the hidden states $\mathbf{h_n^b}$ and $\mathbf{h_n^f}$ of the two LSTM layers at the n_{th} time interval, is directly fed into a fully connected layer. This layer transforms the fed hidden states into a 2-dimensional vector. The vector will go through the softmax function (a normalized exponential function), and turn into two normalized probabilities. These two probabilities will serve as the time series features for the decision maker, indicating whether the user is malicious or legitimate, shown in Table I.

To form the vector $\mathbf{x_t}$, we carefully recognize the trajectory a user generates. For Dianping, the representative time-related activities are check-ins and reviews. Therefore, we explore the time series information by referring to these two types of activities. In our dataset, the time series activities range from Jul. 18, 2010 to Apr. 17, 2017. We need to translate the original records, i.e., check-ins and reviews, into a vector sequence. With a pre-defined time length, we can split the entire time duration into a set of successive time intervals with a fixed-length. For each time interval, we aggregate the check-ins and reviews records respectively, forming the 8-dimensional vector in this time interval. Elements in this vector aim to differentiate the malicious and legitimate users in comparatively small time intervals, including 4 check-in related and 4 review related dimensions. The 4 check-in related elements are the number of check-ins, the number of successive check-ins in less than 3 minutes, the number of visited cities, and the total distance traveled. The 4 review related elements are the number of reviews, the maximum review lengths, the maximum number of reviews to a single venue, and the number of reviewed cities. Therefore, for the t-th time interval, we can use an 8-dimensional vector $\mathbf{x_t}$ to represent the time series activities within the interval.

C. Conventional Features

As shown in Table I, we can still extract several conventional features from the personal page of a user, i.e., *spatio-temporal features*, *UGC features*, and *social features*. These features can reveal the difference between malicious and legitimate users from a statistical view over a comparatively long time period. In addition, the self-filled demographic information varies from one another, forming the user's *demographic features* directly. Details of these feature sets are illustrated as follows.

1) Statistical features: we broadly classify statistical features conventionally used for LBSN analysis into three categories: a) **Spatio-temporal features** record this user's spatio-temporal activities by learning her check-in trajectory. For each check-in, we are able to obtain the precise timestamp, and the ID of the corresponding venue. Since some features require demographic information of the venues, for example, the location, category and city information, we further crawl the venue profile pages for such information. b) **UGC features** represent her content generation behavior. We are interested in the user's lists of favorite venues, published reviews and how she rates the visited venues. c) **Social features** represent the user's status within the Dianping social network, including the numbers of followers and followings. Note that for check-in intervals, lengths of published reviews, and rating scores, we use both the average value and the variance.

2) Demographic features: **Demographic features** depict each user' basic demographic information, including her gender, age, age of the account, and number of days since last login.

D. The Decision Maker

Putting the time series features and all conventional features together, we introduce a decision maker module to determine whether an account is malicious. The decision maker is powered by a supervised machine learning-based classifier, taking the obtained features as the input. Most of the mainstream supervised machine learning algorithms can be used. Once a classifier is determined, we select a training and validation dataset to learn the parameters needed. After that, the classifier will be able to make judgement based on an account's features.

