This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2852008, IEEE Access

Digital Object Identifier

Positive and Unlabeled Learning for User Behavior Analysis based on Mobile Internet Traffic Data

KE YU[1], YUE LIU[1], LINBO QING[2], BINBIN WANG[1], and YONGQIANG CHENG[3],
[1]School of Information and Communication Engineering, Beijing University of Posts and Telecommunications, Beijing 100876, China (e-mail: yuke@bupt.edu.cn, liuyuebupt@bupt.edu.cn, wangbb1994@gmail.com)
[2]College of Electronics and Information Engineering, Sichuan University, Chengdu, China (e-mail: qing_lb@scu.edu.cn)
[3]School of Engineering and Computer Science, University of Hull, UK (email: y.cheng@hull.ac.uk)

Corresponding author: LINBO QING (e-mail: qing_lb@scu.edu.cn).

This work is supported by the National Natural Science Foundation of China under Grant No. 61601046 and No.61171098, the 111 Project of China under Grant No.B08004, and EU FP7 IRSES Mobile Cloud Project under Grant No. 612212.

ABSTRACT With the rapid development of wireless communication and mobile Internet, mobile phone becomes ubiquitous and functions as a versatile and smart system, on which people frequently interact with various mobile applications (Apps). Understanding human behaviors using mobile phone is significant for mobile system developers, for human-centered system optimization and better service provisioning. In this paper, we focus on mobile user behavior analysis and prediction based on mobile Internet traffic data. Real traffic flow data is collected from the public network of Internet Service Providers (ISPs), by high-performance network traffic monitors. We construct User-App bipartite network to represent the traffic interaction pattern between users and App servers. After mining the explicit and implicit features from User-App bipartite network, we propose two positive and unlabeled learning (PU learning) methods, including Spy-based PU learning and K-means-based PU learning, for App usage prediction and mobile video traffic identification. We firstly use the traffic flow data of QQ, a very famous messaging and social media application possessing high market share in China, as the experimental dataset for App usage prediction task. Then we use the traffic flow data from six popular Apps, including video intensive Apps (Youku, Baofeng, LeTV, Tudou) and other Apps (Meituan, Apple), as the experimental dataset for mobile video traffic identification task. Experimental results show that our proposed PU learning methods perform well in both tasks.

INDEX TERMS Mobile user behavior, PU learning, App usage prediction, Mobile video traffic identification, Bipartite network.

I. INTRODUCTION

THE rapid development of information and communication technologies continues to shape our societies, cultures and economies. Nowadays mobile phone is revolutionizing the way we learn, live, work, and even play. Using various mobile applications (Apps) on mobile phones, people get easy access to services and information whenever and wherever they want it. Moreover, by incorporating built-in camera, GPS receiver and other kinds of sensors, mobile phone has been regarded as the substitute for digital camera, navigator, sport tracker, and so on. In the era of Internet/Web of Things, with mobile computing and environment sensing abilities, mobile phone can automatically and intelligently serve people in a collaborative manner [2].

Understanding human behaviors is one of the most significant task for human-centered service systems. More specifically, better understanding of human behavior is not only significant for mobile system developers to optimize their service provisioning, but also valuable for ISPs (Internet Service Providers), for better management of network. For example, since mobile phones are generally kept on at all times and carried everywhere, they function as a platform for real-time self-monitoring and environment sensing in e-Health service. The collected eHealth data from the mobile phone, including patient's physical and mental signs, symptoms, behaviors, as well as self-report data, can aid medical decision making and provide more personalized treatment. Another example is urban computing in Smart city. Mobile

phone tracks people's location every time they text, call or browse Web. By trajectory tracking and behavior predicting, better inter-networking and transportation services will be provided for the public, and suitable location-based services will be recommended for individuals.

The researches in existence on mobile user behavior analysis are mainly utilizing App usage logs from sensor reports from built-in sensors or from App servers [3] [4]. These datasets are in general relatively small and specific to particular application or user. Actually mobile Internet traffic data is collected from ISP's traffic monitoring systems or network routers, which can be exploited to analyze user behavior and make further prediction. A single traffic flow is defined as one or more packets sent from a particular source host and port, to a particular destination host and port, using a specific protocol, over some time interval [5]. The traffic flow data is actually big data that exhibits an overall perspective of user behavior between different mobile applications.

In this study, our focus is mobile user behavior analysis based on mobile Internet traffic data. The information contained in each traffic flow record includes occurring time, duration, source IP addresses, destination IP addresses, total number of packets and bytes in the flow, type of application, and mobile phone brand and model, and mobile phone number. There are two tasks for our study, one is App usage prediction, i.e. prediction of whether a visiting activity will happen from a mobile user to a specific App server based on his or her historical visiting records; and the other is mobile video traffic identification, i.e. to identify a specific traffic flow as video or non-video traffic. We intend to design a framework combining data preprocessing and machine learning to accomplish the above two tasks. The first challenge is how to effectively and efficiently extract feature vectors representing user and App server interaction pattern from the information in traffic flow data. We propose to construct a User-App bipartite network to extract attributes associated with the user and the App Server.

The second challenge comes to the problem of learning simply from data with merely positive and unlabeled records. The real Internet traffic flow data only records the interaction information between users and servers. If a traffic interaction is established between a user and a server, such a flow record is then regarded as a positive example; else it is regarded as unlabeled. In this way, the learning model can only exploit a rather small set of positive examples, along with a relatively large set of unlabeled examples. We propose two Positive and Unlabeled Learning methods (i.e. PU learning) to accomplish the prediction and classification task.

The main contributions of this paper are listed as follows:

1) Construction of a User-App bipartite network based on mobile traffic flow data and the analysis on the attributes associated with the user and App server nodes.
2) Proposal of two PU Learning methods based on the features extracted from the User-App bipartite network for user behavior classification and prediction.
3) Verification and validation of our proposed methods for App usage prediction task and traffic identification task in experiments, using traffic flow data from Tencent QQ and popular mobile video Apps.

The rest of the paper is organized as follows. In Section II, we do literature review on App usage prediction, traffic identification and PU learning. In Section III, our framework of the mobile user behavior analysis, User-App bipartite network construction, feature extraction and selection, and PU learning methods are discussed in details. Section IV describes the experimental results. Section V concludes the paper.

II. RELATED WORK

With the increasing popularization of smartphones, mobile applications (Apps) is being designed and developed at an unprecedented speed to meet users' demands. Understanding mobile user's behavior is helpful not only for App design and information recommendation, but also for better network management and service provisioning. A vast number of researches have been reported in literature on the analysis of user behavior on mobile Internet. In [6], the data usage, application usage, and mobility pattern of mobile user were presented. Li et al. [7] analyzed the user activity pattern and viewing behavior when users access the live streaming system. Ravari et al. [8] investigated mobile users' location search characteristics using GPS-navigation service. In [9], the interactions between a mobile crowdsensing server (MCS) and large quantities of smartphone users were analyzed and modeled as a Stackelberg game. Mo et al. [10] proposed a cloud-based mobile multimedia recommendation system, in which the recommendation rules are generated from users' contexts, relationships, and profiles collected from video-sharing websites. Zhang et al. [11] proposed a user behavior modeling framework based on multi-state model and hidden Markov model and extracted typical sequential behavior patterns by clustering.

