Hybrid Deep Neural Network - Hidden Markov Model Based Network Traffic Classification

Xincheng Tan and Yi Xie(✉)

School of Data and Computer Science, Sun Yat-sen University, Guangzhou, China
tanxch3@mail2.sysu.edu.cn, xieyi5@mail.sysu.edu.cn

Abstract. Traffic classification has been well studied in the past two decades, due to its importance for network management and security defense. However, most of existing work in this area only focuses on protocol identification of network traffic instead of content classification. In this paper, we present a new scheme to distinguish the content type for network traffic. The proposed scheme is based on two simple network-layer features that include relative packet arrival time and packet size. We utilize a new model that combines deep neural network and hidden Markov model to describe the network traffic behavior generated by a given content type. For a given model, deep neural network calculates the posterior probabilities of each hidden state based on given traffic feature sequence; while the hidden Markov model profiles the time-varying dynamic process of the traffic features. We derive the parameter learning algorithm for the proposed model and conduct experiments by using real-world network traffic. Our results show that the proposed approach is able to improve the accuracy of conventional GMM-HMM from 77.66% to 96.11%.

Keywords: Network traffic · Content classification · Hidden Markov model · Deep neural network

1 Introduction

Internet traffic classification is the basis of effective network management. Such as Quality of Services (QoS), network planning and provisioning and network security. Currently, the common approach is to categorize traffic into different types of protocols or applications mainly by three methods: port-based, payload-based, and flow-based. However, with the development of network techniques, the design of network application and protocol is more sophisticated. Which allows one single protocol/application can carry various kinds of communication

Supported by the Natural Science Foundation of Guangdong Province, China (No.2018A03031303), the Fundamental Research Funds for the Central Universities (No.17lgjc26) and the Natural Science Foundation of China(No.U1636118).

X. Liu et al. (Eds.): ChinaCom 2018, LNICST 262, pp. 604–614, 2019.
https://doi.org/10.1007/978-3-030-06161-6_59

contents [6] (e.g. HTTP can supply text, game, video and audio streaming). Just only identifying the protocol/application [1,3,8–10,15] type is not sufficient for effectively monitoring of network flows.On the other hand, traditional port-based and payload-based methods will meet certain limitation when the applications use dynamic port and encrypted transmission.

The goal of this paper is to classify Internet traffic into different content types. The core idea of the program is that the network flows generated by same type of content have similar behavior characteristics evolving over time. Using these characteristics can distinguish the content of network flow. Thus, we propose a new model that combines deep neural network and hidden Markov model to describe the behavior of network traffic generated by a given content type. For a given model, HMM profiles the time-varying dynamic process of the traffic features for specific content type; while DNN calculates the posterior probabilities of each hidden state based on contextual observation values. There are several reasons for using DNN-HMM. First, HMM is a simple and effective model to describe the time series data, which is suitable for modeling the behavior of network flow. However, HMM has two limitations: (i) the independent assumption of observation values; (ii) the ability of processing complex observation values is not efficient enough. The prominent performance of DNN in classification problems can not only solve the context-related problems, but also deal with complex observation values. Therefore, the advantages of DNN can be used to make up for the inherent defects of HMM. We perform experiments on real-world traffic, including six most common content types: audio, game, image, live video, radio, and video of demand (VOD). Results show that our approach has a significant improvement compared to conventional GMM-HMM.

The major contributions of this paper are summarized as follows.: (i) we use a DNN-HMM hybrid structure, the DNN is used to replace conventional probability distribution function (i.e. Gamma, Gauss, Poisson) of HMM state; (ii) we use packet arrival time (instead of inter-arrival time) and packet size as observation values, and multivariate mixed Gaussian distribution is used as the pdf of the observable vector; (iii) we propose a more fine-grained classification method based content type of traffic transferred.