IV. Implementation and Performance Evaluation

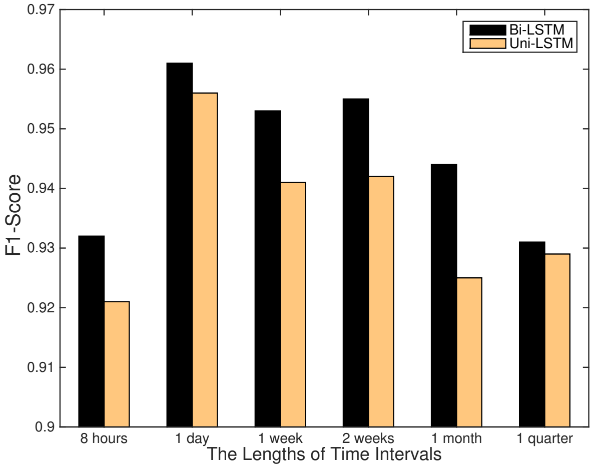

Fig. 2: Performance Comparison of Bi-LSTM and Uni-LSTM

In this section, we present the implementation details of DeepScan, and evaluate the prediction performance of our approach and other relevant proposals. To undertake the LSTM-based time series analysis, we use TensorFlow [4], an open source software library that is able to build and train neural networks. To implement the machine learning algorithms for the decision maker, we adopt XGBoost [5] and Weka [6]. In particular, XGBoost is a scalable end-to-end tree boosting system, which has been frequently used in recent machine learning challenges [2]. Weka can be used to implement several classic supervised machine learning algorithms, such as Random Forest, C4.5 Decision Tree (J48) and support vector machine (SVM).

From our Dianping dataset, we select half of all users to construct a training and validation dataset. The following three classic metrics are used to evaluate the performance of malicious account detection, i.e., precision, recall and F1-score. Precision represents the fraction of predicted malicious accounts who are really harmful. Recall means the fraction of malicious users who are detected accurately. F1-score is defined as the harmonic mean of precision and recall. For a given machine learning model, once we choose a set of parameters, we could calculate the corresponding precision, recall and F1-score using 10-fold cross validation. To find a set of "best" parameters, we apply grid search by sweeping a grid of parameters, and find the parameters that could achieve the highest F1-score. Afterwards, we then use the rest half of the users as a test dataset to evaluate each algorithm based on the parameters obtained by the training and validation phase.

Our evaluation aims to answer three questions. First, how effective the LSTM-based time series user activity modeling is. Second, how DeepScan behaves when incorporating both conventional features and the LSTM-based time series features. Finally, how much the LSTM-based time series features would be attributed to DeepScan, and whether the consideration of dynamic activities is beneficial to other existing detection systems.

A. Evaluating LSTM-based time series user activity modeling

As the critical component of DeepScan, we examine whether the LSTM-based time series features can accurately identify malicious accounts. We use the BasicLSTMCell module in TensorFlow to implement both the Bi-LSTM and Uni-LSTM networks, with default initialization settings. We train the models by sweeping a grid of two parameters, i.e., the number of hidden units that corresponds to the dimension of the hidden state vector, and the length of the time interval. Given a length of the time interval, we will be able to form the sequence of the 8-dimensional vectors, each containing the corresponding check-in and review related elements in the time interval. As we described in Fig. 1, after processed by the LSTM network and the softmax function, the input vector sequence becomes a pair of normalized probabilities, indicating if the user is malicious or legitimate. Here we conclude a user is malicious if the value of "probability of malicious" feature is larger than 0.5. The best performance is obtained for a given length of time interval, represented by the value of F1-score. Considering the long range of the time series activities that users cover in our dataset, we configure the lengths of time intervals as 8 hours, one day, one week, two weeks, one month, and one quarter, respectively. For each time interval, the number of the hidden units of both the Bi-LSTM and Uni-LSTM models are tuned for the highest F1-score. Performance comparison of Bi-LSTM and Uni-LSTM is shown in Fig 2. We have two key findings. First, Bi-LSTM achieves a better prediction performance than Uni-LSTM consistently. Second, we obtain the best results when setting the interval length as one day. The Bi-LSTM based models can accurately detect malicious accounts, with a highest F1-score of 0.961.