There are many researches focusing on predicting the interaction between mobile App users and servers, i.e. App usage prediction problems. In [3], by analyzing real log data of App usage, the authors discovered two categories of features: the Explicit Feature (EF) collected from built-in sensors' readings, and the Implicit Feature (IF) extracted from App usage relations. A personalized algorithm for feature selection was proposed to the further prediction of App usage. In [4], the authors designed a widget AppNow, which is capable of predicting users' App usage based on historical data of App usage. Huang et al. [12] exploited a multiple of contextual information, such as time, location, last used App, and the user profile, to predict users' App usage. Shin et al. [13] used a collection of 37 sensor readings as features to conduct App usage prediction. From the above review, we can see that the existing researches on mobile App usage prediction are majorly exploiting sensor readings from smart devices and App usage logs from App servers, which usually are small and specific to particular application or user.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2852008, IEEE Access

There are also several researches focusing on Internet traffic identification problem, which is useful for network fault detection, network planning, network service quality improving, and anomaly traffic detection etc. Traditional traffic flow detection methods can be grouped into three major categories, i.e. port-based, packet-based, and flow-based methods. Port-based methods directly use the source and destination port number from the header of datagrams in transport layer [14]. While achieving simple and fast identification, these methods are ineffective for some less common, dynamically changing and deliberately hidden port numbers in practical applications. Packet-based methods use Deep Packet Inspection (DPI) algorithm, which first obtain signature strings by parsing packet content and extracting features, then match the signature string to achieve traffic identification. The DPI method is widely applied in practice as it can detect protocol type with high accuracy. However, it is incapable of parsing encrypted packets, faced with invasion of privacy. Also, such methods require prior knowledge of each application, which leads to identification failure for new applications. Flow-based methods classify traffic flow data by analyzing different behavior patterns of different applications on flow level. These methods require neither port numbers nor packet content, but statistical characteristics of packets on network layer, such as time of duration, interval of packet arrival and packet length. By detailed analysis of these differences, such methods achieve good performance. The above three categories of methods focus mainly on the classification of traffic flow data directly, while there are few researches on separately detecting and identifying mobile video traffic. Jiang et al. [15] designed an algorithm based on k-nearest-neighbor (k-NN) to identify traffic flow of multi-media applications. Hao et al. [16] presented a simple yet effective method achieving real time online detection of multi-media traffic by statistical inference of the temporal characteristics of packets.

In our study, we focus on analyzing mobile user behavior on the basis of real Internet traffic data [1] [17]. This paper proposes a framework combining traffic data preprocessing and machine learning to accomplish user behavior prediction and classification.

Positive and unlabeled learning has been studied over the past few years to tackle the inadequacy of negative examples in training data. Primitively, it was put forward to solve the problem of text classification [18] [19]. Subsequently, it was reintroduced to other areas of classification learning. In [20], PU learning was applied to graph classification. In [21], a novel PU learning technique, namely LELC (PU Learning by Extracting Likely positive and negative micro-Clusters) was presented to classify data stream. In [22], for deceptive reviews detection problem, a semi-supervised model, called MPIPUL (Mixing Population and Individual Property PU Learning) based on LDA (Latent Dirichlet Allocation) and SVMs (Support Vector Machines), was proposed. Yang et al. [23] designed a novel positive-unlabeled learning algorithm to identify disease genes (PUDI). Li [24] applied PU learning to the problem of detecting fake reviews in Chinese on Dianping, the counterpart of Yelp on Chinese webs. Lan et al. [25] presented a method named PUDT to predict drug-target interactions, in which unlabeled samples are grouped into reliable negative samples (RN) and likely negative samples (LN) by aggregating three strategies (Random walk with restarts, KNN and heat kernel diffusion) based on majority voting to decide the final label of unlabeled samples, and a classifier is built by weighted SVM.

The existing methods of PU learning are generally categorized into three types [26]. The first category adopts a two-step strategy, firstly extracting some reliable negative examples from unlabeled data, and then executing the classification task on the reliable negative examples and positive examples by using traditional classification algorithms such as Naive Bayes, Support Vector Machines (SVMs) and Expectation Maximization Algorithm (EM) [18] [27] [28] [29] [30]. The second category of methods estimates statistical queries over positive and unlabeled examples [31] [32] [33]. The third category of methods reduces this problem to learning with high one-sided noise by treating the unlabeled set as noisy negative examples [19] [34] [35]. In this study, we propose two PU learning methods using two-step strategy, i.e. Spy-based PU learning and K-means-based PU learning, to finish the task of mobile App usage prediction and mobile video traffic identification.

III. MOBILE USER BEHAVIOR ANALYSIS AND LEARNING

A. MOBILE USER BEHAVIOR ANALYSIS FRAMEWORK

In our study, mobile user behavior analysis includes three steps. Firstly, traffic flow data is gathered based on traffic rules and preprocessed on Map-Reduce system and stored in distributed database. Secondly, user behavior features are analyzed and selected, and machine learning models are trained. Thirdly, the data models are applied in App usage prediction, mobile video traffic identification, etc. The mobile user behavior analysis framework is shown in Fig. 1

B. INTERNET TRAFFIC FLOW DATA

In this study, we use the Internet traffic flow data collected by proprietary traffic monitors with superior performance in the operational mobile network of an ISP, providing services for millions of people from a southwestern province in China. As shown in Fig. 2, an ISP's mobile network is composed of the access network and the core network. A mobile device communicates with Evolved Node B (eNB) in the access network, which forwards its data traffic to the core network. In the core network, Serving Gateway (S-GW) and PDN Gateway (P-GW) provide connectivity to the Internet. The requesting messages from some mobile device enter the Internet and reach the corresponding server through this path. The server's responding messages traverse in the backward path to reach the mobile device. In practical situation, when a user initiates an App on his/her phone, the requesting messages are sent from the mobile device to the corresponding server, through

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2852008, IEEE Access

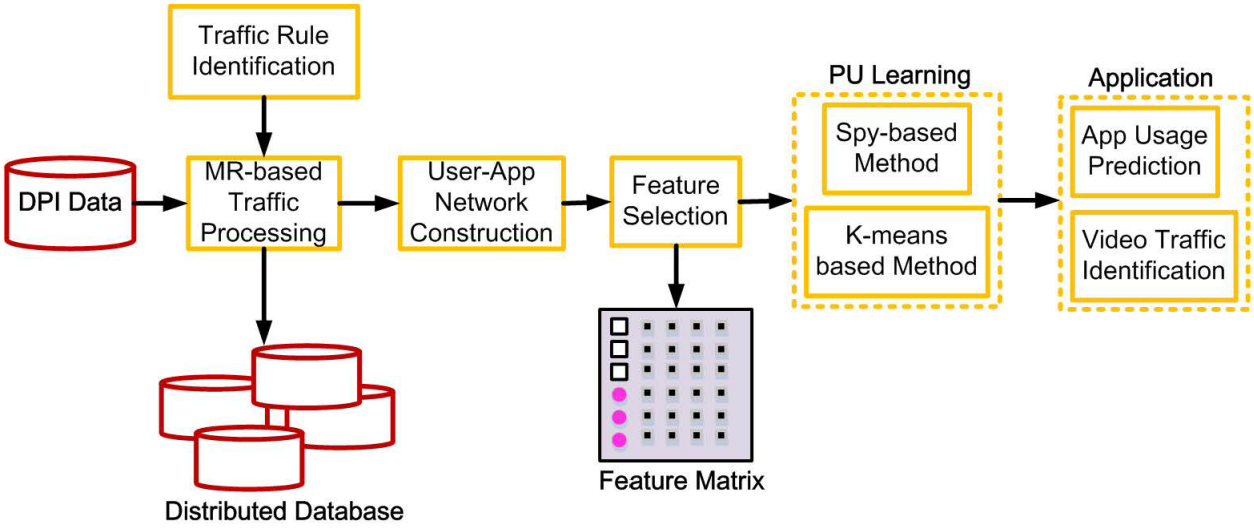


FIGURE 1: Mobile User Behavior Analysis Framework.