The rest of the paper is organized as follows. In Sect. 2 we review related work about network traffic classification. Section 3 introduces the basic idea of DNN-HMM. Section 4 shows the experiment result of proposed model. Section 5 analyzes several key elements for the model performance improvement. Finally, Sect. 6 draws conclusions and outlines the future work.

2 Related Work

Recently, a number of traffic classification techniques have been studied, which can be categorized into three kinds: port-based, payload-based, and flow-based. However, the port-based approach is rather inaccurate due to dynamic port allocation, and some applications override the well-known port in order to bypassing the firewall [7]. The payload-based [4] approach cannot deal with the encrypted traffic and there is privacy concern with inspecting the payload.

The flow-based method was developed to address these problems. This method classifies traffic based on the flow statistics, e.g. flow duration, flow size and number of packets. Such methods usually use machine learning techniques to build profile patterns for specific traffic type, and do not affected by encryption due to not require to access to the payload. The authors of [9,10] provided surveys of techniques for traffic classification using classic ML. Although the classic ML methods can achieve high accuracy under some conditions, it's difficult to extract proper features for classification and calculate so many features are time consuming. An alternative is to use the raw packet-level information, such as the packet size (PS) and inter-arrival time (IAT).

Wright et al. [15] are the pioneers that applied HMM for traffic classification, For each traffic type, they built two HMMs by utilizing the PS and IAT respectively. Alberto et al. [1,3] proposed an improved model on basis of the previous work. They used a two-dimensional observation values trying to take account of the joint characteristics of PS and IAT, where PS and IAT obey Gamma distribution. However, the authors assumed that the PS and IAT are statistically independent, which contradicts our analysis of real-world traffic.

Recent years, deep learning [5] has been the new trend of ML, which has achieved successful applications in the fields of image and speech recognition. The performance of classic ML techniques heavily rely on handcraft features, therefore some researchers make applications of deep learning for IP traffic classification by using the raw traffic data. In [14], the authors used the consecutive payload bytes of TCP session as the input of DNN to identify protocols. Wang et al. [13] proposed an end-to-end scheme to classify encrypted traffic based on convolution neural network. Chen et al. [2] converted the early sequence of packets into image, and then classified the traffic according to the way of processing image by CNN. However, the approaches mentioned above do not consider the temporal correlation of flow features. Thus in this paper, our proposed method tries to combine the advantages of both HMM and DNN, and classify traffic from another point of view based on content.

3 Proposed Method

3.1 Traffic Pattern Characterization

Figure 1 displays the dynamic model of network flow. We can divide the time-varying dynamic process of the traffic flow into two parts: one is the observable layer, used to describe the changes of external flow shape and characteristics over time, such as the packet size and arrival time, i.e. the observation values; the other one is the state layer, used to describe the internal state changes of network flow generation mechanism or working mode over time. The internal states transition of network flow represents the change of flow pattern over time, and determines the measurements index of external shape and characteristics. However, in a practical application, the internal state of the network flow is difficult to measure directly, it can only be inferred and estimated by the measurement indicators of the external characteristics.

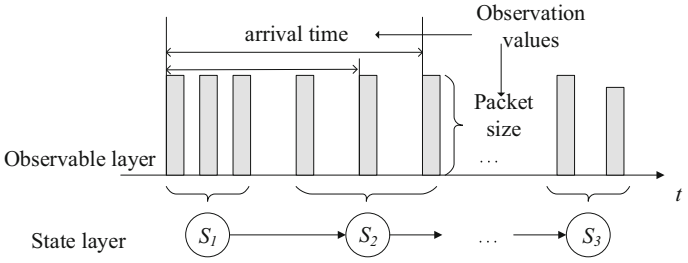

Fig. 1. The dynamic model of network flow

In general, network flows of the same type have relatively fixed change pattern, thus the behavior characteristic for a specific type of network flows can be described by external measurement indicators and internal states together, and used as the basis for identification. More specifically, for a given content type of network flow c, $\overrightarrow{O}_t^c$ denotes measurement vector at time t, Q_t^c is the corresponding internal state. To simplify the complexity of quantitative modeling, we assume that the current state Q_t^c is only related to the previous state Q_{t-1}^c, and independent of $Q_{1:t-2}^c$. In addition, the observation value $\overrightarrow{O}_t^c$ is only related to the current state Q_t^c, and independent of $Q_{1:t-1}^c$, $Q_{t+1:T}^c$, $\overrightarrow{O}_{1:t-1}^c$, $\overrightarrow{O}_{t+1:T}^c$. Thus, this work proposes to use HMM to profile the interaction between external measurement index and internal state of the network flow.

