Towards a Profiling View for Unsupervised Traffic Classification by Exploring the Statistic Features and Link Patterns

Meng Qin
ICNLAB, School of Electronics and Computer Engineering (SECE), Peking University, Shenzhen, China
mengqin_az@foxmail.com

Kai Lei*
ICNLAB, School of Electronics and Computer Engineering (SECE), Peking University, Shenzhen, China
PCL Research Center of Networks and Communications, Peng Cheng Laboratory, Shenzhen, China
leik@pkusz.edu.cn

Bo Bai
Theory Lab, 2012 Labs, Huawei Technologies, Co. Ltd., Hong Kong, China
baibo8@huawei.com

Gong Zhang
Theory Lab, 2012 Labs, Huawei Technologies, Co. Ltd., Hong Kong, China
nicholas.zhang@huawei.com

ABSTRACT

In this paper, we study the network traffic classification task. Different from existing supervised methods that rely heavily on the labeled statistic features in a long period (e.g., several hours or days), we adopt a novel view of unsupervised profiling to explore the flow features and link patterns in a short time window (e.g., several seconds), dealing with the zero-day traffic problem. Concretely, we formulate the traffic identification task as a graph co-clustering problem with topology and edge attributes, and proposed a novel Hybrid Flow Clustering (HFC) model. The model can potentially achieve high classification performance, since it comprehensively leverages the available information of both features and linkage. Moreover, the two information sources integrated in HFC can also be utilized to generate the profiling for each flow category, helping to reveal the deep knowledge and semantics of network traffic. The effectiveness of the model is verified in the extensive experiments on several real datasets of various scenarios, where HFC achieves impressive results and presents powerful application ability.

CCS CONCEPTS

• **Information systems** → **Traffic analysis**; • **Mathematics of computing** → *Graph algorithms*;

KEYWORDS

Network traffic classification, flow profiling, co-clustering, attributed networks, non-negative matrix factorization

ACM Reference Format:
Meng Qin, Kai Lei, Bo Bai, and Gong Zhang. 2019. Towards a Profiling View for Unsupervised Traffic Classification by Exploring the Statistic Features and Link Patterns. In *NetAI '19: ACM SIGCOMM 2019 Workshop on Network Meets AI $ ML, August 23, 2019, Beijing, China.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3341216.3342213

*Corresponding author.

NetAI '19, August 23, 2019, Beijing, China

ACM ISBN 978-1-4503-6872-8/19/08…$15.00
https://doi.org/10.1145/3341216.3342213

1 INTRODUCTION

Traffic classification, which aims to identify the categories of network traffic with different applications or protocols, is an essential step for network management, effectively supporting the downstream applications such as QoS guarantee, network accounting, intrusion detection, etc [17].

As reviewed in [4, 19], existing traffic classification methods can be divided into 3 categories, including the port-based, payload-based and statistics-based approaches. The application of machine learning techniques to statistics-based methods has become a significant topic in the recent researches, due to its high performance and strong adaption to dynamic ports and encrypted traffic [17].

Typical examples of the machine learning based statistical methods involve the applications of classic machine learning techniques (e.g., SVM, AdaBoost, Naive Bayes, etc.) [14, 15, 17, 20] and advanced deep learning models (e.g., sparse auto-encoder, etc.) [9, 16]. Despite their effectiveness, there remain two primary limitations.

First, most existing statistic-based approaches are based on the supervised learning paradigm, in which all the possible traffic classes are assumed to be known in advance. Moreover, they intend to explore the traffic features in a relatively long period (e.g., several hours or days). However, there exists new traffic generated by unknown applications (i.e., zero-day traffic) in a short time window (e.g., several seconds) without enough prior knowledge in the training data. In this case, most supervised methods may not only have poor classification performance, but also fail to keep up with the dynamic changing of the network. Although some unsupervised methods (e.g., k-means) can partly deal with this problem [19], additional manual inspection is still needed.

Second, existing methods tend to fully explore the statistic features (e.g., average packet size, etc.) regarding the transmission behaviors of network traffic, but intrinsically ignore the available source of link pattern (i.e., the links among the source and destination hosts). In fact, the spatial properties hidden in such link behaviors also carry deep knowledge about network traffic. Particularly, the linkage patterns of some real network traffic present

obvious local spatial properties according to our observation in Section 2, which can potentially boost the classification performance.

Different from existing statistic-based supervised method, we adopt a novel view of unsupervised profiling to tackle the classic traffic identification task and focus on the classification in a relatively short time window (e.g., several seconds). Concretely, we introduce a novel Hybrid Flow Clustering (HFC) model based on the non-negative matrix factorization (NMF) framework to formulate the classification problem as a graph co-clustering process with topology and edge attributes, where both the sources of statistic features and linkage patterns are comprehensively incorporated.

The effectiveness of HFC is two-fold. First, by fully combining the two available information sources, the unsupervised model can achieve a high classification performance that significantly outperforms other unsupervised methods and is also competitive to some supervised approaches. Moreover, HFC has the powerful ability to generate the semantic descriptions (with representative statistic features and linkage structures) simultaneously when the clustering process is finished, which can reveal the deep knowledge of each cluster to support the decision of the downstream applications.

The remainder of this paper is organized as follow. We first introduce the datasets and preliminaries based on the real traffic data in Section 2, and then give the problem definition of flow clustering in Section 3. Moreover, we elaborate the HFC model in Section 4 and further verify its effectiveness via the experiments in Section 5. Section 6 finally concludes this paper.

2 DATASETS AND PRELIMINARIES

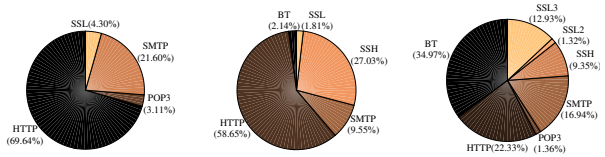
We utilize 3 public datasets of real network traffic provided by [19], which were collected from different Internet positions. *Keio*[1] and *WIDE*[2] are traces maintained by the MAWI working group[2]. The *Keio* trace was captured at a 1Gb/s Ethernet link (lasting for 30 minutes in 2006-08-06) in the Shonan-Fujisawa campus of Keio University, Japan. *WIDE* was taken from a US-Japan trans-Pacific backbone line with 150Mb/s Ethernet link (lasting for 5 hours in 2008-03-18) that carried commodity traffic for the WIDE organizations. *ISP*[2] is a trace dataset captured by [19] via a passive probe with 100-Mb/s Ethernet edge link from an Internet service provider (ISP) in Australia (lasting for 7 days form 2010-11-27).