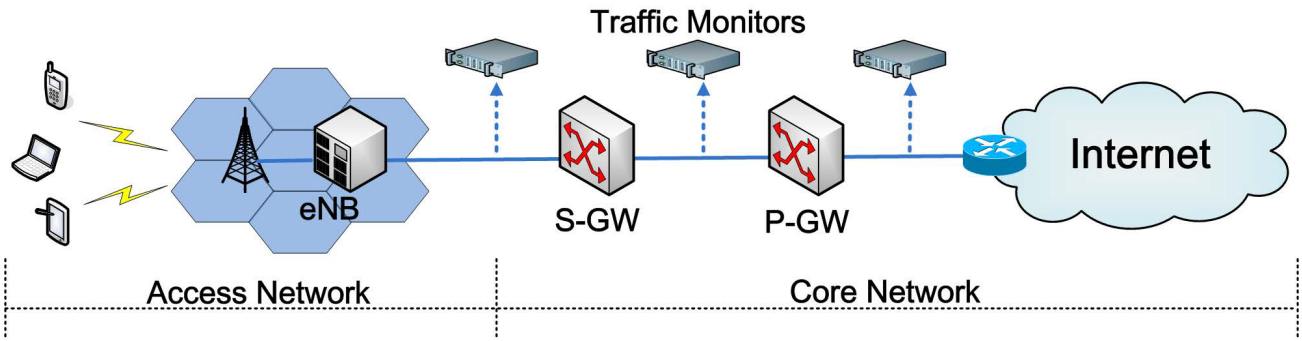


FIGURE 2: Traffic Monitors in Mobile Internet.

the aforementioned key nodes (eNB, S-GW, P-GW). The server's responding messages traverse in the backward path. The protocol interaction process is complicated and depends on scenarios, so there may be a flow record or multiple flow records.

The traffic monitors are deployed in the core network, to capture packets at different types of network interfaces. The traffic monitor can not only support flow composition with accuracy and packet paring at light speed, but also offer real-time traffic classification by combining port-based application deduction, DPI, and deep flow inspection (DFI) technologies, which have been briefly introduced in Section II. By using the above classical methods, the traffic monitors can identify a part of application types, about 80% of the total applications, since it is difficult to deal with new Apps and Apps with encrypted contents.

The dataset we used in this paper was a 96-hour-period traffic flow data collected on December, 2013. During each 24-hour period (0:00 am to 11:59 pm), over 1.5 billion records of traffic flow was collected. The entries of each record in detail include occurring time, flow duration, source IP addresses, destination IP addresses, mobile phone number, application type, total number of packets and bytes in the flow, and mobile phone model and brand. For example, the application type attribute consists of over 15 identified categories, which can be further divided into more than 100 sub-categories, including Web, Music, Game, Search, E-commerce, Advertise, Video, etc. The mobile phone attribute consists of over 20 major smartphone brands and over 100 identified models such as Huawei, Apple, and Samsung, which have large market share in China.

C. USER-APP BIPARTITE NETWORK

Based on the traffic flow data collected, the User-App Bipartite Network is constructed to represent the traffic interaction pattern. In the User-App Bipartite Network, there are two types of nodes, i.e. the user node (denoted by mobile phone number) and the server node (denoted by specific IP address). The user node has the characteristics such as phone number section (the first three digits of the phone number), mobile phone brand and model. The server node has the characteristics such as application category and sub-category. The directed edge is formed between the user node and the server node. The edge between the user node and the server node is

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2852008, IEEE Access

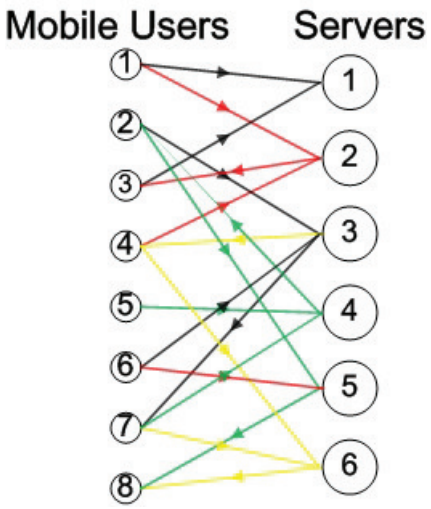


FIGURE 3: User-App Bipartite Network.

formed by the actual data transfer, which is revealed by each traffic flow record. The characteristics of an edge include start time, end time, packets, bytes, direction (from user to server is upstream, from server to user is downstream), application category label (determined by the server node's category at the time). Note that multiple flow records between the same pair of nodes are considered multiple edges, since the characteristics of edges are different.

The User-App Bipartite Network is illustrated in Fig. 3, which is essentially a directed bipartite graph. The edges in the User-App Bipartite Network are identified by different colors, which indicate different application category labels. Since a mobile user can use multiple applications simultaneously, one user node may have more than one colored edges at the same time. The server node may have multiple colored edges at the same time, but this situation is relatively rare, because more than 80 percent servers support only one service according to our actual statistical analysis.

D. FEATURE SELECTION

As is discussed in Section III-B, the traffic flow data is preprocessed and stored in Hadoop Distributed File System (HDFS), containing abundant information. We need to convert the flow data into feature vectors for machine learning algorithms. Based on the actual application scenarios, we select different dimensions of feature vectors with respect to the two different tasks.

For the App usage prediction task, we construct a 194-dimensional feature vector. The features can be divided into two categories, i.e. mobile user node related features and server node related features. The mobile user node related features include National Destination Code (the first three numbers of the user's phone number in China, which convey information about which provider is providing service, i.e. 138 and 139 indicate services provided by China Mobile, 189 and 133 by China Telecom, and 130 and 131 by China Unicom), mobile phone brand, connection duration, user active time by hour, etc. The server node related features include average number of connections, server active time by hour, service category, etc. Note that the User-App Network is a directed bipartite graph. For a node in directed graph, in-degree is defined as the number of incoming connections, while out-degree is defined as the number of outgoing connections. The degree of a node is the sum of in-degree and out-degree. For the User-App Network, the edge/connection means the traffic interaction between user and App server, and mostly the incoming and outgoing connections exist simultaneously. So we use degree as one feature of user and server, instead of in-degree and out-degree. For a weighted graph, the node strength is the sum of weights of all edges it connects to. In the User-App Network, the total number of packets related to a traffic flow is used as edge weight. For a user node, the in-strength and the out-strength are the average downlink and uplink packets respectively. For a server node, the in-strength and the out-strength are the average uplink and downlink packets respectively. The first two categories of features in Table 1 show their composition in detail, each having 8 sub-categories of information.

For the video traffic identification task, we construct a 105-dimensional feature vector. The features are made up of connection duration, the number of connections per hour, uplink and downlink packets, uplink and downlink bytes, the ratio of bytes to packets etc. The statistical properties of the above-mentioned features include maximum value, minimum value, mean value, median, sum, variance, range, coefficient of variation (the ratio of standard deviation to mean value), etc. The third and fourth categories of features in Table 1 show the composition of the selected feature vectors.

E. PU LEARNING METHODS

In order to classify or predict user behavior, a suitable learning model is required. Considering the traffic flow data, we only have the record of interaction information between users and App servers, which is regarded as a positive example. In other words, if the user has a traffic interaction record with the server, it is a positive example, otherwise is an unlabeled example. In our real traffic flow, the set of positive examples is quite small in comparison with large quantities of unlabeled examples. Hence, a PU learning method is proposed based on two-step strategy for mobile user behavior analysis.

1) To identify a set of reliable negative examples from the set of unlabeled data based on spy dataset.
2) To build a classifier based on reliable negative examples and positive examples.

Firstly, by sampling ratio $1-\beta$, we randomly select some positive examples from all positive data (P) to form a set SP. The examples in SP are named spy examples. Considering that in the real flow data, the amount of unlabeled examples is much larger than positive examples, we also randomly select

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2852008, IEEE Access

TABLE 1: The Selected Features

	Features
user related	National Destination Code
	Mobile phone brand
	active time by hour
	number of connections (degree)
	number of connection types
	connection duration
	average uplink packets (out-strength)
	average downlink packets (in-strength)
server related	number of connection types
	service category
	service sub-category
	active time by hour
	number of connections (degree)
	connection duration
	average uplink packets (in-strength)
	average downlink packets (out-strength)
user-server interaction features	connection duration
	the number of connections per hour
	uplink packets
	downlink packets
	uplink bytes
	downlink bytes
	the ratio of bytes to packets
statistical properties	maximum value
	minimum value
	mean value
	median
	sum
	variance
	range
	coefficient of variation

a subset US from U by sampling ratio α. Then, we build a binary classifier using (P - SP) and (US + SP). With this classifier, each example in (US + SP) is categorized into positive or negative set. Next, a probability threshold t is determined using SP. Each example in US with prediction probability greater than t is labeled as reliable negative examples (RN). Fig. 4 dipicts the basic process of extracting reliable negative examples from unlabeled data.