3.2 GMM-HMM

To built a DNN-HMM, firstly we need to train a baseline GMM-HMM. $\boldsymbol{O}_T = (\boldsymbol{o}_1, \boldsymbol{o}_2, ..., \boldsymbol{o}_T)$ is a sequence of observations generated by single HMM, T is the length of sequence. $\boldsymbol{o}_t = (x_t, y_t)'$ is a continuous bi-dimensional observable at time t, $q_t \in \{s_1, s_2, ..., s_Q\}$ is the corresponding state of $\boldsymbol{o}_t$, where Q denotes the number of states for HMM, x_t denotes the packet size (PS) at time t, y_t denotes $10\log_{10}(AT/1\mu s)$. $\lambda = \{\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{\pi}\}$ denotes the set of parameters characterizing the HMM model, the details are given as follows:

1. $\boldsymbol{A} = \{a_{ij}\}$ denotes the state transition probability matrix.

$$a_{ij} = \Pr(q_{t+1} = s_j | q_t = s_j), 1 \leqslant i, j \leqslant Q \tag{1}$$

2. $\boldsymbol{B} = \{b_i(\boldsymbol{o}_t)\}$ denotes the observation probabilities, where $b_i(\boldsymbol{o}_t)$ represents the probability of observation vector $\mathbf{o}_t$ at state s_i.

$$b_i(\boldsymbol{o}_t) = \Pr(\boldsymbol{o}_t | q_t = s_i) \tag{2}$$

3. $\boldsymbol{\pi} = \{\pi_i\}$ denotes the initial state distribution.

$$\pi_i = \Pr(q_1 = s_i), 1 \leqslant i \leqslant Q \tag{3}$$

We calculate the cross-correlation coefficient of packet arrival times and packet sizes within the same session, if the AT and PS is statistically independent, the cross-correlation coefficient will be very close to zero. Figure 2 shows that, in the traffic we analyzed, all content types exhibit significant cross-correlation between AT and PS, which demonstrates the AT and PS are clear related within same session. Thus we use joint distribution as pdf of observable vector, i.e.:

$$\Pr(\boldsymbol{o}_t|s_t = i) = \sum_{m=1}^{M} \frac{c_{i,m}}{2\pi|\boldsymbol{\Sigma}_{i,m}|^{1/2}} \exp[-\frac{1}{2}(\boldsymbol{o}_t - \mu_{i,m})^T \boldsymbol{\Sigma}_{i,m}^{-1}(\boldsymbol{o}_t - \mu_{i,m})] \quad (4)$$

$c_{i,m}$ is the weight of Gaussian mixture, and $\sum_1^M c_{i,m} = 1$ of certain state i. $\mu_{i,m} \in R^2$ is the mean vector of Gaussian, $\boldsymbol{\Sigma}_{i,m}^{-1} \in R^{2\times 2}$ is the covariance matrix.