The ground-truths of the datasets (i.e., the actual classes of the flows) are set by using deep packet inspection (DPI) with high confidence [19]. The class distributions of the datasets are shown in Fig 1, where we ignore the classes with the ratio less than 1% for the convenience of visualization.

For each dataset, we extract 13 typical statistic features, which has been proven effective for the traffic classification task by previous work [14, 15, 19]. Details of these features are elaborated in Table 1, with the last 4 features recorded for both the directions between client and server.

To analysis the link pattern of network traffic, we follow [7] to construct the traffic activity graph (TAG), where we abstract each end host as a node and each flow as an edge between the corresponding nodes. For the convenience of visualization, we randomly

[1]https://drive.google.com/file/d/0B6Qbk6GWTk7NdEJtVHRJQ3ZqSVE/view
[2]http://mawi.wide.ad.jp/mawi/



(a) Keio (b) WIDE (c) ISP

Figure 1: Category distribution of the (a) Keio, (b) WIDE and (c) ISP datasets.

Table 1: Statistics Features of the Datasets

Type	Description	Number
Duration	Duration time of the flow	1
#Pkt.	Number of packets transferred in unidirection	2
Volume	Volume of bytes transferred in unidirection	2
Pkt. Size	Mean and standard deviation of packet size in unidirection	4
Inter-Pkt. Time	Mean and standard deviation of inter-packet time in unidirection	4

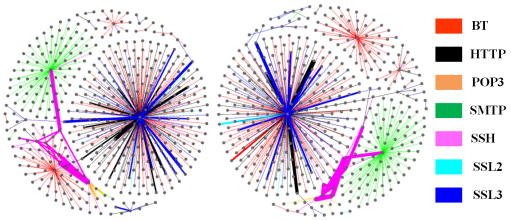


Figure 2: Example TAGs of the ISP dataset.

sampled 500 flows for each dataset by following the class distribution in Fig. 1. To further eliminate the contingency, we repeated the sampling process multiple times, where all the results have similar properties. Two TAGs (generated by different sample processes) on *ISP* are illustrated in Fig. 2, with different edge colors representing different flow classes. It's possible for a certain pair of hosts that generates multiple flows with different classes in a specific time window. We utilized the dominant class to color these edges and found that most of the multiple edges shared by the same host pair have the same class labels.

For both the examples in Fig. 2, there are several obvious link clusters. Especially, the 'BT', 'SMTP' and 'SSH' flows present strong local property that certain host groups tend to generate the traffic in a similar way (i.e., exchanging flows with the same set of hosts). Moreover, other flows (e.g., 'HTTP', 'SSL3', etc.) also forms another mixture linkage cluster. In fact, such link pattern (i.e., spatial distribution) of network traffic can provide additional information to the inference of the flow class, which can be fully utilized to improve the flow identification performance.

In *Keio*, *WIDE* and *ISP*, the numbers of flows reach the magnitude of 1.7×10^5, 2.4×10^6 and 3.4×10^6. According to our statistics, there are respectively 97.43, 133.55 and 5.59 flows per second in the 3 datasets. As we focus on the flow clustering in a relatively short time window, we adopt the proper assumption that there are

Table 2: Details of the Sampled Test Dataset

	Keio(1)	Keio(2)	WIDE(1)	WIDE(2)	ISP(1)	ISP(2)
N	2,000 (supervised) & 1,000 (unsupervised)					
M	50×13	50×13	30×13	30×13	30×13	30×13
L	945	936	800	799	853	866
C	4	4	5	5	7	7

less than 1,000 flows in 5 seconds for the 3 datasets, which satisfies the above statistics.

To simulate the (supervised and unsupervised) traffic inference in a short time period, we set the time window to be 5 seconds and randomly sampled 2,000 flows for the 3 datasets following the class distribution in Fig. 1. For supervised classification methods, we used the first 1,000 flows as the training set (considered as flows in previous time window) with the rest 1,000 flows as the test set (i.e., flows in current time window), while we only utilized the last 1,000 flows for unsupervised clustering methods. We conducted the sampling process twice for each dataset and generated 6 subsets for evaluation. Details of the 6 sampled datasets are presented in Tabel 2, where N, M L and C are the number of flows, features, end hosts (IP addresses) and classes. Especially, $M = P \times M_0$ is the number of features after the discretization (see Section 4.1) with P as the level number and $M_0 = 13$ as the (original) feature number (in Table 1).

3 PROBLEM DEFINITION

Inspired by our observation on TAG in Section 2, we formulate the unsupervised flow classification task as a co-clustering problem of an abstracted graph with edge attributes. Assume that there are L end hosts (i.e., nodes) in a specific time window (e.g., 5 seconds in this study), and the number of flows and features are N and M.

We use the notations $F = \{F_1, \cdots, F_N\}$, $H = \{h_1, \cdots, h_L\}$ and $A = \{a_1, \cdots, a_M\}$ to represent the set of flows, hosts as well as (discrete) attributes. For a certain flow $F_i = \{E_i, A_i\}$, $E_i = \{(h_s, h_d) \,|h_s, h_d \in H\}$ represents an edge in TAG, and $A_i \subseteq A$ is the attribute set of this flow. Given the flow set $F = \{F_1, \cdots, F_N\}$ in a specific time window, the flow clustering problem considered in this study is to partition F into K subsets $\{S_1, \cdots, S_K\}$ with $S_i \subset F$ and $S_i \cap S_j = \emptyset$ $(i \neq j)$, so that:

(i) within each subset, the TAG linkage is dense and the flow features are similar;

(ii) between different subsets, the TAG linkage is relatively loose and the flow features are distinct.