Secondly, another classifier is built using P and RN to complete the specific task, for example, to predict whether a certain connection from a user to an App server will happen in the future. Detailed algorithm for the identification problem of those RN examples with the highest probabilities and the prediction of future connections is elaborated in Algorithm 1. It should be noted that there are many options for the classifier, such as Random Forest, Decision Tree, Naive Bayes, Logistic Regression, SVM, etc. The outcome of Algorithm 1 using different classifiers is usually different. In later parts of this paper, three of them (Random Forest, Decision Tree and Logistic Regression) are chosen to do the experiment separately.

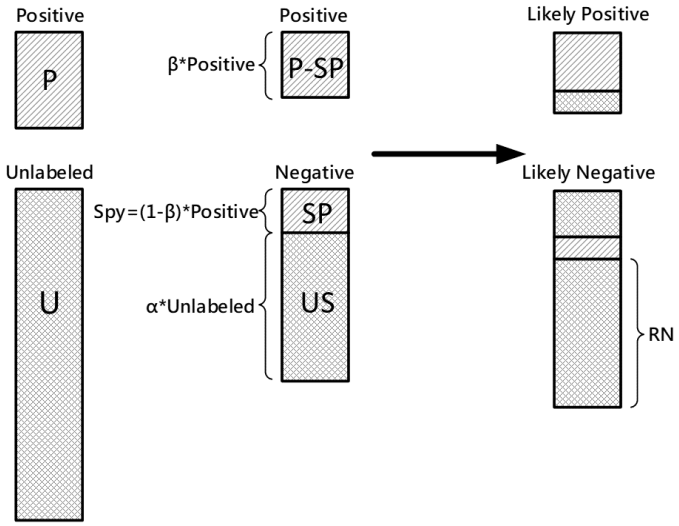


FIGURE 4: Extracting Reliable Negative Examples from Unlabeled Data [1]

In our proposed PU-based learning method, how to determine the threshold t is a key point. Let the set of spy examples SP be $\{s_1, s_2, s_3, \cdots, s_k\}$, and the probabilistic label 0 assigned to each s_i be $P(c_0|s_i)$, where c_0 is the negative class label. The maximum probability of SP predicted to negative is used as the threshold value t, i.e.,

$$t = max\{P(c_0|s_1), P(c_0|s_2), P(c_0|s_3), \cdots, P(c_0|s_k)\}$$

This equation indicates that those unlabeled examples, whose predicted probability of being negative are greater than t, are deemed as reliable negative examples.

Following the idea of the aforementioned PU learning algorithm with two-step strategy, we propose another K-means-based PU learning method. The main difference between K-means-based method and Spy-based method lies in the first step, i.e. the process of extracting RN. In K-means-based PU learning algorithm, firstly the examples from the unlabeled set U are grouped into k clusters with k-means algorithm. Then the distances of the k centroids from the centroid of P are calculated, and the clusters with centroids furthest from the centroid of P were selected as reliable negative examples, which constitute RN. The second step is to train a traditional classifier using P and RN, the same as Spy-based PU learning.

The detailed Spy-based and K-means-based PU learing algorithm are shown in Algorithm 1 and Algorithm 2 respectively.

IV. EXPERIMENT RESULTS

A. EXPERIMENTAL DATASET

As mentioned in Section III, the real traffic flow dataset is preprocessed and stored on Hadoop platform. There are more than 1 million users identified by mobile phone numbers, and more than 5000 App servers identified by IP addresses.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2852008, IEEE Access

Algorithm 1 PU Learning Algorithm based on Spy

Input: Positive examples set, P; Unlabeled examples set, U; Test set, T; α; β
Output: Prediction Label Vector $FL, \forall u \in T$;

```
initializing Reliable Negative examples set RN = ∅;
extracting Unlabeled Subset US = Sample(U, α);
extracting Spy Set SP = Sample(P, 1 − β);
labeling examples in P − SP +1;
labeling examples in US ∪ SP -1;
building classifier C1 with P − SP and US ∪ SP;
classifying each u ∈ US ∪ SP using C1;
determining the probability threshold t with SP;
for each u ∈ US do
    calculating probability p(−1|u) with classifier C1;
    if p(−1|u) > t then
        RN = RN ∪ u
    end if
end for
building classifier C2 with P and RN;
classifying each u ∈ T using classifier C2;
return Prediction Label Vector FL, ∀u ∈ T
```

Algorithm 2 PU Learning Algorithm based on K-means

Input: The set of positive examples, P; The set of unlabeled examples, U; Test set, T; Number of clusters k
Output: Prediction Label Vector $FL, \forall u \in T$;

```
initializing Reliable Negative set RN = ∅;
extracting Unlabeled Subset US = Sample(U, α);
clustering US into k clusters US1, US2, ··· , USk using
k-means algorithm;
calculating the center of each cluster c1, c2, ··· , ck;
calculating the center cp of P;
calculating and sorting the Euclidean distances between
cp and c1, c2, ··· , ck;
adding the top m clusters with maximum distance from
cp into RN;
building classifier C2 with P and RN;
classifying each u ∈ T using C2;
return Prediction Label Vector FL, ∀u ∈ T
```

We construct the User-App Bipartite Network as described in Section III-C.

For the App usage prediction task, we chose the most popular IM application in China, i.e. QQ. Since most mobile users would use IM Apps almost every day, this category accumulates abundant traffic flow data with a large coverage of users. Thus, we deem that QQ related records are most suitable for App usage prediction. We filtered QQ related traffic flow data as the experimental dataset. As one of the most commonly used instant messaging and social media applications in China, QQ also provides users with a wide range of services other than IM, including online social games, music, shopping, micro blogging, streaming videos, voice and video chat, and group chat. We extracted the flow data related to QQ from the 4-day dataset, and then we selected 5000 users at random and all traffic flow records belonging to these 5000 users are extracted. The experimental QQ dataset includes about 2.2 million records. With this filtered data and the feature selection method proposed in Section III-D, relevant characteristics of every user and server are calculated, and feature vectors for each user and server pair are generated. For the prediction problem of the connection relationship between QQ users and servers, the first-three-day data are used for training, and the fourth day for testing. Table 2 gives a summary of this experimental dataset.

TABLE 2: Experimental QQ Dataset [1]

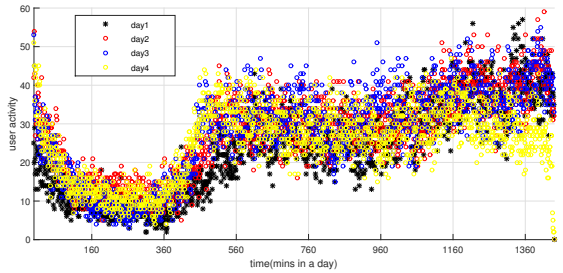
number of user nodes	5000	
number of server nodes	1648	
	day1+2+3 for training	day4 for testing
positive examples	235586	77835
unlabeled examples	8184414	8342165

Based on the experimental QQ Dataset, the User-App bipartite network is constructed and the basic graph properties are analyzed. When there are traffic flow records related to a mobile user or an App server, it is considered active in the time scale. The user activity and server activity are defined as the numbers of active mobile users and active App servers. In Fig. 5, the numbers of active mobile users and App servers with time change are presented. The horizontal axis represents 1440 minutes of a day. If there are traffic transmissions on each minute, the mobile user or the App server is regarded as active. We can see that the trends of user and App server activity are similar day by day. The number of active users and App servers reaches the lowest point between 2a.m.-5a.m. at night.