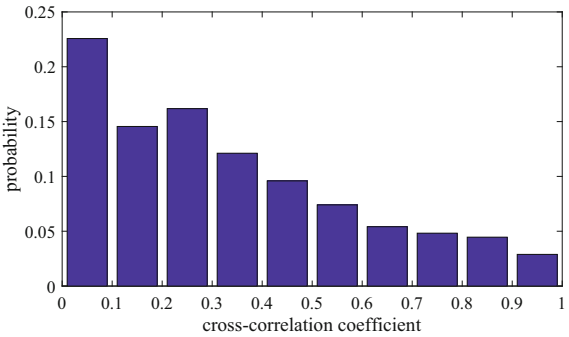

Fig. 2. Cross-correlation coefficient between AT and PS

3.3 DNN-HMM

Once the GMM-HMM is available, we replace the GMM-HMM model's Gaussian mixtures with a DNN and compute the HMM's state emission likelihoods $\Pr(\boldsymbol{o}_t|q_t)$ by converting state posteriors probability $\Pr(q_t|\boldsymbol{o}_t)$ obtained from the DNN. Compared to conventional GMM model, there are two advantage of DNN. Firstly DNN overcomes the complexity of the traditional analysis process and the difficulty of selecting the appropriate model function. Secondly, DNN can use the information of adjacent observation, which partially alleviates the violation to the observation independence assumption made in HMMs [16]. Figure 3 illustrates the architecture of the classifier. The details of the system are shown as follows:

1. The training procedure of DNN-HMM mainly includes three steps:
   (a) For each type of traffic $c \in \{1, 2, ..., C\}$, we train a GMM-HMM with parameters λ_c using the observation sequences of class c, where λ_c is easily estimated by Baum-Welch algorithm [11].

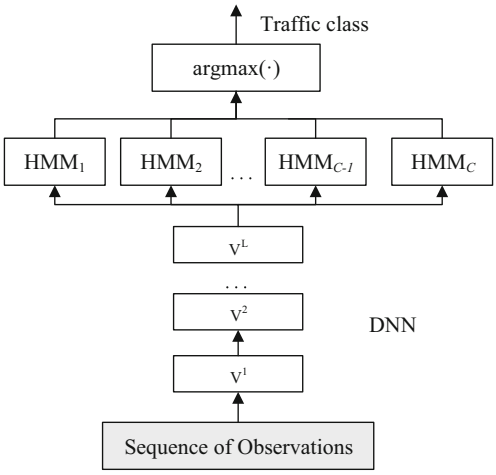

Fig. 3. The architecture of classifier

(b) For each observation sequence $\boldsymbol{O}_T^c$ of class c, using the corresponding model λ_c to obtain the optimal state sequence $\boldsymbol{Q}_T^*$ through the Viterbi algorithm [11]. The states of all HMMs are merged into one set $S = \bigcup S^c$, where S^c is the state space of model c. Then mapping the states space S of the HMMs to the output labels of DNN.

$$\boldsymbol{Q}_T^* = \underset{q_1,q_2,\ldots,q_T \in S^c}{\arg\max} \Pr(q_1, q_2, ..., q_T | \boldsymbol{O}_T^c, \lambda_c) \tag{5}$$

(c) Using the pairs of observable values and corresponding states $(\boldsymbol{o}_t, q_t)$ to train the DNN.Here, we use a regular Feedforward Neural Network (FNN), the more details about DNN can be seen in [14].

2. Classify procedure: For a new traffic flow sequence $\boldsymbol{O}$, firstly it is input into the DNN, and DNN outputs the posterior probability $\Pr(q_t = s|\boldsymbol{o}_t)$. Then we compute its probability $\Pr(\boldsymbol{O}|\lambda_c)$ of being generated by each of the HMMs, since calculating $\Pr(\boldsymbol{O}|\lambda_c)$ requires the likelihood $\Pr(\boldsymbol{o}_t|q_t = s)$ instead of the posterior probability, converted it as follow:

$$\Pr(\boldsymbol{o}_t|q_t = s) = \Pr(q_t = s|\boldsymbol{o}_t) \Pr(\boldsymbol{o}_t) / \Pr(s) \tag{6}$$

Here, $\Pr(s)$ is the prior probability of state s in the training set. $Pr(\boldsymbol{o}_t)$ is independent of the state, thus it can be ignored. Finally, the observation sequence $\boldsymbol{O}$ is assigned to the class that generates it with the highest probability

$$c^* = \underset{1 \le c \le C}{\arg\max} \Pr(\boldsymbol{O}|\lambda_c) \tag{7}$$

Where $\Pr(\boldsymbol{O}|\lambda_c)$ is calculated by using the Forward-backward algorithm [11].