4 THE MODEL

4.1 Modeling the Statistic Features

The original flow statistic features (e.g., duration time, flow volume, etc.) are in the continuous domain (i.e., $a'_i \in \Re$ $(1 \leq i \leq M_0)$). We transform the continuous features into the discrete forms for two reasons. First, the discrete form can reflect the magnitude level of the feature (e.g., high/low level of packet size), which is more suitable for the flow profiling (with clearer semantics) compared with the continuous form. Second, the feature discretization can effectively reduce noise and result in better classification accuracy according to our preliminary experiments (see Appendix).

To realize the discretization, we divide each continuous feature into P discrete levels. Let n_j and m_j be the minimum and maximum values of a certain continuous feature a'_j. The level index of a'_j (notated as l_j) should satisfy $(a'_j - n_j) \in \left[(l_j - 1)s, l_j s\right)$, with $s = (m_j - n_j)/P$ as the span of each level. We use the *flow feature matrix* $\mathbf{F} \in \Re^{N \times M}$ to describe the discrete features, where $\mathbf{F}_{i,(j-1)P+l} = 1$ if the level index of flow F_i's j-th continuous feature is l and $\mathbf{F}_{i,(j-1)P+l} = 0$, otherwise. Moreover, the *feature membership matrix* $\mathbf{X} \in \Re^{N \times K_1}$ and *feature description matrix* $\mathbf{Y} \in \Re^{M \times K_1}$ are introduced to describe the clustering structure (with K_1 clusters). $\mathbf{X}_{ir}$ is defined as the propensity that flow F_i belongs to cluster r, while $\mathbf{Y}_{jr}$ represents the propensity that cluster r can be described by the discrete feature a_j. Consequently, $\sum_{r=1}^{K_1} \mathbf{X}_{ir}\mathbf{Y}_{jr}$ is the expectation that flow F_i has the discrete feature a_j, which should be as close as possible to the real value of $\mathbf{F}_{ij}$. Hence, we have the following NMF-based objective:

$$\underset{\mathbf{X}\geq 0, \mathbf{Y}\geq 0,}{\arg\min} \left\|\mathbf{F} - \mathbf{X}\mathbf{Y}^T\right\|_F^2 + \lambda_1 \sum_{i=1}^{n} \left\|\mathbf{Y}_{i,:}\right\|_2^2 + \lambda_2 \sum_{i=1}^{n} \left\|\mathbf{X}_{i,:}\right\|_1^2, \quad (1)$$

where λ_1 and λ_2 are parameters to control the regularization terms regarding $\mathbf{X}$ and $\mathbf{Y}$. In fact, (1) is equivalent to a typical clustering process [8], with $\mathbf{X}$ and $\mathbf{Y}$ indicating the clustering membership and the cluster centers. The L_1-norm (sparse) regularization on $\mathbf{X}$'s rows and L_2-norm (smoothing) regularization on $\mathbf{Y}$ can make the membership between different clusters more distinguishable while avoiding the problem of overfitting.

For the convenience of derivation, we further transform (1) into the following equivalent form:

$$\underset{\mathbf{X}\geq 0, \mathbf{Y}\geq 0}{\arg\min} \left\|\mathbf{F} - \mathbf{X}\mathbf{Y}^T\right\|_F^2 + \lambda_1 \left\|\mathbf{Y}\right\|_2^2 + \lambda_2 \left\|\mathbf{X}\mathbf{e}\right\|_F^2, \quad (2)$$

where $\mathbf{e}$ is a K_1-dimensional column vector with all elements equal to one. When the solution of (2) (notated as $\{\mathbf{X}^*, \mathbf{Y}^*\}$) is obtained, we use $\mathbf{X}^*$ to extract the clustering membership by assigning the column index with maximum value in the i-th row to be F_i's cluster label, and use $\mathbf{Y}^*$ to generate the feature description by finding the representative features (with relatively large propensities in each column of $\mathbf{Y}^*$) for each cluster.

4.2 Modeling the Link Patterns

We utilize the *link structure matrix* $\mathbf{B} \in \Re^{N \times L}$ to describe the link pattern of network traffic, where $\mathbf{B}_{il} = 1$ if host h_l is an end host of flow F_i and $\mathbf{B}_{il} = 0$, otherwise. Assume that the number of linkage clusters in TAG is K_2. To represent the clustering membership, we introduce the *edge membership matrix* $\mathbf{Z} \in \Re^{N \times K_2}$ and *node membership matrix* $\mathbf{R} \in \Re^{L \times K_2}$, where $\mathbf{Z}_{ik}$ and $\mathbf{R}_{lk}$ are respectively defined as the propensity that edge i (flow F_i) and node l (host h_l) belong to the linkage cluster k. Accordingly, $\sum_{k=1}^{K_2} \mathbf{Z}_{ik}\mathbf{R}_{lk}$ can be considered as the expectation that h_l is an induced node of edge i in TAG, which should be as close as possible to $\mathbf{B}_{il}$. We formulate such relation as the following NMF problem:

$$\underset{\mathbf{Z}\geq 0, \mathbf{R}\geq 0}{\arg\min} \left\|\mathbf{B} - \mathbf{Z}\mathbf{R}^T\right\|_F^2. \quad (3)$$

Similar to (2), one can utilize the solution $\{\mathbf{Z}^*, \mathbf{R}^*\}$ to extract the node and edge members of a certain linkage cluster.