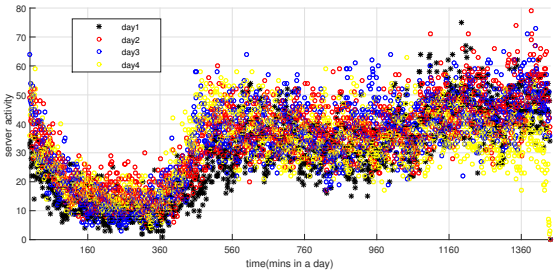
In Fig. 6, the distributions of in-degree and out-degree of mobile users and App servers are presented. Degree of user and server node means the number of connections in the graph. For example, the in-degree of a user node means the number of edges from the server to the user, while the out-degree of a user node means the number of edges from user to server. We can see that all the distributions are approximately power law with exponential cutoff.

For the mobile video traffic identification task, we filtered six popular Apps related traffic flow data as the experimental dataset. The six Apps are Youku, Baofeng, LeTV, Tudou, Meituan and Apple's App Store (abbreviated as Apple later). The first four are mobile video intensive applications. We distilled the flow data related to the six Apps from the real 1-day dataset. We use the flow records of Youku and Baofeng as the identified video traffic, i.e. positive labeled dataset. Flow record of the other four applications is used as unlabeled dataset, which includes both flow records from video intensive Apps (LeTV and Tudou) and other Apps (Meituan and Apple). To maintain the diversification of apps, the two non-video Apps are chosen from two totally different categories, i.e. group-buying/life convenience and mobile application

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2852008, IEEE Access

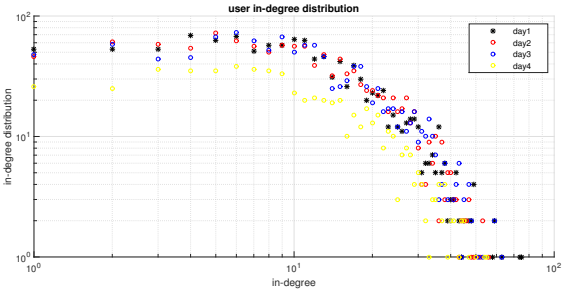


(a) User activity

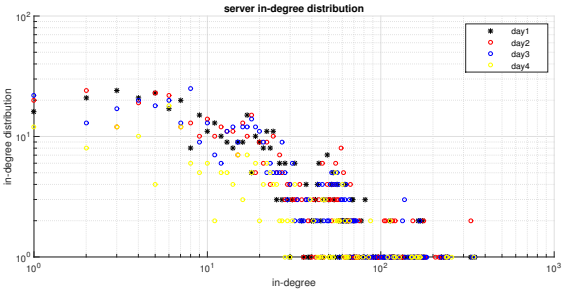


(b) Server activity

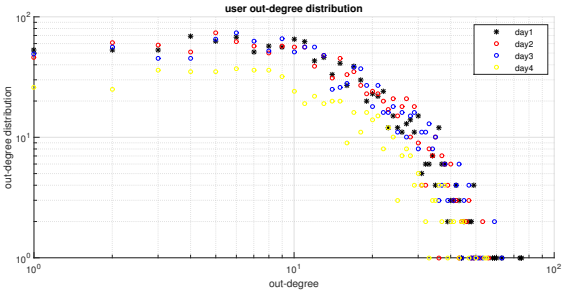
FIGURE 5: User and App Server Activity



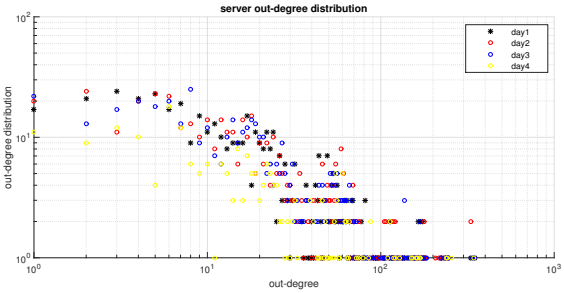
(a) User indegree



(b) Server indegree



(c) User outdegree



(d) Server outdegree

FIGURE 6: User and Server Degree

store, both of which have a large customer base. Page limit is another reason that we focus on only 6 applications for this task. The experimental Video Traffic dataset is summarized in Table. 3.

TABLE 3: Experimental Video Traffic Identification Dataset

App	Number of records
Youku	378220
Baofeng	354721
LeTV	51631
Tudou	13637
Meituan	125596
Apple	4692259

B. EVALUATION MEASURES

In our experimental results, the F score is used for evaluating the performance of the proposed PU learning method. F score has been extensively used as a metric of PU learning technique performances in these study [18] [21] [27] [35]. F score is defined as, $F = 2pr/(p+r)$, where p is the precision and r is the recall. F score presents the average effect of both precision and recall. F score will be small when either of precision or recall is not large, and will be large only when both of them are large. Due to the above property of F score, it is suitable for our task, since we aim at identifying positive examples. Either too small precision or too small recall is undesirable.

We also present accuracy as a metric in the experiments. The accuracy is define as the proportion of examples that are correctly classified in all the examples. However, it should

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2852008, IEEE Access

be noted that accuracy does not fully reflect the performance of our algorithm, as there are a large proportion of negative examples in our experimental datasets. In such cases, the accuracy can be high, but few positive examples may be identified.

C. EXPERIMENTAL RESULTS

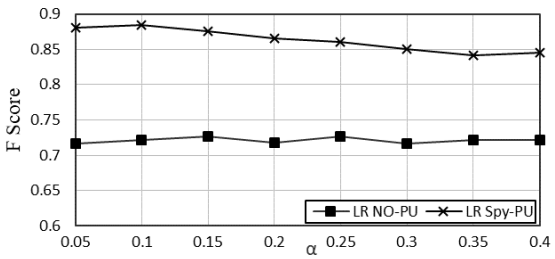
1) App Usage Prediction Task

In App usage prediction experiment, firstly, we select the baseline method. In the baseline method, unlabeled examples are sampled randomly and then used as negative examples directly. To complete the prediction task, a traditional binary classifier is trained using these negative examples, together with the existing positive examples. We compare the baseline method, which does not adopt PU learning (NO-PU), and our Spy-based PU learning algorithm (Spy-PU) for prediction.

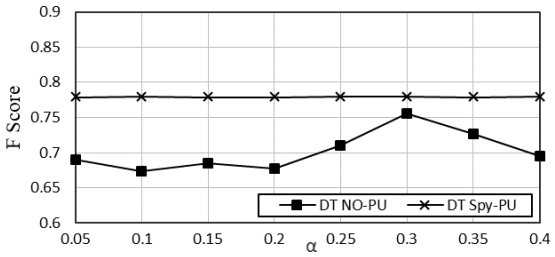
Secondly, in the Spy-based PU learning method, which classifier to adopt needs to be decided. This step is of critical importance, because the final results is severely impacted by the classifier. So for the binary classifier, we use Random Forest (RF), Decision Tree (DT), and Logistic Regression (LR), and compare their performances.

Thirdly, we compare the two important parameters in our proposed Spy-based PU learning method. Among all the user and server pairs, only a quite small proportion will have records of connection in the User-App Network, i.e. there is merely a small number of labeled positive examples. The data imbalance between labeled positive examples and unlabeled examples poses severe impact on the later training and prediction steps. In our Spy-based PU learning method, we propose a method that randomly samples the unlabeled data with sampling ratio α. Intuitively, to train our PU learning models better, the α value needs to be set relatively small, to mitigate data imbalance. We investigated the effects of α on learning performance from 0.05 to 0.4, with a step size of 0.05. To get reliable negative examples, we propose the sampling of a spy set from positive set, and the sampling ratio is $1-\beta$, i.e. $1-\beta$ of the positive examples set is put into the unlabeled set. Since we only have small amount of labeled positive data, the β vale intuitively needs to be set large. We also investigated the effects of β on learning performance, from 0.2 to 0.9, with a step size of 0.1 (Fig. 9).