B. Evaluating DeepScan system as a whole

After determining the length of the time interval, we evaluate DeepScan as a whole. We compare among several supervised machine learning algorithms including XGBoost, Random Forest, C4.5 Decision Tree and SVM. In particular, for SVM, we consider both SVMr (with radial basis function kernel) and SVMp (with polynomial kernel). For each user, a feature vector is introduced, including the output of the LSTM-based time series features, and other conventional features listed in Table I. As shown in Table II, the comprehensive

[4] https://www.tensorflow.org/, accessed on October 15, 2017
[5] https://github.com/dmlc/xgboost/, accessed on October 15, 2017
[6] http://www.cs.waikato.ac.nz/ml/weka/, accessed on October 15, 2017

TABLE II: Comparison of Different Supervised Machine Learning Algorithms for the Decision Maker

Algorithm	Parameter	Precision	Recall	F1-Score
XGBoost	learning_rate=0.1,min_child_weight=2,max_depth=5, gamma=9, subsample=0.9,colsample_bytree=0.2, booster=gbtree,objective=binary:logistic	0.950	0.978	0.964
RF	MaxDepth=5, numFeatures=15	0.942	0.982	0.962
J48	Confidence factor =0.3, Instance/leaf M=5	0.941	0.982	0.961
SVMr	γ =0.5, Cost parameter = 20	0.940	0.970	0.955
SVMp	Degree = 3, Cost parameter = 10	0.940	0.955	0.948
BN	Default	0.757	0.829	0.791

design of DeepScan leads to a better performance than using an LSTM network only. By combining LSTM-based time series features and conventional features, DeepScan can further improve the detection performance. We achieve a new state-of-the-art by using XGBoost, with an F1-score of 0.964. We adopt XGBoost in the following subsection to compare the contributions by different features.

C. Evaluating the contribution of different feature sets of DeepScan

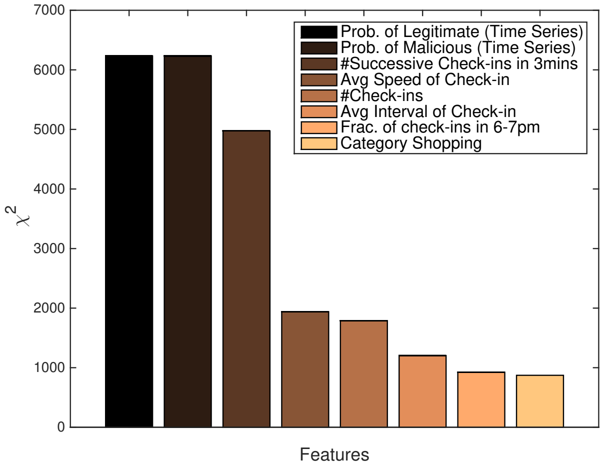

Fig. 3: Feature Importance: χ^2 Analysis

In this subsection, we investigate the role that the time series features play in DeepScan. By configuring the length of the time interval as one day, and adopting XGBoost as the machine learning algorithm of the decision maker, we can evaluate the prediction performance of DeepScan. To quantify the contributions of different features, we list the top 8 features ranked by χ^2 (Chi Square) statistics [11]. As shown in Fig. 3, we can see that the two most influential features are the two LSTM-based time series features. This reflects the significance of applying time series analysis.

We further investigate how each subset of features, i.e., time series features, demographic features, social features, UGC features, and spatio-temporal features, contributes to the overall detection performance of DeepScan. We remove one subset at a time, and see the prediction performance without this subset. Results are shown in Table III. Obviously, excluding the set of time series features deteriorates the detection performance the most. Looking at this problem from another perspective, we start from a random guess classifier. We add one subset of features at a time to evaluate the performance. We can see that adding the set of LSTM-based time series features could increase the F1-score the most, and adding the spatio-temporal features also achieves a very good performance.

We implement an SVM-based malicious account detection approach, which is a representative existing solution for detecting malicious accounts in LBSNs, designed by Zhang et al. [14]. We compare their approach and DeepScan using our Dianping dataset. Results reported in Table III suggest that DeepScan yields a better performance. Furthermore, we find that adding the time series features can also improve Zhang et al.'s approach, indicating the effectiveness of incorporating these features.