4.3 The Unified Model

We introduce a novel *transition matrix* $\mathbf{U} \in \Re^{K_1 \times K_2}$ to explore the correlation between the clustering structures of statistic features (encoded in $\mathbf{X}$) and TAG linkage (encoded in $\mathbf{Z}$). Concretely, $\mathbf{U}_{rk}$ is defined as the transition propensity from the feature cluster r to the linkage cluster k, indicating the relation that $\mathbf{Z}_{ik} = \sum_{r=1}^{K_1} \mathbf{X}_{ir}\mathbf{U}_{rk}$. Hence, we can construct the unified model of HFC by combining (2) and (3):

$$\arg\min_{\mathbf{X},\mathbf{Y},\mathbf{Z},\mathbf{R}} \left\|\mathbf{F} - \mathbf{X}\mathbf{Y}^T\right\|_F^2 + \alpha \left\|\mathbf{B} - \mathbf{X}\mathbf{U}\mathbf{R}^T\right\|_F^2 + \lambda_1 \left\|\mathbf{Y}\right\|_F^2 + \lambda_2 \left\|\mathbf{X}\mathbf{e}\right\|_F^2, \quad (4)$$

where we utilize $\mathbf{XU}$ to replace $\mathbf{Z}$, and α is used to control the second term regarding TAG linkage.

For the solution of (4) (notated as $\{\mathbf{X}^*, \mathbf{Y}^*, \mathbf{U}^*, \mathbf{R}^*\}$) we utilize $\mathbf{X}^*$ to extract the flow clustering membership, while use $\{\mathbf{Y}^*, \mathbf{U}^*, \mathbf{R}^*\}$ to generate the flow profiling. The profiling includes (i) feature description extracted from $\mathbf{Y}^*$ and (ii) spatial description based on $\{\mathbf{U}^*, \mathbf{R}^*\}$. For a certain *flow cluster* r, we can extract its corresponding *dominant linkage (spatial) clusters* by finding the dominant elements (with relatively large value) in the r-th row of $\mathbf{U}^*$, and use $\mathbf{R}^*$ to get the representative hosts of each *spatial cluster*.

4.4 Model Optimization

We adopt the block coordinate descent method [13] to solve the non-convex NMF problem defined in (4). To obtain the solution, we first properly initialize $\{\mathbf{X}, \mathbf{Y}, \mathbf{U}, \mathbf{R}\}$ and then in terms take the following four steps to continuously update their values until converge.

(i) The X-Step: In this step, we update $\mathbf{X}$ with $\{\mathbf{Y}, \mathbf{U}, \mathbf{R}\}$ fixed. The corresponding updating rule can be derived by solving the following NMF problem only related to $\mathbf{X}$:

$$\arg\min_{\mathbf{X}\geq 0} \mathrm{O}(\mathbf{X}) = \left\|\mathbf{F} - \mathbf{X}\mathbf{Y}^T\right\|_F^2 + \alpha \left\|\mathbf{B} - \mathbf{X}\mathbf{U}\mathbf{R}^T\right\|_F^2 + \lambda_2 \left\|\mathbf{X}\mathbf{e}\right\|_F^2. \quad (5)$$

The gradient descent method can be applied to obtain the updating rule of $\mathbf{X}$ (with the multiplicative form). We leave the derivation details in Appendix and directly list the result as follow:

$$\mathbf{X}_{ir} \leftarrow \mathbf{X}_{ir} \frac{(\mathbf{F}\mathbf{Y} + \alpha\mathbf{B}\mathbf{S})_{ir}}{(\mathbf{X}(\mathbf{Y}^T\mathbf{Y} + \lambda_2\mathbf{E} + \alpha\mathbf{S}^T\mathbf{S}))_{ir}}, \quad (6)$$

where $\mathbf{S} = \mathbf{R}\mathbf{U}^T$ and $\mathbf{E} = \mathbf{e}\mathbf{e}^T$.

(ii) The Y-Step: In the Y-step, we fix $\{\mathbf{X}, \mathbf{U}, \mathbf{R}\}$ and update $\mathbf{Y}$ by using the following updating rule:

$$\mathbf{Y}_{jr} \leftarrow \mathbf{Y}_{jr}(\mathbf{F}^T\mathbf{X})_{jr} \Big/ (\mathbf{Y}(\mathbf{X}^T\mathbf{X} + \lambda_1\mathbf{I}))_{ir}, \quad (7)$$

with $\mathbf{I}$ as a K_1-dimensional identity matrix. The derivation of (7) is similar to that of (6) (see Appendix).

(iii) The U-Step: We update $\mathbf{U}$ with the value of $\{\mathbf{X}, \mathbf{Y}, \mathbf{R}\}$ fixed in the U-step. By using the similar strategy to the derivation of (6), we obtain the following updating rule:

$$\mathbf{U}_{rk} \leftarrow \mathbf{U}_{rk}(\mathbf{X}^T\mathbf{L})_{rk} \Big/ (\mathbf{X}^T\mathbf{X}\mathbf{U}\mathbf{V})_{rk}, \quad (8)$$

where $\mathbf{L} = \mathbf{B}\mathbf{R}$ and $\mathbf{V} = \mathbf{R}^T\mathbf{R}$.

(iv) The R-Step: In this step, we fix the values of {**X**,**Y**,**U**} and update $\mathbf{R}$ with the following updating rule:

$$\mathbf{R}_{lk} \leftarrow \mathbf{R}_{lk}(\mathbf{B}^T\mathbf{Z})_{lk} \Big/ (\mathbf{R}\mathbf{Z}^T\mathbf{Z})_{lk}. \quad (9)$$

Algorithm 1: Hybrid Flow Clustering

Input: $\mathbf{F}$, $\mathbf{B}$, K_1, K_2, $\{\alpha, \lambda_1, \lambda_2\}$
Output: $\mathbf{X}^*$, $\mathbf{Y}^*$, $\mathbf{U}^*$, $\mathbf{R}^*$

```
initialize {X, Y, U, R} by using the NNDSVD strategy
while not converge do
    update X via (6) with {Y, U, R} fixed //The X-Step
    update Y via (7) with {X, U, R} fixed //The Y-Step
    update U via (8) with {X, Y, R} fixed //The U-Step
    update R via (9) with {X, Y, U} fixed //The R-Step
extract the flow clustering membership via X*
generate the feature description of each flow cluster via Y*
explore the correlation between feature and linkage clusters via U*
extract the node/host member of each linkage cluster via R*
```

In this study, we adopt a criterion based on the objective function's value (i.e., (4)'s value) to determine whether the updating process has converged. In each iteration, we record the relative error of (4) with respect to the previous iteration. When the recorded error is less than a pre-set threshold (e.g., 10^{-5} in our experiments), we determine the process has converged. Moreover, we use the NNDSVD strategy [2] to initialize $\{\mathbf{X}, \mathbf{Y}, \mathbf{U}, \mathbf{R}\}$, with the details introduced in Appendix.