4 Experiment and Results

4.1 Dataset

The traffic we considered includes six most common content types of traffic, namely audio, game, image, live video streaming, radio, and video of demand, shown in Table 1. The audio topology represents the online music, including several audio formats (mp3, m4a and mp4), and the radio typology represents the web radio broadcast, which is a real-time online audio streaming. As regards game traffic, web game is becoming a new trend, because it does not require download and installation, we choose one popular web cards game: Fight the Landlord. We only study dozens of game flows, because the traffic of game is strongly dependent of user behavior and distinctly different form other types of traffic, so it can be easily classified. As for the image traffic, we analyzed the image types from about 100 web pages and two most common image types JPEG and PNG were chosen. As for video streaming, the live represents the live video streaming, and the VOD represents the video that allows the viewer to select content and view it at a time of his own choosing, the more detailed difference between live streaming and VOD can be found in [12].

In this paper, we only model the traffic from server to client, because we want to infer the type of content the user is browsing from the flow patterns. And we do not take account into packets with empty payload, like TCP control packets (SYN, ACK, Keep-Alive). All the sessions consisting of less than 5 packets are omitted from our analysis.

We collected the traffic on my PC by accessing the well-known websites and labeled it manually, the collection process lasted for multiple time periods. The dataset is separated into two portions: a training set and a testing set.

Table 1. The details of dataset

Type of traffic	Training set	Testing set
Audio	148	122
Game	32	30
Image	2697	1148
Live	241	210
Radio	221	101
VOD	227	135

4.2 Parameters of the Model

Each content typology generates a HMM model. For simplicity, we set each HMM has the same number of Q states, and each state has same number of M gauss mixtures. We try to keep the values for Q and M as small as possible in order to obtain lower computational complexity, and at same time provide

sufficient accuracy in modeling all typologies of traffic. We make comprehensive comparison of the model performance for different values of Q and M, and then choose values of $Q = 3$ and $M = 2$ as the model parameters.

After the GMM-HMM training finished, we can obtain the training data for DNN. The dataset is divided into three subsets: 75% as the training set, 15% as the validation set, and 15% as the testing set. The validation set is used for checking the DNN parameters.

4.3 Result

We use Accuracy to evaluate the overall performance of a classifier, and use Precision and Recall to evaluate the performance for each class of traffic.

Table 2. Classification accuracy of GMM-HMM and DNN-HMM

Model	Training set	Testing set
GMM-HMM	89.46%	77.66%
DNN-HMM(4×20)	98.21%	96.11%

The overall classification accuracy of different system is compared in Table 2. The baseline GMM-HMM was trained on the training set, and achieved accuracy of 77.66% on the testing set. A DNN with four hidden layers each of which has 20 neurons was trained on basis of the GMM-HMM model. We can see that the accuracy of DNN-HMM is 8.75% higher than that of the GMM-HMM on the training set. Moreover, when the hybrid model is applied to the testing set, the classification accuracy increases from GMM-HMM's 77.66% to DNN-HMM's 96.11% - a 23% relative improvement, which is a very high gain.