V. Related Work

Malicious account detection in OSNs: Various efforts have been made to the detection of malicious accounts in OSNs. Many of the earlier approaches are based on the assumption that malicious accounts have few social connections with legitimate users. For example, Cao et al. [1] propose a social graph-based approach to detect malicious accounts in Tuenti. However, Yang et al. [12] find that in Renren, there are numerous social connections between legitimate users and malicious accounts. They propose a malicious account detector based on interactions between users, for example, the activities of friend requests. Their detector has been deployed in Renren since August 2010. Unfortunately, such solutions require some private information only available to the OSN service provider. These approaches are designed for general-purpose OSNs, and they do not take the spatio-temporal information into consideration. Still, they serve as important references for detecting malicious accounts in LBSNs to select social features and UGC features.

Malicious account detection in LBSNs: Zhang et al. [14] propose an SVM-based approach to detect malicious accounts in Dianping. They extract location-based features to distinguish between legitimate users and malicious users. For comparison, we also implement and evaluate their approach with DeepScan using our Dianping dataset. According to our evaluation, DeepScan achieves a better prediction performance than their approach. Moreover, we also demonstrate that the addition of time series features can make their approach more accurate, indicating the usefulness of our LSTM-based time series features. By collaborating with Dianping, Li et al. [6] investigate the detection of opinion spams on Dianping by using an SVM-based model. However, their solution requires

TABLE III: Contribution analysis of each feature subset in DeepScan

Feature sets	Precision	Recall	F1-Score
DeepScan	0.950	0.978	0.964
- Time Series Features	0.863	0.864	0.864
- Spatio-temporal Features	0.941	0.981	0.961
- UGC Features	0.943	0.980	0.960
- Social Features	0.942	0.977	0.959
- Demographic Features	0.943	0.977	0.960
Random Guess	0.390	0.500	0.438
+ Time Series Features	0.941	0.982	0.961
+ Spatio-temporal Features	0.872	0.876	0.874
+ UGC Features	0.741	0.932	0.825
+ Social Features	0.723	0.930	0.832
+ Demographic Features	0.724	0.961	0.826
Zhang et al.'s approach [14]	0.912	0.924	0.918
Zhang et al.'s approach + Time Series Features	0.941	0.981	0.961

Dianping to provide some private information that are not accessible for the public, such as the IP address of each user. As a result, their solution can only be deployed in the single specific LBSN platform. Differently, we only use the publicly-accessible information of LBSN users. Our approach can help different third-party vendors who are interested in using the rich user activity data recorded by LBSNs.

LSTM in online user behavior analysis: LSTM has been widely used to analyze the objects in the form of time sequence. Ma et al. [7] detect rumors on microblogging platforms by applying LSTM to the modeled time series event information. Suhara et al. [9] forecast people's depression moods based on individual's historical moods, behavioral type, and medical records, collected by their self-developed smartphone application. They use LSTM to analyze the time series of individual histories and obtain an indicative output to forecast depressive moods. However, to our best knowledge, LSTM or more generally, the time series features have not yet been incorporated in LBSN user behavior analysis.

VI. Discussion and Future Work

In this paper, we propose DeepScan, a machine learning-based malicious account detection system for LBSNs. DeepScan carefully considers the users' activities from an evolutionary view by dividing each user's activity data into a number of continuous time intervals. Using deep learning technologies, DeepScan makes use of time series features, which are more discriminative and informative than conventional features, i.e., statistical or demographic features. Using the real data collected from Dianping, we find that DeepScan can achieve an excellent performance with an F1-score of 0.964.

Our study still has some limitations. Given the massive user activities recorded by LBSN platforms, we believe that the idea of time series analysis can be further explored to accurately identify malicious accounts in different LBSNs. Nevertheless, we only use the dataset from Dianping to evaluate our design. There are two key reasons. On one hand, Dianping is a popular LBSN service, and all check-ins and reviews of each user in Dianping are publicly-accessible. This gets us a large-scale dataset of an LBSN. On the other hand, in other LBSNs such as Foursquare, the spatio-temporal information is not fully open to the public. For example, a user's check-ins in Foursquare are only available to her friends. Therefore, crawling a dataset containing a large number of users' check-in trajectories is challenging. Expanding the idea of DeepScan to other LBSNs would be an important future work for us. One possible way is to collaborate with some LBSN operators.

