A novel classification model of collective user web behaviour based on network traffic contents

Hongri Liu\textsuperscript{1,2,3} | Shuo Wang\textsuperscript{1,3} | Yuliang Wei\textsuperscript{1,3} | Bailing Wang\textsuperscript{1,3}

\textsuperscript{1}College of Computer Science & Technology, Harbin Institute of Technology at Weihai, Weihai, China
\textsuperscript{2}Cyberspace Security Research Center, Peng Cheng Laboratory, Shenzhen, China
\textsuperscript{3}Research Institute of Cyberspace Security, Harbin Institute of Technology, Harbin, China

Abstract

Web behaviour analysis of a collective user has provided a powerful means for studying the collective user interests on the Internet. However, the existing research merely analyses the behaviour of a single user who accesses multiple applications or multiple users who access one application. The authors propose a web behaviour classification model for collective user, in which the title fields in HTTP flows are extracted from the mirrored network traffic that has been already captured for any given period of time. The title fields, considered as the short abstract of the whole web pages browsed by the users, are vectorized by natural language processing technologies. Specifically, the Latent Dirichlet allocation (LDA) algorithm is used to calculate the topic distribution probability matrix. Afterward, the multi-class classifiers are trained and tested using the manually labelled probability distribution matrix from the output of the LDA algorithm to classify the user behaviour topics. The experiments demonstrate that the highest classification accuracy of the model reaches 81.2\% by combing the LDA algorithm with Random Forest classifiers under the classification model.

1 | INTRODUCTION

Collective behaviour refers to the relatively spontaneous and relatively unstructured behaviour of a group of people acting with or being influenced by each other [1]. Collective web behaviour is one form of collective behaviour, which is the interaction between a group of people and browsers [2]. Widely used in network measurement and network environment simulation [3], collective web behaviour classification is a method used to classify the web accessing behaviour of a group of users in a continuous period. The web behaviour from the classification results of a collective user can also be generalized and applied in many critical industrial applications, such as search engines [4], user behaviour modelling [5] and recommendation systems [6]. The classification results could also be used to provide data support for the collective user behaviour simulation [7] in all kinds of network testbeds, for example, National Cyber Range [8]. In order to realistically simulate the web behaviour of a group of users over a period of time, we classify the behaviours into six categories, namely, video, music, social network/blog, news, Internet information retrieval, and e-commerce according to the content of user's accessing objects mentioned in [9], which have covered almost all the behaviours of users' access to web objects. The six categories that we use are distinct from each other. We further summarize the research methods for the analysis of user web behaviour into but not limited to the following three types:

(1) Web-content-based analysis method. This method analyses user behaviour by mining text, HTML and other text formats on a typical business management website. For example, for incorporating content, structure and usage data from web sources, Loyola et al. [10] analysed web users' behaviour with an Ant Colony Optimization algorithm. Awad et al. [11] predicted the next set of web pages to be visited by a user based on the Markov model.
Web-connection-based analysis method. In the Social Network, connections are abstracted into user relationship and interactions [12–14], which have been widely used to mine and analyse the evolution of social networks, for example, Al-Garadi et al. [13] predicted the probability of a connection occurring at a specific time using time-series models of single-link occurrences.

Log-based analysis method. In this method, the users’ logs, collected from servers [15–17] or clients [18–20], are mapped into terminal operation behaviour. The researchers explored the interactions of users with existing systems to study the users’ most concerned issues and to understand users’ accessing intent further.

Although the methods mentioned above can analyse the user web behaviour either for a single user or for a single website, the calculation of collective user web behaviour is still a challenge in terms of data collection and classification methods. On the one hand, it is difficult to collect large-scale users’ sessions or cookies from clients, and on the other hand, most of the necessary data for web logs analysis is saved in web servers, proxy servers or enterprise log servers, and so forth. Therefore, it is almost impractical to collect logs from them due to the servers’ security policy.

Table 1 describes the meanings of various abbreviations and acronyms used throughout this thesis. The page on which each one is defined or first used is also given.

Inspired by the web-content-based analysis methods mentioned above and to overcome the above limitations, we propose a new model for classifying the web behaviour of a collective user (CWBCU) based on network traffic contents, which makes use of network traffic to analyse the topics of web behaviour of collective users. To classify the topics the web behaviour of the collective user, in our research, we acquire network traffic from a network device and then employ our proposed model, which combines the LDA algorithm [21] with a multi-class classifier [22]. Based on the CWBCU model, we can classify the topics of web behaviour of the collective user into the six categories from network traffic.

In summary, the main contributions of our work are listed as follows:

1. We propose an algorithm to extract web page names (i.e. title fields in an HTTP flow) from pcap files, by first extracting HTTP flows from a pcap file and then splitting the flows and finally saving the results into HTML files. The web page names are acquired by analysing the header of the HTML files.
2. An algorithm is designed to classify the topics of collective user web behaviour. First, we generate a document-word frequency matrix via analysing the web page names using the techniques in NLP. Then, the matrix is transformed into a probability matrix by the LDA algorithm. Finally, the multi-class classifiers are trained with the probability matrix, which should be labelled by using the topics of user behaviour types. The trained model can be used to classify the topics of collective user web behaviour.
3. We conduct extensive experiments on the CWBCU model with three implementations, the LDA algorithm with the Random Forest classifier, the LDA algorithm with the Extreme Gradient Boosting (XGBoost) classifier [23] and the LDA algorithm with the Naïve Bayesian classifier, to evaluate the performance of the model in classification accuracy. All the three implementations have been evaluated, respectively. The experimental results on a real-world dataset show that our proposed three methods from the classification model can relative accurately classify the topics of web behaviour of a collective user from network traffic contents. Moreover, experimental results demonstrate that the LDA algorithm with a Random Forest classifier achieves the highest classification accuracy.