In summary, we conclude the overall process of HFC in Algorithm 1. Assume that $\max\{K_1, K_2\} \ll \min\{N, M, L\}$. The time complexity of HFC is within $\mathrm{O}(NMK_1T + NLKT + NPK_2T)$ for T iterations, with $K = \min\{K_1, K_2\}$. The complexity can be further reduced by considering the sparsity of $\mathbf{F}$ and $\mathbf{B}$.

5 EXPERIMENTAL EVALUATION

5.1 Performance Evaluation

We first evaluate the effectiveness of HFC (i.e., whether the clustering membership learned by HFC has a satisfying correspondence to the manual inspected categories of network traffic) by comparing it with 10 baselines with 2 types. *Naive Bayes Classifier*[3] (NBC), *Support Vector Machine*[4] (SVM), *Logistic Regression*[4] (LR), *AdaBoost*[5] (AB) and *K-Nearest Neighbor*[6] (KNN) are mature supervised classification methods, while *K-means*[7] (KM), *Spectral Clustering*[8] (SC) and *Hierarchical Clustering*[9] (HC) are typical unsupervised clustering approaches. All of the above methods have been proven to be effective for the task of flow statistics-based traffic classification [1, 6, 18, 19], so we used them as the baselines that consider the flow statistic features alone. Moreover, we also utilized the statistics-based model defined in (2) (notated as $\mathrm{NMF}_{(S)}$) and the linkage-based model defined in (3) (notated as $\mathrm{NMF}_{(L)}$) as other two baselines to verify the performance improvement given by the integration of the two sources. In this case, $\mathrm{NMF}_{(S)}$, $\mathrm{NMF}_{(L)}$ and HFC are all unsupervised clustering methods to be evaluated.

Note that the supervised classification and unsupervised clustering are two different learning paradigms. To make all the methods comparable, we adopted the Accuracy (AC) [12] and Normalized Mutual Information (NMI) [11] as the evaluation metrics. Due to

[3] https://www.mathworks.com/help/stats/classificationnaivebayes-class.html
[4] https://www.csie.ntu.edu.tw/~cjlin/liblinear/
[5] https://www.mathworks.com/help/stats/fitensemble.html
[6] https://www.mathworks.com/help/stats/classificationknn.fit.html
[7] https://www.mathworks.com/help/stats/kmeans.html
[8] https://sites.google.com/site/speclust/
[9] https://www.mathworks.com/help/stats/linkage.html

Table 3: Evaluation Result in Terms of AC

	Keio(1)	Keio(2)	WIDE(1)	WIDE(2)	ISP(1)	ISP(2)
NBC	0.2750	0.2560	0.4350	0.5150	0.3180	0.3830
SVM	0.6550	0.6690	0.8840	0.8820	0.7130	0.7140
LR	0.6240	0.6470	0.8560	0.8510	0.5810	0.5770
AB	0.8770	0.8620	0.8260	0.8190	0.4520	0.4780
KNN	0.8900	0.8680	0.9200	0.9300	0.8310	0.8300
KM	0.5710	0.5690	0.4490	0.4360	0.4070	0.4090
SC	0.5620	0.6010	0.6310	0.4940	0.5140	0.4420
HC	0.5790	0.5480	0.4510	0.4440	0.4280	0.4180
NMF(S)	0.7920	0.7900	0.7680	0.7420	0.5990	0.5790
NMF(L)	0.5410	0.4110	0.5720	0.5750	0.5160	0.4990
HFC	0.8870	0.8770	0.8540	0.8570	0.6830	0.6870

Table 4: Evaluation Result in Terms of NMI

	Keio(1)	Keio(2)	WIDE(1)	WIDE(2)	ISP(1)	ISP(2)
NBC	0.3612	0.4591	0.5186	0.5173	0.3503	0.3605
SVM	0.2001	0.1993	0.6416	0.6279	0.5160	0.4763
LR	0.1851	0.1838	0.5712	0.5637	0.3450	0.3156
AB	0.5481	0.4960	0.5938	0.5895	0.2945	0.2019
KNN	0.5533	0.4983	0.7378	0.7607	0.6428	0.6423
KM	0.1763	0.1817	0.2220	0.2102	0.2488	0.2299
SC	0.3210	0.3609	0.4238	0.4347	0.4139	0.3804
HC	0.1803	0.1741	0.2280	0.2321	0.2753	0.2629
NMF(S)	0.2791	0.2880	0.5217	0.5690	0.5019	0.4601
NMF(L)	0.1956	0.1829	0.5174	0.5181	0.5047	0.4740
HFC	0.6090	0.5760	0.6934	0.7068	0.5361	0.5477

the space limit, we leave the concrete definition of the two metrics in Appendix. To evaluate a supervised baseline, we first used the training set of a dataset to train the model and then utilized the test set to get the evaluation result. With regard to the unsupervised methods, we set the number of clusters according to the ground-truth, and directly use the test set to obtain the clustering membership. To make HFC comparable with other baselines, we set the number of feature clusters and linkage clusters to be the number of classes given by the datasets (i.e., $K_1 = K_2 = C$).

In the experiment, we fine-tuned the parameters for all the methods (including HFC) with the best metric values reported. The evaluation results on the 6 test datasets (see Table 2) in terms of AC and NMI are illustrated in Table 3 and Table 4, respectively.

According to Table 3 and 4, HFC significantly outperforms the unsupervised competitors on all the datasets for both AC and NMI (with average improvement of 13.73% and 38.04% compared to the second-best baseline). Although it's challenging for an unsupervised clustering method to outperform the supervised classification approaches (with prior knowledge given by the training set), HFC is still competitive to the supervised methods on all the datasets.