We wish that our solution can motivate various application providers to provide large-scale and reliable LBSN datasets to investigate behavior patterns of massive urban users. For example, modeling the real-time user distribution, predicting the user mobility and discovering popular venues.

Acknowledgement

This work is sponsored by National Natural Science Foundation of China (No. 61602122, No. 71731004, No. U1636220, No. 61472423), Natural Science Foundation of Shanghai (No. 16ZR1402200), Shanghai Pujiang Program (No. 16PJ1400700), EU FP7 IRSES MobileCloud project (No. 612212) and Lindemann Foundation (No. 12-2016). Yang Chen is the corresponding author.

References

[1] Q. Cao, M. Sirivianos, X. Yang, and T. Pregueiro. Aiding the Detection of Fake Accounts in Large Scale Social Online Services. In Proc. of the 9th USENIX Symposium on Networked Systems Design and Implementation (NSDI), pages 197–210, 2012.

[2] T. Chen and C. Guestrin. XGBoost: A Scalable Tree Boosting System. In Proc. of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), pages 785–794, 2016.

[3] A. Graves and J. Schmidhuber. Framewise phoneme classification with bidirectional LSTM networks. In Proc. of IEEE International Joint Conference on Neural Networks, volume 4, pages 2047–2052, 2005.

[4] S. Hochreiter and J. Schmidhuber. Long short-term memory. Neural Computation, 9(8):1735–1780, Nov. 1997.

[5] X. Hu, J. Tang, Y. Zhang, and H. Liu. Social Spammer Detection in Microblogging. In Proc. of the 23rd International Joint Conference on Artificial Intelligence (IJCAI), pages 2633–2639, 2013.

[6] H. Li, Z. Chen, A. Mukherjee, B. Liu, and J. Shao. Analyzing and Detecting Opinion Spam on a Large-scale Dataset via Temporal and Spatial Patterns. In Proc. of the 9th International Conference on Web and Social Media (ICWSM), pages 634–637, 2015.

[7] J. Ma, W. Gao, P. Mitra, S. Kwon, B. J. Jansen, K. Wong, and M. Cha. Detecting Rumors from Microblogs with Recurrent Neural Networks. In Proc. of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI), pages 3818–3824, 2016.

[8] T. H. Silva, P. O. S. Vaz de Melo, J. M. Almeida, J. Salles, and A. A. F. Loureiro. Revealing the City That We Cannot See. ACM Transactions on Internet Technology (TOIT), 14(4):26:1–26:23, 2014.

[9] Y. Suhara, Y. Xu, and A. S. Pentland. DeepMood: Forecasting Depressed Mood Based on Self-Reported Histories via Recurrent Neural Networks. In Proc. of the 26th International Conference on World Wide Web (WWW), pages 715–724, 2017.

[10] G. Wang, S. Y. Schoenebeck, H. Zheng, and B. Y. Zhao. “Will Check-in for Badges”: Understanding Bias and Misbehavior on Location-based Social Networks. In Proc. of the 10th International Conference on Web and Social Media (ICWSM), pages 417–426, 2016.

[11] Y. Yang and J. O. Pedersen. A Comparative Study on Feature Selection in Text Categorization. In Proc. of the 14th International Conference on Machine Learning (ICML), pages 412–420, 1997.

[12] Z. Yang, C. Wilson, X. Wang, T. Gao, B. Y. Zhao, and Y. Dai. Uncovering Social Network Sybils in the Wild. ACM Transactions on Knowledge Discovery from Data (TKDD), 8(1):2:1–2:29, 2014.

[13] N. J. Yuan, F. Zhang, D. Lian, K. Zheng, S. Yu, and X. Xie. We know how you live: exploring the spectrum of urban lifestyles. In Proc. of ACM Conference on Online Social Networks (COSN), pages 3–14, 2013.