The rest of this paper is organized as follows. Section 2 provides a summary of related work, and the CWBCU model is proposed in Section 3. Section 4 outlines the experimental results on the dataset followed by Section 5, which provided some conclusions and suggestions for future research.

2 | RELATED WORK

Web behaviour calculation of a collective user is one of the most important means of understanding users' intentions, which provides data support for collective user behaviour simulation. The analysis of behaviour of a collective user is more important than that of a single behaviour in many ways as shown in the following scenario. For example, in using the platform Cyber Range, a realistic network environment needs to be established, and this environment includes building the foreground streams. The foreground streams are generated by simulating user behaviours and in our experiment, for a collective user as a unit, we can get its behavioural topics in percentage, for example, 20% of the behaviours belongs to music. The simulation of collective user behaviour can be performed using the platform Cyber Range according to this percentage to generate foreground streams and so that the simulation environment is more realistic. However, the foreground streams generated by analysing single user's behaviours are less realistic.
Currently, most of the research focuses on the study of web-service-based user behaviour analysis, while our study based on network traffics.

2.1 User web behaviour analysis

Based on the assumption that web behaviour is dependent on the type of query, Park et al. [24] analysed user search behaviour in a large-scale query log from Yahoo Image Search, in which two orthogonal classifications are employed to classify queries and identify consequential query types. Considering the sequential aspects of a web user session and taking the sequential information of web pages into account, Mishra et al. [25] proposed a recommendation system for web users. Lopes et al. [26] did similar research, in which a real-time dynamics recommendation system with web mining techniques was constructed to analyse web logs. Also, based on the knowledge of the previously accessed pages, Awad et al. [11] predicted the set of web pages that a user might access, and a new modified Markov model is proposed to reduce the memory requirements. Web logs were also used to assess user learning behaviour and cognition, for example, Rao et al. [27] collected mouse click and movement data during human-machine interaction, and then applied statistical and data mining techniques to model users' learning behaviours. The results of the model were used to predict possible data cognitive bias.

In a nutshell, the methods mentioned above can be used to analyse or infer user web behaviour in some interesting applications. However, they are not suitable for analysing the topics of collective web behaviour or topic distribution during any period of time. On the one hand, the data such as web logs, connections and mouse movements are collected from only some concerned web applications; on the other hand, their analysis seldom involves web content. In our study, for data collection, we adopt the network traffic captured from the managed switches instead of web logs. With respect to content analysis, the LDA algorithm is employed to calculate the collective behaviour topics by analysing the name of the web pages that the users accessed.

The LDA is a probability generation model, in which each document is viewed as a mixture of a small number of topics in a corpus, and each topic is viewed as an infinite mixture over a set of topics probabilities [22]. Krestel et al. [28] applied the LDA-based method to elicit a shared topical structure from a recommending resource to solve the tags’ recommendation problem, which shows this method achieves better accuracy and recall rate than using association rules in their evaluation.

Chen et al. [29] proposed the Forum-LDA to model the root post-process based on the assumption that the subject of the reply is determined only by the author’s serious subjects and root posts. The model is capable of not only learning more coherent topics and serious interests, but also identifying unserious users who posted irrelevant replies. Guo et al. [30] employed the LDA algorithm to identify the critical dimensions of users’ evaluation of hotel services and obtained a total of 19 controllable dimensions, which are the key to hotel management.

2.2 Network traffic analysis

Network traffic analysis is the process of inspecting and analysing network traffic for performance, security, or general network management. Supervised multi-class classifiers, assigning each sample to one and only one class label, are broadly used to perform traffic classification, for example, Random Forest is used to traffic identification [31]. Nair et al. [32] proposed an improved decision tree classifier to detect P2P traffic. Guy et al. [33] integrated a Random Forest classifier into the AdaBoost algorithm to perform traffic congestion prediction, and the error rate is lower than the naive predictors. In [34], Random Forest classifiers were also used to detect the presence of domain name system tunnels in a complex defence-in-depth enterprise environment, and their experimental results showed that the classifiers detected not only three types of tunnels that were in training dataset, but also four new types of tunnels that were not in the training dataset.

To sum up, the methods mentioned above merely studied traffic classification by using machine learning algorithms trained and evaluated by annotated datasets. However, the traffic contents have not been analysed. Some researchers focus on the studies of learning specific actions that a user is doing via network traffic analysis [35, 36]. For example, Conti et al. [35] designed a system to identify user’s actions for mobile devices via network traffic by implementing machine learning classifiers. Furthermore, in [37], fine-grained user activities were identified within the wireless network traffic generated by some APPs based on traffic behaviour. In contrast, our work focuses on classifying the topics of web behaviour of a collective user while not analysing the specific actions. To the best of our knowledge, our research is the first work on studying collective user behaviour by using network traffic contents, in which the title fields from HTTP traffic are aggregated into multi-category documents using the LDA algorithm and the probability matrix under each category is exported. Considered as an eigenvalue of the topics, the probability matrix has been manually annotated by the six categories of user behaviour topics. Finally, supervised machine learning algorithms are introduced to perform a multi-class classification. The trained model is capable of calculating the topics of web behaviour of the collective user from the contents of network traffic captured from most network devices, such as managed switches and firewalls.