In the evaluation, we adjust the parameters of HFC by setting $\alpha \in \{1, 2, \cdots 10\}$, $\lambda_1 \in \{0.01, 0.1, 0.2 \cdots, 1\}$ and $\lambda_2 \in \{0.1, 1, 2, \cdots, 10\}$. The best parameter settings (with respect to the results in Table 3 and 4) for all the datasets are presented in Table 5. According to Table 5, it's hard to find a fixed parameter setting that can ensure HFC to achieve the best performance. In fact, $\{\alpha, \lambda_1, \lambda_2\}$ represent the introduction of prior knowledge about the flow features and link patterns. Some supervised information (e.g., historical labeled features) can help to reduce the search space of the parameters (in a semi-supervised way). We intend to consider it in our future work.

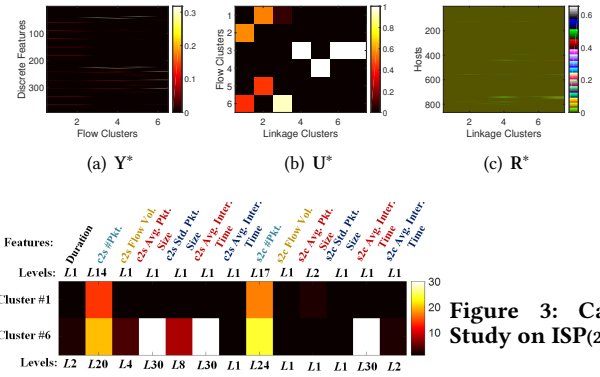
We used MATLAB to implement HFC and tested the running time for all the datasets on a server with Intel Xeon CPU (E5-2680v4 @2.40GHz) and 32GB main memory. The average results are shown in Tabel 6, where the running time on all the test datasets are less than 1 second, indicating that HFC can give the classification result within the specific time window of 5 seconds (see Section 2). The computation can be further speeded up by using some distributed NMF techniques [3] and optimized matrix operation libraries[10].

Table 5: The Best Parameter Settings of HFC

	Keio(1)	Keio(2)	WIDE(1)	WIDE(2)	ISP(1)	ISP(2)
α	9.0	10.0	6.0	9.0	2.0	4.0
λ_1	0.01	0.01	1.0	1.0	0.1	0.1
λ_2	9.0	9.0	10.0	10.0	2.0	0.1

Table 6: The Average Running Time of HFC

	Keio(1)	Keio(2)	WIDE(1)	WIDE(2)	ISP(1)	ISP(2)
HFC	0.3057s	0.3158s	0.7951s	0.7894s	0.6510s	0.6821s



(a) $\mathbf{Y}^*$ (b) $\mathbf{U}^*$ (c) $\mathbf{R}^*$

(d) Example Feature Descriptions

Figure 3: Case Study on ISP(2).

5.2 Case Study

The traffic classification based on the manually inspected class labels (discussed in Section 5.1) may not be the best decision for the downstream traffic applications (e.g., QoS guarantee, etc.), due to the high dynamics of network systems. In fact, we can determine the number of classes and the semantics of each class in an unsupervised manner by fully utilizing the statistic features and link patterns, which is also the basic motivation of HFC.

We use ISP(2) as an example to illustrate HFC's ability to generate flow profiling. First, we utilized the NMF-based model selection strategy introduced in [5] to determine the proper number of feature clusters and linkage clusters (i.e., K_1 and K_2). As a result, we set $K_1 = 6$ and $K_2 = 7$. Then, we used the parameter setting $\alpha = \lambda_1 = \lambda_2 = 1$ to get the solution $\{\mathbf{X}^*, \mathbf{Y}^*, \mathbf{U}^*, \mathbf{R}^*\}$.

Besides the flow clustering membership from $\mathbf{X}^*$, $\{\mathbf{Y}^*, \mathbf{U}^*, \mathbf{R}^*\}$ can be utilized to generate the flow profiling. We visualize $\mathbf{Y}^*$, $\mathbf{U}^*$, $\mathbf{R}^*$ and the example feature description in Fig. 3.

Concretely, we can extract the level index of each statistic feature for a certain cluster r, by finding the row index with maximum (propensity) value in the feature's corresponding range in $\mathbf{Y}^*$'s r-th column (see Fig. 3 (a)). Fig. 3 (d) presents the feature descriptions

[10] http://www.openblas.net

of the flow cluster #1 and #6, where their representative features (i.e., level indexes) are clearly distinguishable.

Moreover, $\mathbf{U}^*$ (see Fig. 3 (b)) encodes the correlation between the *flow clusters* and *linkage clusters*. For instance, #2 and #3 are the dominant *linkage clusters* of the *flow cluster* #1 according to the 1st row of $\mathbf{U}^*$. One can also use $\mathbf{R}^*$ (Fig. 3 (c)) to extract the representative hosts of each linkage cluster by finding the row indexes with large propensities in each column. For instance, #554 and #737 are two representative hosts of *linkage cluster* #2.

6 CONCLUSION

In this paper, we adopted a novel unsupervised profiling view to formulating the classic traffic identification task (in a relatively short time window) as a graph co-clustering problem with topology and edge attributes. Based on such motivation, we proposed a novel HFC model, which effectively leverages the available sources of feature and linkage. In summary, HFC can not only achieve impressive classification performance for manually inspected ground-truths, but also has the powerful ability to generate flow profiling in an unsupervised manner, giving a new insight into the traffic classification problem.

In our future work, we intend to explore a semi-supervised parameter adjustment strategy, which can effectively reduce the search space of HFC's hyper-parameters by utilizing only a small fraction of the supervised information, and is also capable to tackle the zero-day traffic flows.

7 ACKNOWLEDGMENT

This work has been financially supported by the Shenzhen Key Lab for Information Centric Networking & Blockchain Technology (ICNLAB) (ZDSYS201802051831427).