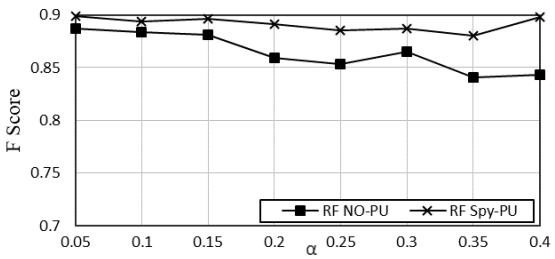
In Fig. 7, the F score performances of the baseline method and our proposed method with Spy-based PU learning are compared. From Fig. 9, we can see that three classifiers have near optimal performances when β equals to 0.8, so β is set to 0.8, while α is set to 8 different values, i.e. 0.05, 0.10, 0.15, 0.20, 0.25, 0.30, 0.35 and 0.40. From Fig. 7, it can be seen that our proposed Spy-based PU learning method achieves a better result on the App usage prediction task. Regardless of classifier used in the algorithm, the proposed Spy-based PU learning method obtains F score between 0.75 and 0.9. By comparing the performances of three classifiers, Random Forest works better than Logistic Regression and Decision Tree. The different values of α have little effects on the final F score performance results of Spy-based methods, while for three baseline methods without PU learning, the influence of parameter α varies depending on which classifier is adopted. Different classifiers express different sensitivity to α. For LR, F score decreases with Îś increasing, which is theoretically reasonable since LR are sensitive to unbalanced dataset. For DT, α value has some slight influence on the prediction result, but not too much in general. For RF, the performance gets worse with α increasing, indicating that RF is also sensitive to data unbalance. From experimental results, we conclude that Spy based PU learning outperforms all three classifiers without PU learning.



(a) Logistic Regression (β=0.8)



(b) Decision Tree (β=0.8)



(c) Random Forest (β=0.8)

FIGURE 7: F score performance of LR, DT and RF with Spy-PU and without PU on App usage predicton task [1]

The accuracy performances of the basesline method without PU learning and our proposed method with Spy-based PU learning are presented in Fig. 8. We can see our proposed Spy-based PU learning method is superior to the baseline method without PU learning. Random Forest proves to be the most suitable classifier in the experiments.

In Fig. 9, we kept the parameter α to be 0.15, and investigated the effects of different β values on F score perfomance. It can be seen that Random Forest with Spy-based PU learning method outperforms Logistic Regression and Decision Tree, obtaining F score about 0.9 under different β values

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2852008, IEEE Access

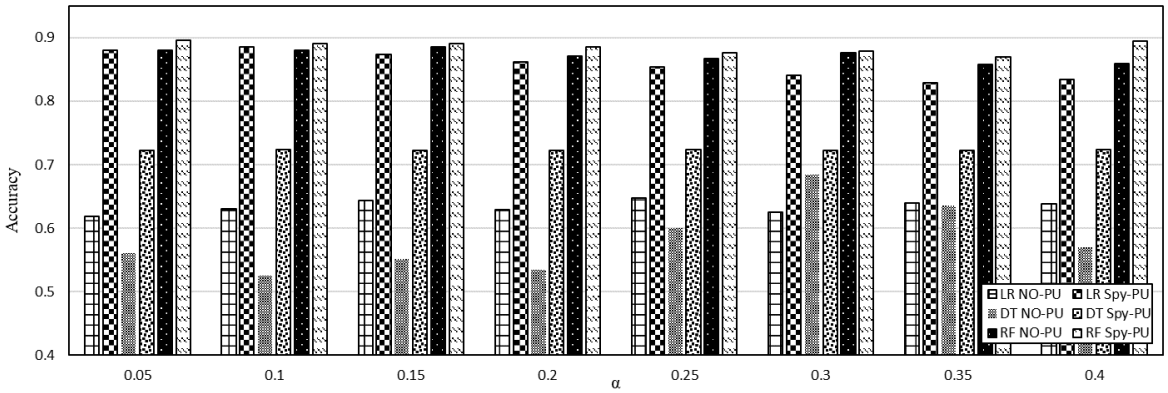


FIGURE 8: Accuracy performance of LR, DT and RF with Spy-PU and without PU on App usage predicton task.

between 0.2 and 0.9. For Logistic Regression with PU learning method, F score reaches the highest when β is 0.7. For Decision Tree with PU learning method, F score increases as the β increases. Moreover, the accuracy performances of the Spy-based PU learning method under different β values are presented in Fig. 10. We can see from Fig. 9 and Fig. 10 that the value of β has more influence on the prediction results than α. A larger β means a lower proportion of examples extracted from positive records to be Spy. When the amount of records in Spy set is more properly chosen, it will be more conducive to mine as much reliable negative examples as possible, contributing to better prediction results.

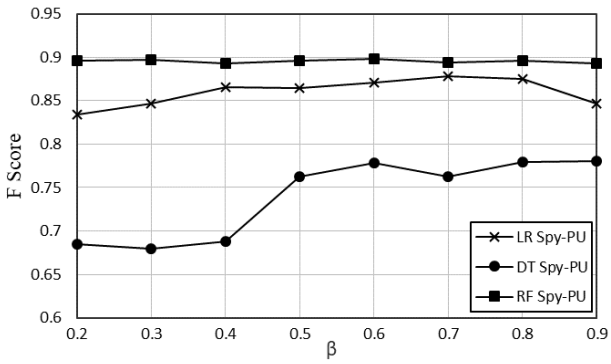


FIGURE 9: F score performance of Spy-based PU learning method under different β values on App usage predicton task [1]

We also did experiment with K-means-based PU learning algorithm. As described in Algorithm 2, the number of clusters k is an important parameters in the step of extracting reliable negative examples. We investigated the effect of different values of k on F score and accuracy performance, as shown in Fig. 11 and Fig. 12. It can be seen that our proposed K-means-based PU learning method produces a fairly satisfying result on the task of App usage prediction, with F score between 0.65 and 0.85. We can also see that Linear Regression has the best performance when K-means-based PU learning is adopted. The value of k has great impact on the prediction results when using Linear Regression as the classifier. Larger k, i.e., more clusters, means better separability, and as a consequence the extracted negative examples are more reliable.

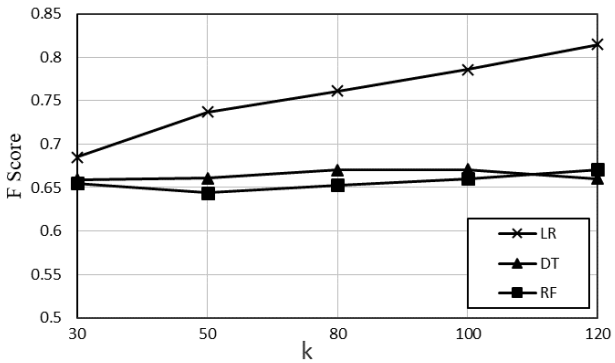


FIGURE 11: F score performance of K-means-based PU learning on App usage prediction task

To sum up, for the App usage prediction task, Spy-based PU learning with RF classifier has the best performance with respect to both F score and accuracy, and K-means-based PU learning methods are inferior to Spy-based methods in general. Still, K-means-based PU learning with LR classifier when k equals to 120 gets relatively satisfying result, where F score exceeds 0.8 and accuracy exceeds 0.78.

2) Mobile Video Traffic Identification Task

In mobile video traffic identification experiment, we consider a Spy-based and K-means-based PU learning as a comparison. They both consist of two steps. We compare the two PU learning methods in the experiments.

For the binary classifier, same as in the App usage prediction task, we also use Logistic Regression, Decision Tree, and Random Forest, and compare their performances.