3 MODEL DESCRIPTION

In this section, we define and describe our proposed CWBCU model, which is composed of three layers, the pre-processing layer, the encoding layer and the computing layer, as shown in Figure 1.

The network traffic contents are processed by these three layers in the given order, and finally, web behaviour topics of a collective user are classified by the model. In Figure 1, network traffic, in the form of pcap files, is the input for the pre-
processing layer, which performs extraction of the web page names from the pcap files acquired by bypassing network traffic from network devices. Each web page name is considered as a document, which implies the behaviour topics of the user who is browsing the websites. The document is pre-processed by NLP techniques and technologies and then a document-word frequency matrix is created, which serves as the input to the LDA algorithm.

The encoding layer transforms the document-word matrix into a probability distribution matrix for each topic of web behaviour. The probability distribution matrix is considered as the eigenvectors of web page names and labelled manually with the six categories of topics, which will be used in the computing layer as a dataset to train and test the classifiers. The LDA algorithm cannot directly compute the web behaviour topic of each web page name; therefore, supervised machine learning algorithms are necessary for web behaviour computing.

In the computing layer, the dataset is used to train the supervised classifiers and test the classification accuracy of the classifiers applied in the computing layer iteratively, and finally, the best model is used to classify the topics of web behaviour of the collective user from traffic network contents. It should be noted that any supervised multi-class classifier can be applied in the computing layer, for example, Support Vector Machine (SVM) or Random Forest. The trained classifier will be employed in the CWBCU model to calculate the web behaviour of a collective user.

Overall, by extracting valid information from the network traffic contents, the CWBCU model can be used for classifying the topics of collective web behaviour in a continuous period that passes through pre-processing layer, encoding layer and computing layer. The calculated result can be widely used in user web behaviour simulation and user interest analysis. A detailed description of the three implementations of the CWBCU model is in the following sections.

### 3.1 Data collection and extraction mechanism

In this section, firstly, we introduce the data collection method from network flow; and then propose an algorithm to extract interested data, the web page names, from the traffic data.

The network traffic is captured in managed switches by port mirroring, as shown in Figure 2, in which all the packets passing through the managed switch are bypassed to the storage server and saved as pcap files.

In Figure 2, the traffic generated by any device in the network will be captured by the managed switch. However, to extract the web page names from the traffic, we have to merge TCP packets into TCP flows through a quintuple < timestamp, source IP, source port, destination IP, destination port>. The quintuple reveals which source IP has accessed which service (destination port) in which server (destination IP) at what time (timestamp). All the fields in the quintuple can be directly obtained by parsing the pcap files. When a user accesses a web page with a URL by a web browser, the web browser sends an HTTP request to server. Afterward, the server responds with an HTML file to the client. The title tag (web page name) displayed at the top of the web browser, which implies the user's network behaviour topics, can be obtained by parsing the HTTP response packets from a pcap file. In detail, when responding to the request, the server performs chunked transfer-encoding and GZIP compression on the HTML file and then sends them to destination IP chunk by chunk in order to enhance the data confidentiality and improve data transmission efficiency.

To extract web page names from a pcap file, we first detect HTTP response packets according to the HTTP status codes (e.g. HTTP 200 OK), and meanwhile, search the packet whose content-type is 'text/html' in HTTP flows. Next, we seek the next packet in the same HTTP flow according to the ACK number and sequence number in TCP header until the PSH value is 1. Finally, the packets are merged into one HTTP flow.

![Figure 1](image1.png)  
**Figure 1** The description of the CWBCU model

![Figure 2](image2.png)  
**Figure 2** Illustration of traffic capture approach with port mirroring
A detailed procedure for extracting web page names from a pcap file is shown in Algorithm 1.

Algorithm Extracting web page names from a pcap file

Input: the pcap file.
Optimization:
1. read the pcap file;
2. for each packet in the pcap file do
3. {
4. if the packet is an HTTP response header and content-type field is "text/html"
5. {
6. find all the packets and merge them into one HTTP flow;
7. if the transfer-encoding field is "chunked"
8. {
9. combine all the chunks to an HTTP flow;
10. }
11. if the content-encoding field is "Gzip"
12. {
13. decompress the HTTP flow;
14. }
15. extract {title_name, start_time, end_time, source_IP, source_port, destination_IP, destination_port} field from the HTTP flows and append it to the title_infor;
16. }
17. }
18. return title_infor;
Output: title_infor, which is a matrix to store the web information extracted from the pcap file.

In the matrix title_infor, each row vector is composed of title_name, start_time, end_time, source_IP, source_port, destination_IP and destination_port in sequence. Furthermore, the start_time means what time the HTTP flow is established and the end_time indicates what time the HTTP flow disconnects. Also, Keep-alive connections in HTTP/1.1 allow a client and a server to send and receive multiple HTTP requests and responses by the same TCP connection, respectively. Therefore, there are multiple web page names in one TCP flow with the same quadruple < source_IP, source_port, destination IP, destination port > in HTTP/1.1 during a period of time.

For training data, we capture the traffic and save it, then extract the titles of the pages in it as short text. Then, data tagging is done by a group of nine people, in groups of three, with each person first tagging the well-recognized titles, and the uncertain titles will be decided by the three people to which category they belong.

In the next section, we propose a web behaviour classification method of a collective user by combining the LDA algorithm and a multi-class classifier, as shown in Figure 3. The web page name extracted by Algorithm 1 is a piece of short text, which is used as the input of the encoding layer, as mentioned above. We perform feature engineering to construct the document-word frequency feature matrix based on word segmentation.