A DERIVATION DETAILS OF HFC

To get the updating rule of a specific variable in $\{\mathbf{X}, \mathbf{Y}, \mathbf{U}, \mathbf{R}\}$, we first extract the terms only related to the variable (e.g., $\mathbf{X}$) in (4) to form an NMF-based optimization problem (e.g., (5)). The corresponding updating rule can be obtained by using the gradient descent method.

Take the solving of (5) as an instance. By utilizing $\|\mathbf{M}\|_F^2 = \mathrm{tr}(\mathbf{M}\mathbf{M}^T)$, we can get $\mathrm{O}(\mathbf{X})$'s partial derivative with respect to $\mathbf{X}$:

$$\frac{\partial \mathrm{O}(\mathbf{X})}{\partial \mathbf{X}} = 2(\mathbf{X}\mathbf{Y}^T\mathbf{Y} - \mathbf{F}\mathbf{Y}) + 2\lambda_2\mathbf{X}\mathbf{E} + 2\alpha(\mathbf{X}\mathbf{S}^T\mathbf{S} - \mathbf{B}\mathbf{S}), \tag{10}$$

where $\mathbf{E} = \mathbf{e}\mathbf{e}^T$ and $\mathbf{S} = \mathbf{R}\mathbf{U}^T$. According to the gradient descent method, we have the following addictive updating rule of $\mathbf{X}$:

$$\mathbf{X}_{ir} \leftarrow \mathbf{X}_{ir} - \eta_{ir}([\cdot]_+ - [\cdot]_-)_{ir}, \tag{11}$$

where $[\cdot]_+ = 2(\mathbf{X}\mathbf{Y}^T\mathbf{Y} + \lambda_2\mathbf{X}\mathbf{E} + \alpha\mathbf{X}\mathbf{S}^T\mathbf{S})$, $[\cdot]_- = 2(\mathbf{F}\mathbf{Y} + \alpha\mathbf{B}\mathbf{S})$ and η_{ir} is the learning rate. Namely, we use the simplified notation $[\cdot]_+$ and $[\cdot]_-$ to represent the terms with positive coefficients and negative coefficients, respectively. By properly setting $\eta_{ir} = \mathbf{X}_{ir}/([\cdot]_+)_{ir}$, we can transform (11) into the following multiplicative form:

$$\mathbf{X}_{ir} \leftarrow \mathbf{X}_{ir}\frac{([\cdot]_-)_{ir}}{([\cdot]_+)_{ir}} = \mathbf{X}_{ir}\frac{(\mathbf{F}\mathbf{Y} + \alpha\mathbf{B}\mathbf{S})_{ir}}{(\mathbf{X}\mathbf{Y}^T\mathbf{Y} + \lambda_2\mathbf{X}\mathbf{E} + \alpha\mathbf{X}\mathbf{S}^T\mathbf{S})_{ir}}, \tag{12}$$

which is equivalent to (6). This solving strategy is effective for most NMF-based problem. [13] has proved that if the variable (e.g., $\mathbf{X}$) is initialized as a non-negative value, the non-negative constraint (e.g., $\mathbf{X} \geq 0$) can be ensured during the updating process.

Moreover, the convergence of the solving strategy is also ensured, since it's based on the gradient descent process. According to our records, by applying Algorithm 1, (4)'s relative error reaches the precision of 10^{-5} within the first 200 iterations for all the datasets, indicating the fast convergence of the above solving strategy.

Table 7: Comparison of Continuous and Discrete Features in Terms of AC

	Keio(1)	Keio(2)	WIDE(1)	WIDE(2)	ISP(1)	ISP(2)
HFC(C)	0.7520	0.7540	0.8080	0.8160	0.5680	0.5770
HFC(D)	**0.8870**	**0.8770**	**0.8540**	**0.8570**	**0.6830**	**0.6870**

B THE INITIALIZATION SETTING

NNDSVD [2] is originally designed for the initialization of standard NMF problem: $\min \left\|\mathbf{M} - \mathbf{W}\mathbf{V}^T\right\|_F^2$. We use the notation that $\{\mathbf{W}^{(0)}, \mathbf{V}^{(0)}\} = \mathrm{N}(\left\|\mathbf{M} - \mathbf{W}\mathbf{V}^T\right\|_F^2)$ to represent the NNDSVD initialization for the variables $\{\mathbf{W}, \mathbf{V}\}$. For HFC, we first initialize $\{\mathbf{X}, \mathbf{Y}\}$ via $\{\mathbf{X}^{(0)}, \mathbf{Y}^{(0)}\} = \mathrm{N}(\left\|\mathbf{F} - \mathbf{X}\mathbf{Y}^T\right\|_F^2)$. Then, we utilize $\{\mathbf{Z}^{(0)}, \mathbf{R}^{(0)}\} = \mathrm{N}(\left\|\mathbf{B} - \mathbf{Z}\mathbf{R}^T\right\|_F^2)$ to initialize $\mathbf{R}$ with $\mathbf{Z}^{(0)}$ as the auxiliary variable. Finally, we use $\{:, \mathbf{U}^{(0)}\} = \mathrm{N}(\left\|\mathbf{Z}^{(0)} - \mathbf{X}'\mathbf{U}^T\right\|_F^2)$ to initialize $\mathbf{U}$. For the concrete algorithm of NNDSVD, please refer to [2].

C DETAILS OF EVALUATION METRICS

We use sequence $R = \{r_1, \cdots, r_N\}$ and $L = \{l_1, \cdots, l_N\}$ to represent the ground-truth (given by the dataset) and result label (given by a certain method) with respect to the flow sequence $F = \{F_1, \cdots, F_N\}$. For the supervised methods to be evaluated, AC is defined as follow:

$$\mathrm{AC}(R, L) = \sum_{i=1}^{N} \delta(r_i, l_i) \Big/ N, \tag{13}$$

where $\delta(r, l) = 1$ if $r = l$ and $\delta(r, l) = 0$, otherwise. For unsupervised methods, we utilize the AC defined as follow:

$$\mathrm{AC}(R, L) = \sum_{i=1}^{N} \delta(r_i, \mathrm{map}(l_i)) \Big/ N, \tag{14}$$

where $\mathrm{map}(\cdot)$ is the permutation mapping function that uses the Kuhn-Munkres algorithm [10] to find the 'best map' between L and R. The NMI metric is defined with the following form for both supervised and unsupervised methods:

$$\mathrm{NMI}(R, L) = 2\mathrm{MI}(R, L)/(\mathrm{H}(R) + \mathrm{H}(L)), \tag{15}$$

where $\mathrm{MI}(R, L) = \sum_{r\in R}\sum_{l\in L}\frac{n_{r,l}}{n}\log\frac{n\times n_{r,l}}{n_r \times n_l}$ is the mutual information between R and L, while $\mathrm{H}(S) = -\sum_{s\in S}\frac{n_s}{n}\log\frac{n_s}{n}$ is the entropy of the label sequence S.