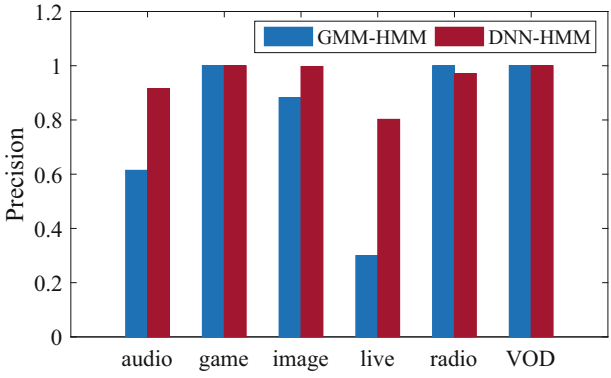

Fig. 4. The precision comparision of GMM-HMM and DNN-HMM

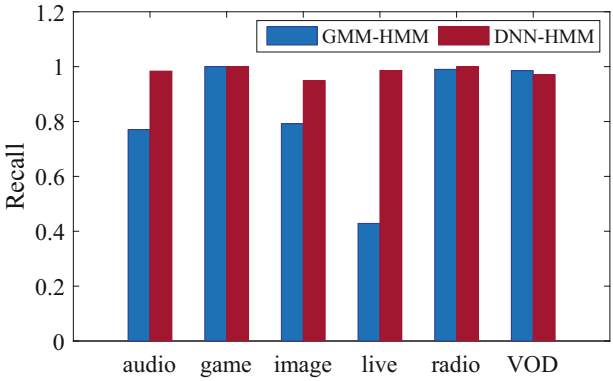

Fig. 5. The recall comparision of GMM-HMM and DNN-HMM

Figures 4 and 5 show the precision and recall comparision of six classes of traffic. As seen in Fig. 4, the precision of audio, image, and live in DNN-HMM is higher than GMM-HMM, and two classes is approximately equal, the radio class has a inferior precision of DNN-HMM. As seen in Fig. 5, the DNN-HMM can achieve more than 90% recall for each traffic type. Three typologies of traffic (audio, image and live) have a significant increase compared to GMM-HMM, and radio only has a slightly increase because it already has a very high recall. Both of the models obtain high recall for game, radio and VOD. In summary, the DNN-HMM achieves better performance than the GMM-HMM on the task of content classification of network traffic.

5 Discussion and Analysis

5.1 The Influence of Hidden Layers

Table 3 lists the results for multiple hidden layers (DNN-HMM) and corresponding single layer (NN-HMM) with the same number of parameters on testing set. One can notice that the classification accuracy increases with more hidden layers. When only one layer is used, the accuracy is 93.07%. When four hidden layers (4×20) are used, the DNN-HMM get the best accuracy of 96.11%, while the corresponding single layer (1×63) model obtains 95.25%. However, the classification accuracy starts to decrease when the number of hidden layers more than five. In summary, compared to the single layer model, a deeper model has better performance. The experimental results demonstrate that DNN with deeper structure has more strong nonlinear modeling power.

5.2 The Size of Contextual Window

Table 4 shows the accuracy comparison for whether using the information of adjacent packets. It can be clearly seen that regardless of whether shallow or deep

networks are used, using information of adjacent packets significantly increases the classification accuracy. The performance of using five consecutive packets information is better than three packets, and three packets better than only one packet. Moreover, a deep network improves performance more than the single hidden layer network when using adjacent packets information, which can achieve high accuracy of 96.11%, while the corresponding network with single layer achieves 95.25%. Note that if only one single packets information is used, the DNN-HMM is even worse than GMM-HMM (77.15% versus 77.66%).

Table 3. Accuracy for multiple hidden layers versus single layer

L×N	DNN-HMM	1×N	NN-HMM
1×20	93.07%		
2×20	95.07%	1×34	94.16%
3×20	95.65%	1×50	94.67%
4×20	96.11%	1×63	95.25%
5×20	94.50%	1×77	94.33%

Table 4. Comparision for different size of window

Model type	1	3	5
NN-HMM (1×63)	76.98%	94.27%	95.25%
DNN-HMM(4×20)	77.15%	94.79%	96.11%

6 Conclusion

In this work, we propose a Deep Neural Network-Hidden Markov Model (DNN-HMM) for accurate traffic classification based on packet-level properties (packet arrival time and packet size). The classification scheme takes advantage of DNN's strong representation learning power and HMM's dynamic modeling ability. Moreover, we classify traffic into different content categories, including streaming audio/video, game, and image. We place our classifier in a real-world trace environment, the experiment results show that the proposed technique can achieve a significant performance improvement compared to the conventional GMM-HMM. Our results suggest that two key factors contributing most to the performance improvement of DNN-HMM are: (1) using deep neural networks with sufficient depth; (2) using the information of multiple consecutive packets. We will expand our method by considering more content typologies of traffic and real-time classification for the future work.