3.2 Text pre-processing

Here, we first perform word segmentation by using the API of jieba [38] on the title_name column from the matrix title_infor, which is the output of Algorithm 1, and then the stop words in the web page names are removed to exclude interference items for topic analysis, and finally, a document-word frequency matrix, the result of text pre-processing, is generated as shown in Figure 4.

As demonstrated in Figure 4, the row $d_i$ is the $i$th document, and the column $w_j$ is the $j$th word in the corpus, and $d_{w_j}$ is the frequency that $w_j$ appears in $d_i$.

3.3 Encoding with the LDA algorithm

In this section, to suit our purpose, we build an algorithm to implement the LDA framework.

The LDA algorithm is a generative probabilistic model of a corpus, a set composed of all user browsing records. Each topic in the corpus is characterized by a probabilistic distribution over web page names. The web page name can be considered as a document, and each document is associated with one of those six topics. The LDA algorithm can analyse these web page names to find not only the hidden topics in the
corpus but also the hidden topics of the users. A detailed procedure of our algorithm is shown in Algorithm 2.

Algorithm Hidden topics calculation

- **Input:** the pcap file
- **Optimization:**
  1. for all topics $k \in [1, K]$ do
  2. { draw topic distribution $\phi \sim \text{Dir}(\beta)$; 
  3. draw topic distribution $\theta \sim \text{Dir}(\alpha)$; 
  4. }
  5. for a web page name, $m \in [1, M]$ do
  6. { draw the topic distribution of web page name $\theta_m \sim \text{Dir}(\alpha)$; 
  7. draw the length of web page names $N_m$; 
  8. for all words $n \in [1, N_m]$ in web page names $m$ do
  9. { choose the topic assignment $Z_{m,n} \sim \text{Multinomial}(\theta)$; 
  10. choose the term for word $W_{m,n} \sim \text{Multinomial}(\phi_{Z_{m,n}})$; 
  11. } 
  12. }
  13. }
  14. }

The LDA algorithm is composed of three hierarchies, as shown in Figure 3. The LDA algorithm assumes the following generative process to create a web page name in user browsing records $d$ using $K$ topics.

Our algorithm has implemented all the elements and processing steps in Figure 3, in which $\alpha$ is the per-document topic distribution, $\beta$ is the per-topic word distribution, $\theta_{m}$ is the topic distribution for document $m$, $\phi_{Z_{m,n}}$ is the word distribution for topic $k$, $Z_{m,n}$ is the topic for the $n$th word in document $m$ and $W_{m,n}$ is the specific word. The $W_{m,n}$ represents observable variables, and $Z_{m,n}$ and $\theta_{m}$ refer to latent variables. The outer plate, the document level, refers to a document, while the inner plate, the word level, refers to the process of iteratively choosing a latent topic and a word in the document in order to traverse all the topics and words. According to the graphical model, a document's generation probability can be obtained by Equation (1).

\[
p(Z_m, W_m, \theta, \phi | \alpha, \beta) = \prod_n p(w_{m,n} | \phi_{Z_{m,n}}) p(Z_{m,n} | \theta_m) p(\theta_m | \alpha) p(\phi | \beta)
\]

In Equation (1), variable $\theta_{m}$ and variable $\phi_{Z_{m,n}}$ are term distributions in each topic and the topic distribution in each document, respectively. $Z_m$ is all topics in document, $W_m$ is all specific words in document $m$, $n$ is the number of all words in the range $[1, N_m]$. $N_m$ is the length of the webpage name and obeying the Poisson distribution with parameter $\xi$. $\Phi$ is the parameter of word distribution. Both variables $\theta_m$ and $\phi_{Z_{m,n}}$ are inferred by Gibbs sampling methods [39], which are capable of extracting the topics from a large-scale text. Furthermore, parameter estimation with the Gibbs sampling method is regarded as the inverse process of the text generation process, that is, in the case of a known text set (the generated result), the parameter values are obtained by parameter estimation. Furthermore, to estimate actual parameters, the topic of each word is sampled instead of integration. Once the topic of each word is determined, the parameters are calculated after frequency statistics. Therefore, the parameter estimation problem is transformed into calculating the conditional probability of the subject sequence under the word sequence via Equation (2).

\[
p(Z_i = k | z_{-i}, \omega) = \frac{p(\omega, z_i) \sum_{k=1}^K n_{k,z_i} + \beta_k}{\sum_{k=1}^K n_{k,z_i} + \beta_k} (n_{i,z_i} + \alpha_k)
\]

In Equation (2), variable $z_i$ corresponds to the topic variable of ith word, $\emptyset i$ indicates that the ith item is not included and $n_{k,z_i}$ indicates the number of times that the term $t$ appears in the topic $k$. $\beta_k$ is Dirichlet prior distribution of the word term $t$, and $n_{k}^{m}$, $n_{m}^{k}$ means the number of topic $k$ appears in document $m$ and $z_{i}$ is the Dirichlet prior distribution of top $k$. The necessary parameters are calculated by Equations (3) and (4) once the topic index is obtained.

\[
k_{t} = \frac{n_{k}^{m} + \beta}{\sum_{m=1}^{M} n_{k}^{m} + \beta}
\]

\[
\theta_{m,k} = \frac{n_{m}^{k} + \alpha}{\sum_{k=1}^{K} n_{m}^{k} + \alpha}
\]

In Equations (3) and (4), the value of $\theta_{m,k}$ is the probability of word term $t$ in topic $k$, while the value of $\phi_{Z_{m,n}}$ indicates the probability of word term topic $k$ in document $m$.