D FEATURE DISCRETIZATION ANALYSIS

Besides the discrete features, we also applied their original continuous forms to HFC, where $\mathbf{F}$ is an $N \times M_0$ matrix with $\mathbf{F}_{ij}$ as the (normalized) real-value of F_i's j-th continuous feature. The fine-tuned best AC metrics with respect to the discrete and continuous features (notated as HFC(C) and HFC(D)) are shown in Table 7, where HFC(D) significantly outperforms HFC(C), indicating the effectiveness of feature discretization discussed in Section 4.1.

REFERENCES

[1] Laurent Bernaille, Renata Teixeira, Ismael Akodkenou, Augustin Soule, and Kave Salamatian. 2006. Traffic classification on the fly. *ACM SIGCOMM Computer Communication Review* 36, 2 (2006), 23.
[2] C. Boutsidis and E. Gallopoulos. 2008. SVD based initialization: A head start for nonnegative matrix factorization. *Pattern Recognition* 41, 4 (2008), 1350–1362.
[3] Tianxiang Gao and Chris Chu. 2018. DID: Distributed Incremental Block Coordinate Descent for Nonnegative Matrix Factorization. In *AAAI Conference on Artificial Intelligence*. 2991–2998.
[4] Bin Hu and Yi Shen. 2012. Machine learning based network traffic classification: a survey. *JOURNAL OF INFORMATION & COMPUTATIONAL SCIENCE* 9, 11 (2012), 3161–3170.
[5] Brunet Jean-Philippe, Tamayo Pablo, Todd R Golub, and Jill P Mesirov. 2004. Metagenes and molecular pattern discovery using matrix factorization. *Proc Natl Acad Sci USA* 101, 12 (2004), 4164–4169.
[6] Yu Jin, Nick Duffield, Patrick Haffner, Subhabrata Sen, and Zhi Li Zhang. 2010. Inferring applications at the network layer using collective traffic statistics. *ACM SIGMETRICS Performance Evaluation Review* 38, 1 (2010), 351–352.
[7] Yu Jin, Esam Sharafuddin, and Zhi Li Zhang. 2009. Unveiling core network-wide communication patterns through application traffic activity graph decomposition. *ACM SIGMETRICS Performance Evaluation Review* 37, 1 (2009), 49–60.
[8] Jingu Kim and Haesun Park. 2008. *Sparse nonnegative matrix factorization for clustering*. Technical Report. Georgia Institute of Technology.
[9] Mohammad Lotfollahi, Ramin Shirali Hossein Zade, Mahdi Jafari Siavoshani, and Mohammdsadegh Saberian. 2017. Deep packet: A novel approach for encrypted traffic classification using deep learning. *arXiv preprint arXiv:1709.02656* (2017).
[10] Laszlo Lovasz and Michael Plummer. 2009. Matching Theory. *Annals of Discrete Mathematics* 367 (2009).
[11] Qin Meng, Jin Di, Lei Kai, Gabrys Bogdan, and Musial Katarzyna. 2018. Adaptive Community Detection Incorporating Topology and Content in Social Networks. *Knowledge-Based Systems* 161 (2018), 342–356.
[12] Cui Peng, Wang Xiao, Pei Jian, and Wenwu Zhu. 2017. A Survey on Network Embedding. *IEEE Transactions on Knowledge & Data Engineering* PP, 99 (2017), 1–1.
[13] Guo Jun Qi, Charu C. Aggarwal, and Thomas Huang. 2012. Community Detection with Edge Content in Social Media Networks. In *IEEE International Conference on Data Engineering*. 534–545.
[14] Muhammad Shafiq, Xiangzhan Yu, Asif Ali Laghari, Lu Yao, Nabin Kumar Karn, and Foudil Abdessamia. 2016. Network Traffic Classification techniques and comparative analysis using Machine Learning algorithms. In *ICCC*. 2451–2455.
[15] Muhammad Shafiq, Xiangzhan Yu, and Dawei Wang. 2017. Network Traffic Classification Using Machine Learning Algorithms. In *Advances in Intelligent Systems and Interactive Applications*. 621–627.
[16] Hongtao Shi, Hongping Li, Zhang Dan, Chaqiu Cheng, and Xuanxuan Cao. 2018. An Efficient Feature Generation Approach based on Deep Learning and Feature Selection Techniques for traffic classification. *Computer Networks* 132 (2018).
[17] Guanglu Sun, Lili Liang, Chen Teng, Xiao Feng, and Lang Fei. 2018. Network traffic classification based on transfer learning. *Computers & Electrical Engineering* (2018), S004579061732829X.
[18] Guanglu Sun, Chen Teng, Yangyang Su, and Chenglong Li. 2018. Internet Traffic Classification Based on Incremental Support Vector Machines. *Mobile Networks & Applications* 14 (2018), 1–8.
[19] Jun Zhang, Xiao Chen, Yang Xiang, Wanlei Zhou, and Jie Wu. 2015. Robust Network Traffic Classification. *IEEE/ACM Transactions on Networking* 23, 4 (2015), 1257–1270.
[20] Jun Zhang, Xiang Yang, Wang Yu, Wanlei Zhou, Xiang Yong, and Guan Yong. 2012. Network Traffic Classification Using Correlation Information. *IEEE Transactions on Parallel & Distributed Systems* 24, 1 (2012), 104–117.