In the experiment, the sampling ratio of the unlabeled data, α, is set to 1, i.e. without sampling, because the dataset of

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2852008, IEEE Access

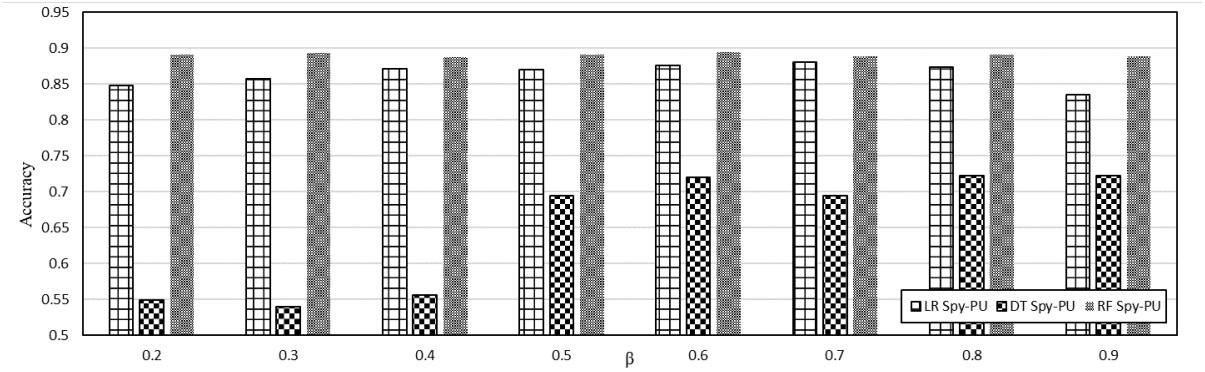


FIGURE 10: Accuracy performance of Spy-based PU learning method under different β values on App usage predicton task

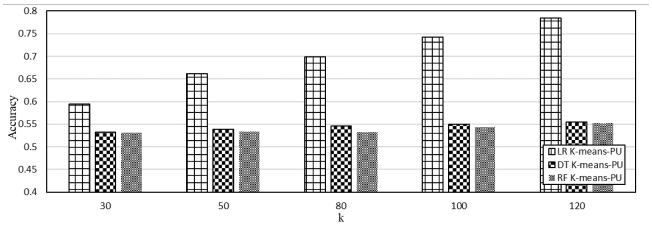


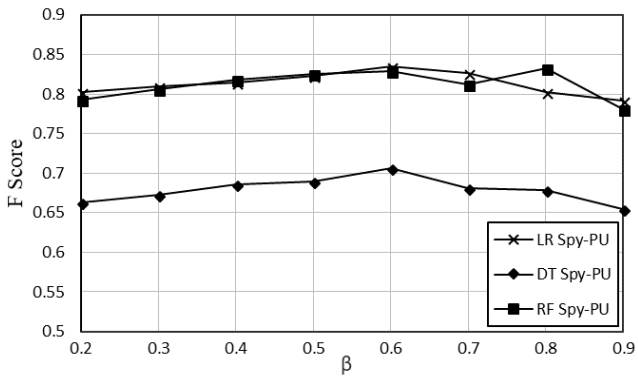
FIGURE 12: Accuracy performance of K-means-based PU learning on App usage prediction task

unlabeled record is not very large. We studied the effects of different β values on experimental results from 0.2 to 0.9 with step size 0.1.

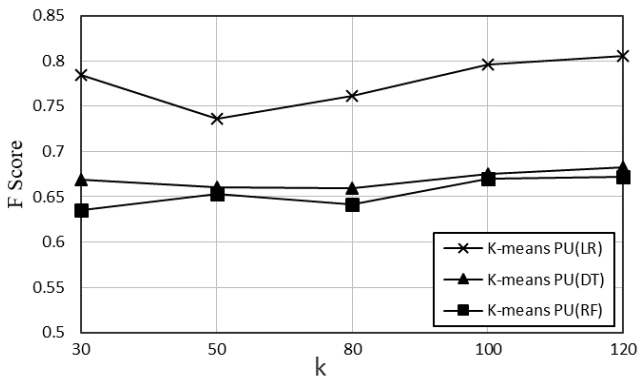
In Fig. 13, F score performances of two PU learning methods are compared. For Spy-based PU learning, we can see Random Forest and Logistic Regression performs similarly for video traffic identification task, while decision tree is inferior to the above two classifiers. The effect of β values on F score performance is not obvious. For K-means-based PU learning, Logistic Regression performs better than Random Forest and decision tree. The k values partially affect the F score performance.

To sum up, for the video traffic identification task, Spy-based PU learning with RF and LR classifiers have similarly the best performances with respect to both F score and accuracy. K-means-based PU learning methods with LR when k is 120 also achieves satisfying result. However, Spy-based PU learning methods with DT classifier and K-means-based PU learning methods with DT and RF classifiers are much inferior to other methods, no matter what values of α and β are set.

The accuracy performances of the two PU learning methods under different β or k values are presented in Fig. 14 and Fig. 15. For Spy-based PU learning, Linear Regression with $\beta = 0.5$ performs best; for K-means-based PU learning, logistic Regression with $k = 120$ performs the best.



(a) Spy-PU



(b) K-means-PU

FIGURE 13: F score performance of LR, DT and RF with Spy-PU and K-means-PU on mobile video traffic identification task

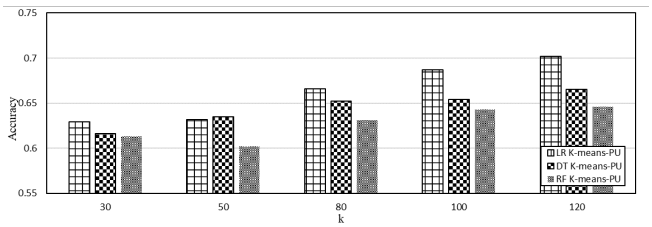


FIGURE 15: Accuracy performance of K-means-based PU Learning on video traffic identification task

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2018.2852008, IEEE Access

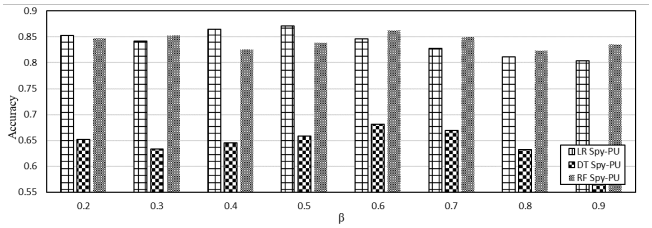


FIGURE 14: Accuracy performance of Spy-based PU Learning on video traffic identification task

V. CONCLUSION

This paper investigates the mobile user behavior based on real life mobile Internet traffic data, specifically for human-centered smart service provisioning. By considering the App usage prediction as well as mobile video traffic identification task, we propose a framework combining data preprocessing and machine learning to accomplish the tasks. Utilizing the real mobile Internet traffic data, we elaborate how we construct User-App bipartite network and choose proper features from the network. And then we propose two PU algorithms based on two-step strategy for learning tasks, i.e. Spy-based and K-means-based PU Learning. We use real QQ traffic flow data and mobile video traffic flow data to testify the efficacy of the proposed algorithm. Experimental results show that the proposed PU prediction methods can improve both efficiency and accuracy on classifiers with regard to F score and accuracy.

For the future work, the complexities of proposed learning algorithms will be analyzed. And streaming framework will be implemented for large-scale mobile Internet traffic dataset.

REFERENCES

[1] B. Wang, K. Yu, X.F. Wu, et al. "Positive and Unlabeled Learning for Mobile App User and Server Interaction Prediction," International Conference on Communicatins and Networking in China,, Springer, Cham, 2017, pp. pp. 481-491.

[2] A. Kamilaris, A. Pitsillides, "Mobile phone computing and the Internet of Things: A survey," IEEE J. Internet of Things., vol. 3, no. 6, pp. 885-898, 2016.

[3] Z. X. Liao, S. C. Li, W. C. Peng et al., "On the feature discovery for app usage prediction in smartphones," in IEEE 13th ICDM, Dallas, TX, USA, 2013, pp. 1127-1132.

[4] Z. X. Liao, P. R. Lei, T. J. Shen et al., "Mining temporal profiles of mobile applications for usage prediction," IEEE 12th ICDMW, Brussels, Belgium, 2012, pp. 890âĂŞ893.

[5] J. Quittek, T. Zseby, B. Claise et al., "Requirements for ip flow information export (IPFIX),", IETF RFC 3917, 2004.

[6] J. Yang, y. Qiao, X. Zhang, et al.,"Characterizing user behavior in mobile internet,"IEEEtran J. Emerging topics in Computing., vol. 3, no. 1, pp. 95-106, 2014.