As an unsupervised classification algorithm, the LDA algorithm can only divide the web page names into set categories according to the theme but cannot calculate what the topic is. In the next section, supervised methods are introduced into the model to learn the topics based on the calculated result of the LDA algorithm.

### 3.4 Topic classification based on multi-class classifiers

In this paper, we view the output of the LDA algorithm as an eigenmatrix, which is manually labelled as a dataset with six categories of user behaviour topics, that is, video, music, social...
network/blog, news, Internet information retrieval, and e-commerce. Supervised multi-class classifiers, for example, Random Forest, XGBoost, and Naïve Bayesian, are employed to train and test the classification model in our evaluation experiments, respectively. Naïve Bayes classifiers are particularly popular in text classification and are effective solutions for problems such as spam detection. While XGBoost is a decision-tree-based ensemble machine learning algorithm that uses a gradient boosting framework to solve data science problems in a fast and accurate way. In XGBoost, trees are generated in such a way that each new tree corrects errors made by a previously trained decision tree. Compared with XGBoost, Random Forest is also a tree-based ensemble machine learning algorithm that constructs multiple decision trees and merges them to acquire a more accurate and stable result by training each tree independently. The three algorithms are widely used in classification problems, so we use these three methods as representatives to test our model. The trained model will be used to calculate the topics of web behaviour of the collective user. The flow diagram of the model is shown in Figure 5.

As depicted in Figure 5, firstly, web page names are extracted from the title fields of the filtered HTTP flows in a pcap file. Then, text processing is conducted to generate a document-word frequency matrix from the web page names. Next, we input the document-word frequency matrix into the LDA algorithm to calculate the probability distributions over the possible topics. Finally, the calculated results are labelled and then divided into two datasets, a training dataset and a test dataset, for training and testing the multi-class classifiers, respectively. The trained model is used to classify the topics of collective web behaviour from pcap files.

4 | EXPERIMENTAL RESULTS AND DISCUSSION

The primary purpose of this section is to test and evaluate the classification accuracy of the CWBCU models by using the dataset from Section 3. We also test the convergence of the model with different iterative processes.

4.1 | Dataset and experiment setup

As shown in Figure 2, the network traffic is captured by bypassing network traffic in a managed switch and is saved in the form of pcap files. We have collected the network traffic from one managed switch for three months with 56 users at most in our laboratory. The age of the laboratory members is between 20 and 40 years old, of whom more than 80% are students between the ages of 20 and 30. The HTTP flows are extracted from all the pcap files. We have conducted experiments on a balanced dataset with 30,000 non-repeating samples, which are the output of the LDA algorithm and then marked by six categories of topics. Five persons in our laboratory work as assessors who independently performed data labelling. The disagreed labels will be relabelled by another two persons, and the final result is decided by voting of the seven assessors. We validate the label results for each type by random sampling and fix incorrectly labelled data and confirm similar labelling. The raw agreement rate in the first round reaches 99.5% and eventually reaches 100% for all labelled data. To further improve the performance of the LDA model in the sparse and noisy short texts, we aggregate the web page names with the user(source IP in the title_infor), web site (destination_IP in the title_infor) and time interval(20 min) for the training set [40, 41].

4.2 | Experiment evaluation

We first verify the convergence of the model and then compare the classification accuracy among three concrete methods under the CWBCU model.

4.2.1 | The convergence evaluation of the CWBCU model

In order to evaluate the convergence of the model, we write a testing program, in which the perplexity [42] and loglikelihood [43] are used. The perplexity, defined as the geometric average value of the inverse marginal probability of each word in the held-out set of documents, is used to evaluate the LDA algorithm on held-out data. The perplexity is widely used in indirect measuring of predictive performance, as shown in Equation (5).

Because \( \log p(n_{\text{test}} | \alpha, \beta) \) cannot be computed directly, the lower bound on perplexity shown in Equation (6) is used to approximate it.

**Figure 5** The HTTP traffic topic classification model
Figure 6 depicts the testing result of log perplexity and log likelihood as iterated functions for the LDA algorithm on the dataset, respectively. The results indicate the log perplexity and log likelihood are convergent after 150 iterations when \( K = 23 \) and \( K \) is the number of topics.

\[
\text{perplexity}(\eta^{\text{test}}, \lambda, \alpha) \triangleq e^{-\sum_{i} \log p(n_{i}^{\text{test}} | \alpha, \beta)} / \sum_{i}^{n_{\text{test}}}
\quad (5)
\]

\[
\text{perplexity}(\eta^{\text{test}}, \lambda, \alpha) \leq e^{-(\sum E_{i} \log p(n_{i}^{\text{test}}, \theta_{i} | Z_{i} | a, \beta) - E_{i} \log(\theta_{i} | Z_{i}))(\sum_{i}^{n_{\text{test}}})}
\quad (6)
\]

4.2.2 | The classification accuracy of the CWBCU model

In this section, the classification accuracy of LDA-Random Forest, LDA-XGBoost and LDA-Naïve Bayesian are used to evaluate the classification accuracy of the CWBCU model with the same dataset, respectively. Ten-fold cross validation is performed to evaluate the CWBCU model. Classification accuracy is defined as the percentage of correct predictions, as shown in Equation (7).

\[
\text{accuracy}(y, \hat{y}) = \frac{1}{n_{\text{samples}}} \sum_{i=1}^{n_{\text{samples}}} I(y_i, \hat{y}_i)
\quad (7)
\]

In Equation (7), \( \hat{y}_i \) is the predicted values of the \( i \)th sample and is the corresponding actual value.