References

1. Alberto, D., Antonio, P., Pierluigi, S., et al.: An hmm approach to internet traffic modeling. In: proceeding of IEEE GLOBECOM (2006)
2. Chen, Z., He, K., Li, J., Geng, Y.: Seq2img: A sequence-to-image based approach towards ip traffic classification using convolutional neural networks. 2017 IEEE International Conference on Big Data (Big Data), pp. 1271–1276. IEEE (2017). https://doi.org/10.1109/bigdata.2017.8258054
3. Dainotti, A., De Donato, W., Pescape, A., Rossi, P.S.: Classification of network traffic via packet-level hidden markov models. In: Global Telecommunications Conference, 2008. IEEE GLOBECOM 2008. IEEE. pp. 1–5. IEEE (2008). https://doi.org/10.1109/glocom.2008.ecp.412
4. Finsterbusch, M., Richter, C., Rocha, E., Muller, J.A., Hanssgen, K.: A survey of payload-based traffic classification approaches. IEEE Commun. Surv. Tutor. **16**(2), 1135–1156 (2014). https://doi.org/10.1109/surv.2013.100613.00161
5. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **521**(7553), 436 (2015). https://doi.org/10.1038/nature14539
6. Li, W., Moore, A.W., Canini, M.: Classifying http traffic in the new age. In: ACM SIGCOMM 2008 Conference (2008)
7. Madhukar, A., Williamson, C.: A longitudinal study of p2p traffic classification. In: IEEE International Symposium on Modeling, Analysis, and Simulation. pp. 179–188 (2006). https://doi.org/10.1109/mascots.2006.6
8. Mu, X., Wu, W.: A parallelized network traffic classification based on hidden markov model. In: 2011 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), pp. 107–112. IEEE (2011)
9. Nguyen, T.T.T., Armitage, G.: A survey of techniques for internet traffic classification using machine learning. IEEE Press (2008)
10. Perera, P., Tian, Y.C., Fidge, C., Kelly, W.: A comparison of supervised machine learning algorithms for classification of communications network traffic. In: International Conference on Neural Information Processing, pp. 445–454. Springer, Berlin (2017). https://doi.org/10.1007/978-3-319-70087-8_47
11. Rabiner, L.R.: A tutorial on hidden markov models and selected applications in speech recognition. Proc. IEEE **77**(2), 257–286 (1989). https://doi.org/10.1109/5.18626
12. Thang, T.C., Le, H.T., Nguyen, H.X., Pham, A.T., Kang, J.W., Ro, Y.M.: Adaptive video streaming over http with dynamic resource estimation. J. Commun. Netw. **15**(6), 635–644 (2013)
13. Wang, W., Zhu, M., Wang, J., Zeng, X., Yang, Z.: End-to-end encrypted traffic classification with one-dimensional convolution neural networks. In: 2017 IEEE International Conference on Intelligence and Security Informatics (ISI), pp. 43–48. IEEE (2017). https://doi.org/10.1109/isi.2017.8004872
14. Wang, Z.: The applications of deep learning on traffic identification. BlackHat USA (2015)
15. Wright, C., Monrose, F., Masson, G.M.: Hmm profiles for network traffic classification. In: Proceedings of the 2004 ACM workshop on Visualization and data mining for computer security. pp. 9–15. ACM (2004). https://doi.org/10.1145/1029208.1029211
16. Yu, D., Deng, L.: Deep neural network-hidden markov model hybrid systems. In: Automatic Speech Recognition, pp. 99–116. Springer, Berlin (2015). https://doi.org/10.1007/978-1-4471-5779-3_6