[14] X. Zhang, H. Zheng, X. Li, S. Du, and H. Zhu. You are Where You Have Been: Sybil Detection via Geo-location Analysis in OSNs. In Proc. of IEEE Global Communications Conference (GLOBECOM), pages 698–703, 2014.

[15] Y. Zhao, L. Nie, X. Wang, and T. Chua. Personalized Recommendations of Locally Interesting Venues to Tourists via Cross-Region Community Matching. ACM Transactions on Intelligent Systems and Technology (TIST), 5(3):50:1–50:26, 2014.

Qingyuan Gong (qgong12@fudan.edu.cn) received her B.S. degree in Computer Science from Shandong Normal University in 2012. She is now a PhD candidate in Computer Science at Fudan University. Her research interests include network security, user behavior analysis and modeling, and distributed storage systems. She published referred papers in IEEE ICPP and the Journal of Supercomputing. She has been a visiting student at the University of Göttingen in 2015.

Yang Chen (chenyang@fudan.edu.cn) is a Pre-tenure Associate Professor within the School of Computer Science at Fudan University. Before that, he was a postdoctoral associate at the Department of Computer Science, Duke University, USA, and a research associate at the Institute of Computer Science, University of Göttingen, Germany. He received his B.S. and Ph.D. degrees from Tsinghua University in 2004 and 2009, respectively. His research interests include online social networks, Internet architecture and mobile computing.

Xinlei He (xlhe17@fudan.edu.cn) received his B.S. degree in Computer Science from Fudan University in 2017. He is now a master student in Computer Science at Fudan University. His research interests include social network analytics and data mining.

Zhou Zhuang (zzhuang14@fudan.edu.cn) is an undergraduate student in Computer Science at Fudan University. He visited the Institute of Software, Chinese Academy of Sciences (in 2016) and the University of Göttingen (in 2017). His research interests include massive data analytics and machine learning.

Tianyi Wang (wangtianyi.data@bytedance.com) is a senior R&D at ByteDance Inc. Previously he was a research scientist at Baidu Research. He received his B.S. and Ph.D. degrees from Department of Electronic Engineering, Tsinghua University in 2011 and 2016, respectively. His research interests include data mining, machine learning, NLP and security, mostly from a data-driven perspective. He published more than 10 referred papers in international journals and conferences, including ACM TWEB, IEEE Communications Magazine, USENIX Security, ACM IMC, MobiSys and CSCW.

Hong Huang (honghuang@hust.edu.cn) is currently an assistant professor in School of Computer Science and Technology, Huazhong University of Science and Technology and an adjunct researcher at the Computer Networks Group, University of Göttingen, Germany. Before that, she received her Ph.D. degree (summa cum laude) from the University of Göttingen in 2016 and M.E. degree in Electronic Engineering from Tsinghua University, China in 2012. Her research interests include social network analysis, social influence and data mining.

Xin Wang (xinw@fudan.edu.cn) received his BS Degree in Information Theory and MS Degree in Communication and Electronic Systems from Xidian University, China, in 1994 and 1997, respectively. He received his Ph.D. Degree in Computer Science from Shizuoka University, Japan, in 2002. He is currently a professor at Fudan University, Shanghai, China. His research interests include quality of network service, next-generation network architecture, mobile Internet and network coding.

Xiaoming Fu (fu@cs.uni-goettingen.de) received his Ph.D. from Tsinghua University and is currently Professor of Computer Science at University of Göttingen. He is interested in networked systems and services, cloud computing, mobile computing, big data and social networks. He has served as member of several journals' editorial boards (IEEE TNSM, ComMag, Elsevier ComNet, ComCom...) and conference committees (SIGCOMM, MobiCom, INFOCOM, ICNP...), and as elected officers of IEEE ComSoc's Technical Committees on Computer Communications (TCCC) and Internet (ITC).