(1) The classification accuracy of the CWBCU model with LDA-Random Forest method

We first combine the LDA algorithm with a Random Forest classifier to test the classification accuracy of the model. Specifically, the relationship between classification accuracy and the number of decision trees is evaluated by a wide range of experiments, as shown in Figure 7. The range of the number of decision trees is from 5 to 70 and the accuracy is the average value of the ten rounds of experiments in each group by running a grid searching of hyperparameters with a combination of maxDepth = 5, maxBins = 32 and minInfoGain = 0.0.

As shown in Figure 7, the classification accuracy with \( K = 23 \) is superior to \( K = 20 \) and \( K = 30 \), whose classification accuracy ranged from 72.5% to 81.2%. Table 2 shows the maximum accuracy with different values of \( K \) and numtrees.

(2) The classification accuracy of the CWBCU model with LDA-XGBoost method

We then employ LDA-XGBoost method to evaluate the classification accuracy of the CWBCU model with different

![Figure 6](image6.png)

**Figure 6** (a) The convergence plot of log perplexity on \( K = 23 \).
(b) The convergence plot of log likelihood on \( K = 23 \)

![Figure 7](image7.png)

**Figure 7** The relationship between the number of trees and the accuracy with different \( K \) values used in the LDA algorithm.
combinations of hyperparameters. The experimental results are shown in Figure 8.

As illustrated in Figure 8, with different hyperparameter combinations, the classification accuracy is different. The highest classification accuracy is 79.2% when $K = 14$, max_depth = 10, and learning rate = 0.15. In addition, the lowest accuracy is 71.1% when $K = 33$, max_depth = 15, and learning rate = 0.25.

(3) The classification accuracy of the CWBCU model with the LDA-Naïve Bayesian method

We finally evaluate the model by combing the LDA algorithm with Naïve Bayesian classifier under different $K$ values. The experiment result is shown in Figure 9.

In Figure 9, the classification accuracy of the LDA-Naïve method is above 70.6%, and the accuracy is stable under the combination of different $K$ values ($K = 10, 14, 20, 23, 30$ and $33$) and smoothing values (from 1 to 19).

Furthermore, we use a one-way analysis of variance (ANOVA) with a significance level of 0.05 to evaluate the difference among the three classifiers under $K = 10, 14, 23, 30$ and $33$. The calculation results show that $F$-

| $K$  | 5 | 10 | 12 | 14 | 18 | 20 | 23 | 30 | 33 |
|------|---|----|----|----|----|----|----|----|----|
| Numtrees | 23 | 14 | 59 | 9  | 22 | 18 | 19 | 29 | 8  |
| Maximum accuracy | 0.743 | 0.786 | 0.8 | 0.768 | 0.729 | 0.786 | 0.812 | 0.757 | 0.768 |

**TABLE 2** The maximum accuracy with different values of $K$ and numtrees

**FIGURE 8** The accuracy of LDA-XGBoost with different depths and learning rates (a) $K = 10$, (b) $K = 14$, (c) $K = 23$, (d) $K = 33$
value = 31.4265, p-value = 4.32 × 10^{-6} and F critical value = 3.68232. We can conclude that the difference among them is significant but not a random effect.

Under the best classification result with the LDA-Random Forest, the classification accuracy of each type is summarized in Table 3.

From Table 3, we can see that the classification accuracy is relatively balanced, among which classification accuracy of the news is the lowest due to the keywords of the news are not sensitive enough that might be classified as other types. Through calculation, the classification accuracy of the video is the highest. Furthermore, we have noticed that most web page names with word ‘video’ are really classified into the type of video. From Table 2, it should be noticed that the number of web sites for each category does not impact the classification accuracy directly. The unstructured nature of the web page names extracted from pcap files makes it challenging to classify the topics. Moreover, the same words emerging in different web page names lower the discrimination of the web behaviour categories. Considering the context of the web page names with the LDA model, the CWBCU model can acquire better classification results.

### 4.2.3 Comparison between LSA and pLSA

We also compare LDA with LSA and pLSA with the same dataset using different K values (K = 5, 10, 15, 20, 23, 30, and 35) and the number of decision trees of Random Forest ranges from 5 to 70. The relationship between classification accuracy and the number of decision trees is evaluated by a wide range of experiments, as shown in Figures 10 and 11.

As illustrated in Figure 9, the accuracy of the classification ranges from 72% to 78% for the LSA, and the highest accuracy is 77.9% under K = 30 and numtrees = 53. In Figure 10, the accuracy for classification for pLSA ranges from 62% to 79% and the best values is 78.5% under K = 30 and numtrees = 67. According to the analysis results of LSA and pLSA, the accuracy obtained by using the Random Forest algorithm is improved with the increase of the value of K but becomes stable after reaching a specific value. The classification accuracy of pLSA with K = 20 is lower than others obviously.

### 4.2.4 Discussion

We compare the classification accuracy of the LDA-Random Forest method, the LDA-XGBoost method and the LDA-Naïve Bayesian method, and the result shows the highest classification accuracy of each method is 81.2%, 79.2% and 70.6%, respectively. The accuracy of the LDA-Naïve Bayesian method is the lowest. This might be due to the less strong independence between the features; for example, the same words are probably divided into different types in different contexts. Random Forest is a bagging algorithm that reduces variance and is not sensitive to outliers. However, XGBoost is a supervised classifier that implements parallel boosting. As shown in Figure 8, the maximum accuracy is 79.2% for the LDA-XGBoost method obtained by grid hyperparameter searching. The LDA-Random Forest method is superior to the LDA-XGBoost method in terms of classification accuracy. Moreover, in terms of running time, the former takes much less time than the latter. LDA-Random Forest takes approximately 10.2 s to perform grid hyperparameter searching, while XGBoost takes 105.3 s. The above-mentioned time doesn’t include time for data importing and LDA analysis (approximately 84 s).