[7] Z. Li, G. Xie, M. A. Kaafar et al., "User behavior characterization of a large-scale mobile live streaming system," in Proc. 24th Int. Conf. World Wide Web, ACM, Florence, Italy, 2015, pp. 307-313.

[8] Y. N. Ravari, I. Markov, A. Grotov et al., "User behavior in location search on mobile devices," Eu. Con. Information Retrieval, Springer, Cham, 2015, pp. 728-733.

[9] L. Xiao, Y. Li, G. Han et al., "A Secure mobile crowdsensing game with deep reinforcement Learning," IEEE Trans. Information Forensics & Security, vol. 13, no. 1, pp. 35-47, Jan. 2018.

[10] Y. Mo, J. Chen, X. Xie et al., "Cloud-based mobile multimedia recommendation system with user behavior information," IEEE Systems Journal, vol. 8, no. 8, pp. 184-193, 2014.

[11] X. Zhang, C. Wang, Z. Li et al., "Exploring the sequential usage patterns of mobile Internet services based on Markov models," Electronic Commerce Research and Applications, vol. 17, pp. 1-11, 2016.

[12] K. Huang, C. H. Zhang, X. X. Ma, and G. L. Chen, "Predicting mobile application usage using contextual information," Proc. ACM Conf. Ubi. Comp., Pittsburgh, PA, USA, 2012, pp. 1059âĂŞ1065.

[13] C. S. Shin, J. H. Hong, and A. K. Dey, "Understanding and prediction of mobile application usage for smart phones," Proc. ACM Conf. Ubi. Comp., Pittsburgh, PA, USA, 2012, pp. 173âĂŞ182.

[14] A. L. Barabasi, R. Albert, "Emergence of scaling in random networks," Science J. vol. 286, no. 5439, pp.509, 1999.

[15] W. Jiang, M. Gokhale, "Real-time classification of multimedia traffic using FPGA," IEEE Computer Society, Int. Conf. FPL., Milano, Italy, 2010, pp. 56-63.

[16] F. Hao, M. Kodialam, T. V. Lakshman, "On-line detection of real time multimedia traffic," IEEE Computer Society, Int. Conf. Network Protocols, Princeton, NJ, USA, 2009, pp. 223.

[17] K. YU, Y. Cao, X. H. Huang, X. F. Wu, "Entropy analysis for inter-domain Internet application flows," The Journal of China Universities of Posts and Telecommunications 18, 2011, pp. 54-60.

[18] B. Liu, Y. Dai, X. L. Li, W. S. Lee, and P. S. Yu, "Building text classifiers using positive and unlabeled examples," IEEE 3rd Int. Conf. Data Mining, Melbourne, FL, USA, 2003, pp. 179âĂŞ186.

[19] W. S. Lee, B. Liu, "Learning with positive and unlabeled examples using weighted logistic regression," Int. Conf. Machine Learning, vol. 3, pp. 448âĂŞ455, 2003.

[20] Y. C. Zhao, X. N. Kong, and P. S. Yu, "Positive and unlabeled learning-for graph classification," IEEE 11th Int. Conf. Data Mining, Vancouver, Canada, 2011, pp. 962âĂŞ971.

[21] X. L. Li, P. S. Yu, B. Liu, and S. K. Ng, "Positive unlabeled learningfor data stream classification," Proc. SIAM. Int. Conf. Data Mining, Denver, CO, USA, 2009, pp. 259âĂŞ270.

[22] Y. F. Ren, D. H. Ji, and H. B. Zhang, "Positive unlabeled learning for deceptive reviews detection," EMNLP, Doha, Qatar, 2014, pp. 488âĂŞ498.

[23] P. Yang, X. L. Li, J. P. Mei, C. K. Kwoh, and S. K. Ng, "Positive-unlabeled learning for disease gene identification," Bioinformatics, vol. 28, no. 20, pp. 2640, 2012.

[24] H. Li, B. Liu, A. Mukherjee et al., "Spotting fake reviews using positive-unlabeled learning," ComputaciÃşn y Sistemas, 2014, vol. 18, no. 3,pp. 467-475, 2014.

[25] W. Lan, J. Wang, M. Li et al., "Predicting drugâĂŞtarget interaction using positive-unlabeled learning," Neurocomputing, vol. 206, pp. 50-57, 2016.

[26] B. Z. Zhang, W. L. Zuo, "Learning from positive and unlabeled examples: a survey," IEEE ISIP, Moscow, Russia, 2008, pp. 650âĂŞ654.

[27] B. Liu, W. S. Lee, P. S. Yu, and X. L. Li, "Partially supervised classification of text documents," ICML, Citeseer, vol. 2, pp. 387âĂŞ394, 2002.

[28] H. Yu, J. W. Han, and K. C. Chang, "Pebl: positive example based learning for web page classification using svm," Proc. 8th ACM. SIGKDD., Edmonton, AB, Canada, 2002, pp. 239âĂŞ248.

[29] X. L. Li and B. Liu, "Learning to classify texts using positive and unlabeleddata,". IJCAI, vol. 3, pp. 587âĂŞ592, 2003.

[30] G. Fung, J. X. Yu, H. J. Lu, and P. S. Yu, "Text classification without negative examples revisit," IEEE Tran. Knowledge and Data Engineering, vol. 18, no. 1, pp.6âĂŞ20, 2006.

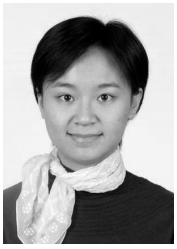
[31] F. Denis, R. Gilleron, and M. Tommasi, "Text classification from positive and unlabeled examples," Proc. 9th Int. Conf. IPMU'02., pp. 1927âĂŞ1934, 2002.

[32] F. Denis, A. Laurent, R. Gilleron, and M. Tommasi, "Text classification and co-training from positive and unlabeled examples,"Proc. ICML workshop: the continuum from labeled to unlabeled data, Washington D.C., USA, 2003, pp. 80âĂŞ87.

[33] A. B., and T. Mitchell, "Combining labeled and unlabeled data with co-training,". Proc. 11th ACM. Ann. Conf. Computational Learning Theory, 1998, pp. 92âĂŞ100.

[34] D.Zhang, and W. S. Lee, "A simple probabilistic approach to learning from positive and unlabeled examples," Proc. 5th UKCI, 2005, pp. 83âĂŞ87.

[35] J. Z. He, Y. Zhang, X. Li, and Y. Wang, "Naive bayes classifier for positive unlabeled learning with uncertainty," Proc. SIAM. Int. Conf. Data Mining, Pittsburgh, PA, USA, 2010, pp. 361âĂŞ372.



KE YU is an associate professor with the School of Information and Communications Engineering at Beijing University of Posts and Telecommunications (BUPT), China. She received the B.S. degree in Computer Science in 2000, and the Ph.D. degree in Signal and Information Processing in 2005, both from BUPT. In 2011, she held visiting position at University of Agder, Norway. From 2015 to 2016, she worked as a visiting scholar at University of Illinois at Chicago, USA. Her current research interests include communication network theory, network data mining, mobile Internet application, machine learning and human-machine intelligence.

YUE LIU is currently a master student in Beijing University of Posts and Telecommunications (BUPT), China. She received her B.E. degree from Harbin Institute of Technology at Weihai in 2017. Her research area includes data mining, machine learning and Internet service provisioning.

LINBO QING (M'16) is an associate professor with the College of Electronics and Information Engineering at Sichuan University. He received his B.S. degree in electronic Information Science and Technology from Sichuan University, China in 2003. In 2008, he received his Ph.D. degree in Communication and Information System from the same university. His main research interests include machine learning, image processing, video coding and transmission, and information theory. He is the corresponding author of this paper.

BINBIN WANG is currently a master student in Beijing University of Posts and Telecommunications (BUPT), China. He received his B.E. degree from University of Electronic Science and Technology of China in 2015. His research area includes machine learning, big data analysis and mining.

YONGQIANG CHENG is currently a Lecturer with the School of Engineering and Computer Science at the University of Hull, UK. His research interest includes digital healthcare technologies, embedded systems, control theory and applications, AI and data mining.