Compared with LSA-Random Forest method and pLSA-Random Forest method, the LDA-Random Forest method that instantiated the CWBCU model yields higher classification accuracy.

![Figure 9](image)

**Figure 9** The accuracy of LDA-Naïve Bayesian with different K values and smoothing values

*Table 3* The classification accuracy for each type under best classification accuracy

| Type name                        | The no. of distinct web sites | Classification accuracy |
|----------------------------------|------------------------------|-------------------------|
| Video                            | 382                          | 85.63%                  |
| Music                            | 390                          | 82.79%                  |
| Social network/blog              | 387                          | 79.07%                  |
| News                             | 380                          | 64.17%                  |
| Internet information retrieval   | 412                          | 84.79%                  |
| e-commerce                       | 406                          | 71.01%                  |
accuracy by aggregating the users and websites. In our experiments, the best method is LDA-Random Forest, and next LDA-XGBoost, and then pLSA-Random Forest, then LSA-Random Forest, and finally LDA-Naïve Bayesian.

5 | CONCLUSION AND FUTURE WORK

In this study, we use recent developments in topic model technologies and techniques to study user behaviour calculation methods and propose a novel model to classify collective web behaviour from network traffic contents. We have performed extensive experiments to compare the classification accuracies among the LDA-Random Forest method, LDA-XGBoost method and LDA-Naïve Bayesian method, respectively. The highest classification accuracy reaches 81.2% with the LDA-Random Forest method, which demonstrates that the model is useful in classifying the topics of collective web behaviour. Overall, by analysing network traffic contents, the CWBCU model can obtain the topics of web behaviour of a collective user. The network traffic is easily collected from network devices. In addition, compared with the analysis of the whole web page, the calculation of the web page name is quicker to get relatively accurate results.

In order to obtain the distribution of the behaviour of collective users in a community on the six categories, the model we build is based on analysing the web topics in the traffic generated by the collective users in a community and identifying which of the six categories each topic belongs to. Therefore, the results are able to reflect the relationship between collective user behaviour and multiple behavioural topics.

Since our research is the first in the area of web behaviour calculation of a collective user from network traffic contents, it may have some limitations. One of them is that the parameters in the LDA algorithm and the multi-class classifier may be coupled, which should be considered carefully in future research. In future work, we wish to apply deep learning algorithms for improving the classification accuracy to understand a collective user’s web behaviour better.

ACKNOWLEDGEMENTS

This research was supported by the National Key Research and Development Plan under Grant 2016YFB0800802, Key Research and Development Plan of Shandong Province under grant 2017CXGC0706, Peng Cheng Laboratory Project of Guangdong Province PCL2018KP004, frontier science and technology innovation of China under grant 2016QY05X10022, national regional innovation centre scientific and technological special projects grant 2017QYCX14, University Co-construction Project in Weihai City.

REFERENCES

1. Goldstone, R.L., Janssen, M.A.: Computational models of collective behavior. Trends Cognit. Sci. 9(9), 424–430 (2005)
2. Yang, Y.C.: Web user behavioral profiling for user identification. Decis. Support Syst. 49(3), 261–271 (2010)
3. Binxing, F., et al.: Cyber ranges: state-of-the-art and research challenges. J. Cyber Secur. 1(3), 1–8 (2016)
4. Juneja, S., Dixit, A.: Improving search results based on users’ browsing behavior using Apriori algorithm. In: Hoda, M., Chauhan, N., Quadri, S., Srivastava, P. (eds.), Software Engineering: Advances in Intelligent Systems and Computing, pp. 73–82. Springer (2019)
5. Karami, G., Tian, J.: Maintaining accurate web usage models using updates from activity diagrams Inform. Softw. Technol. 96, 68–77 (2018)
6. Xie, X., Wang, B.: Web page recommendation via twofold clustering: considering user behavior and topic relation. Neural Comput. Appl. 29(1), 235–243 (2018)
7. Dumais, S., et al.: Understanding user behavior through log data and analysis. In: Ways of Knowing in HCI, pp. 349–372. Springer (2014)
8. Ferguson, B., Tall, A., Olsen, D.: National cyber range overview. In: IEEE Military Communications Conference, Baltimore, MD, USA, pp. 123–128. (Oct. 2014)
9. http://www.caic.gov.cn/web_pdf/CNNIC42.pdf.
10. Loyola, P., Román, P.E., Velásquez, J.D.: Clustering-based learning approach for ant colony optimization model to simulate web user behavior Proceedings of the 2011 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology, Lyon, France, vol. 01, pp. 457–464 (2011)
11. Awad, M.A., Khalil, I.: Prediction of user's web-browsing behavior: application of Markov model. IEEE Trans. Syst. Man Cybern. Part B Cybern., vol. 42(4), pp. 1131–1142 (2012)

12. Joachims, T., et al.: Accurately interpreting click-through data as implicit feedback. In ACM SIGIR Forum. 51(1), 4–11 (2017)

13. Al-Garadi Ali, M., et al.: Analysis of online social network connections for identification of influential users: survey and open research issues. ACM Comput. Surv. 5(1), 16–37 (2018)

14. Papadimitriou, A., Symeonidis, P., Manolopoulos, Y.: Fast and accurate link prediction in social networking systems. J. Syst. Softw. 85(9), 2119–2132 (2012)

15. lidong, B., et al.: Learning a unified embedding space of web search from large-scale query log. Knowl. Base Syst. 150, 38–48 (2018)

16. Janssen, B.J.: Search log analysis: what it is, what’s been done, how to do it. Library Inform. Sci. Res., vol. 28(3), pp. 407–432 (2006)

17. Arapakis, I., Xiao, B., Barla Cambazoglu, B.: Impact of response latency on user behavior in web search. In: Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval, Queensland, Australiapp. 103–104 (2014)

18. White, R.W., Bilenko, M., Cucerzan, S.: Studying the use of popular destinations to enhance web search interaction. In: Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 159–166 (2007)

19. Weinreich, H., et al.: Off the beaten tracks: exploring three aspects of web navigation. In: Proceedings of the 15th International Conference on World Wide Web, Edinburgh, Scotland pp. 133–142 (2006)

20. Gildersleve, P., Vasset, T.: Inspiration, captivation, and misdirection: emergent properties in networks of online navigation. In: International Workshop on Complex Networks, pp. 271–282. Springer, Cham (2018)

21. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent Dirichlet allocation. J. Mach. Learn. Res. 3, 993–1022 (2003)

22. Pal, M.: Random forest classifier for remote sensing classification. Int. J. Remote Sens. 26, pp. 217–222 (2005)

23. Tianqi, C., Guestrin, C.: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, California, USA, pp. 785–794 (2016). XGBoost: a scalable tree boosting system

24. Park, J.Y., et al.: A large-scale study of user image search behavior on the web. In: Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems, pp. 985–994 (2015)

25. Mishra, R., Kumar, P., Bhasker, B.: A web recommendation system considering sequential information Decis. Support Syst. 75, 1–10 (2015)

26. Lopes, P., Roy, B.: Dynamic recommendation system using web usage mining for e-commerce users Procedia Comput. Sci., vol. 45, pp. 60–69 (2015)

27. Rao, R.J., et al.: Assessing learning behavior and cognitive bias from web logs. In: 2018 IEEE Frontiers in Education conference, San Jose, USA pp. 1–5 (2018)

28. Krestel, R., Funkhauser, P., Nejdl, W.: Latent Dirichlet allocation for tag recommendation. In: Proceedings of the Third ACM Conference on Recommender Systems, New York, USA, pp. 61–68 (2009)

29. Chen, C., Ren, J.: Forum latent Dirichlet allocation for user interest discovery Knowl Based Syst. 126, 1–7 (2017)

30. Guo, Y., Barnes, S.J., Jia, Q.: Mining meaning from online ratings and reviews: tourist satisfaction analysis using latent Dirichlet allocation Tour. Manag. 59, 467–483 (2017)

31. Liu, Z., Wang, R., Tang, D.: Extending labeled mobile network traffic data by three levels traffic identification fusion Future Gen. Comput. Syst. 88, 453–466 (2018)

32. Nair, L.M., Sajeev, G.P.: Internet traffic classification by aggregating correlated decision tree classifier. In: Seventh International Conference on Computational Intelligence, Modelling and Simulation, Kaanu, Malaysia (2016)

33. Guy, L., Ritov, Y.: Traffic flow prediction using adaboost algorithm with random forests as a weak learner Proceedings of World Academy of Science, Engineering and Technology, vol. 19, pp. 193–198 (2007)

34. Buczak, A.L., et al.: Detection of tunnels in PCAP data by random forests. In: Proceedings of the 11th Annual Cyber and Information Security Research Conference, New York, USA pp. 16. (2016)

35. Conti, M., et al.: Can't you hear me knocking: identification of user actions on android apps via traffic analysis. In: Proceedings of the 5th ACM Conference on Data and Application Security and Privacy, San Antonio, USA pp. 297–304 (2015)

36. Conti, M., et al.: Analyzing android encrypted network traffic to identify user actions. IEEE Trans. Int. Forensics Secur. 11(1), 114–125 (2015)

37. Saltatormaggio, B., et al.: Eavesdropping on fine-grained user activities within smartphone apps over encrypted network traffic. In: 10th USENIX Workshop on Offensive Technologies, Austin, USA (2016)

38. Chinese Text Segmentation: Built to Be the Best Python Chinese Word Segmentation Module,. https://github.com/fxsjy/jieba, Accessed 6 Jun. 2019

39. Griffiths, L.T., Steyvers, M.: Finding scientific topics. Proc. Natl. Acad. Sci. USA. 101(Suppl 1), 5228–5235 (2004)

40. Hong, L., Davison, B.D.: Empirical study of topic modeling in Twitter. In: Proceedings of the First Workshop on Social Media Analytics, San Diego, USA, pp. 80–88 (July 2010)

41. Tang, J., et al.: Under-standing the limiting factors of topic modeling via posterior contraction analysis Int. Conf. Mach. Learn. 32, 190–198 (2014)

42. Hoffman, M., Bach, F.R., Blei, D.M.: Online learning for latent Dirichlet allocation. Adv. Neural Infor. Proc. Syst. 1, 856–864 (2010)

43. Buntine, W.: Estimating likelihoods for topic models. Asian Conf. Mach. Learn. 5828, 51–64 (2009)

---

**How to cite this article:** Liu H, Wang S, Wei Y, Wang B. A novel classification model of collective user web behaviour based on network traffic contents. *IET Netw*. 2021;10:173–184. [https://doi.org/10.1049/ntw2.12010](https://doi.org/10.1049/ntw2.12010)