An Improved Stacked Auto-Encoder for Network Traffic Flow Classification

Peng Li, Zhikui Chen, Laurence T. Yang, Jing Gao, Qingchen Zhang, and M. Jamal Deen

Abstract

Network flow classification plays a very important role in various network applications and is a fundamental task in network flow control. However, the innovations in the multi-source network application and the elastic network architecture with the network flows of high volume, velocity, variety, and veracity pose unprecedented challenges on accurate network flow classification. In this article, an improved stacked auto-encoder is proposed to learn the complex relationships over the multi-source network flows by stacking several basic Bayesian auto-encoders. Specifically, to model the uncertainty contained in the network flows, the Bayesian auto-encoder is trained on the objects using the unsupervised learning strategy. Furthermore, the stacked auto-encoder is trained by the back-propagation algorithm using the supervised learning strategy to capture the complex relationships over the network flows. Finally, to assess the performance of the improved model, extensive experiments are conducted on two synthetic datasets based on the representative network flow datasets, that is, MAWI and DARPA 99. The results demonstrate that the improved stacked auto-encoder outperforms the traditional one in terms of classification accuracy.

Introduction

Network flow classification is a fundamental task in network flow control [1]. It plays a very important role in various network applications, such as network security monitoring, network flow accounting, and network resource allocation [2]. However, innovations in the multi-source network application and the elastic network architecture with network flows of high volume, velocity, variety, and veracity pose unprecedented challenges in accurate network flow classification. In particular, the current ultra-dense networks caused by the fast increase of smart mobile devices of the Internet of Things together with edge computing devices have greatly raised the dynamic characteristics of the network topology [3]. Furthermore, the big network flows between multi-source network applications increase the scale, difference, and uncertainty of network flows [4, 5]. Therefore, effective and efficient network flow classification requires novel models and architectures.

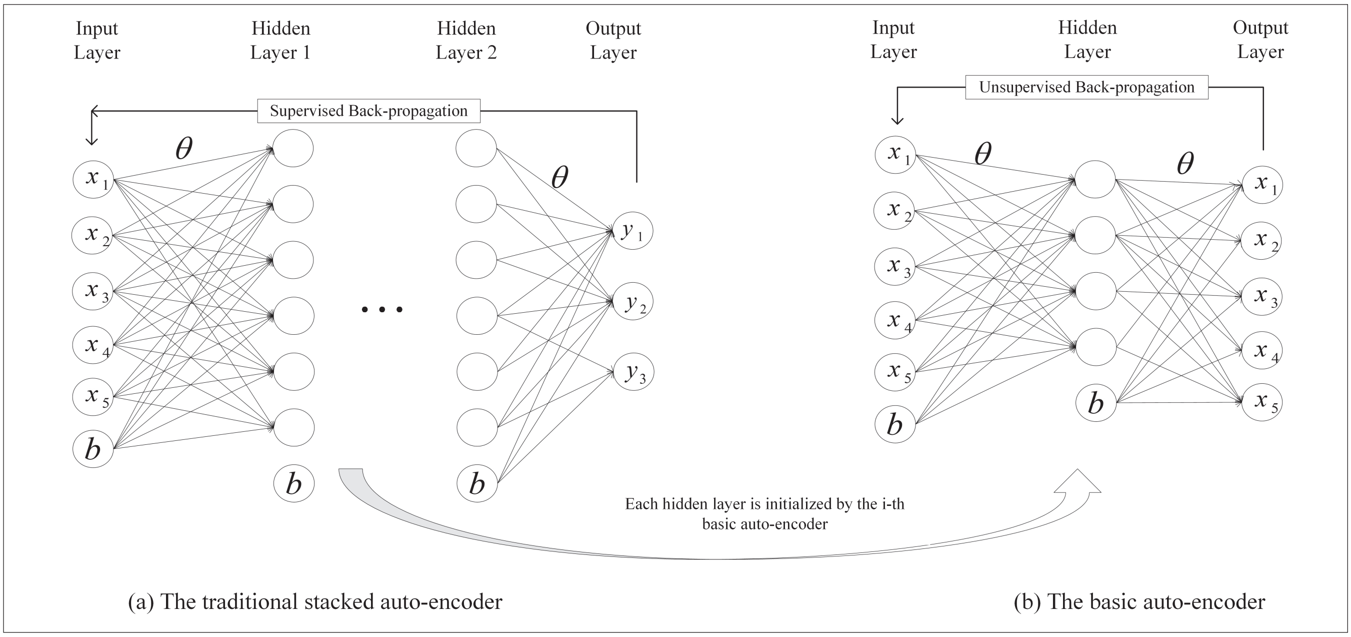
In the past few years, many network traffic classification methods were proposed to improve the performance of network-flow-based applications [1, 6]. The early methods mainly focused on the rules of the application port, such as the well-known http port. These kinds of methods cannot get the desired performance due to the appearance of dynamic allocation and disguised technologies of the port [7]. For example, it is well known that port 22 is used for SSH. A virus application can change the port information of its flow to 22. To address this problem, some machine learning methods have been introduced to classify the network flows, such as the hidden Markov method, the Bayesian method, and the neural network method [8]. These machine learning methods employ the statistical features of the network flow instead of the port information. They have made some progress in network flow classification. However, these methods cannot yield the desired results for the big network flow produced by multi-source network applications since they cannot fully capture the deep relationship over the big network flow.

Deep learning can learn the deep features of big data by using supervised, semi-supervised, and unsupervised strategies with the nonlinear activation function [9]. It achieves state-of-the-art performance in image recognition and natural language processing, as well as network detection. However, it cannot perform well in the big network flow since the deep learning model cannot handle the uncertainty contained in the big network flow.

In this article, an improved stacked auto-encoder (ISAE) is proposed to learn the complex relationships over the multi-source network flows by stacking several basic Bayesian auto-encoders. To model the uncertainty contained in the network flows, the Bayesian auto-encoder is trained on the objects by the unsupervised learning strategy. More specifically, the object is fed into the auto-encoder to get the prior distribution. Then the Bayesian probability theory is used to compute the posterior distribution of the parameters of the model. Finally, the parameters are adjusted by maximizing the log probability function of the posterior distribution. After the initial unsupervised learning, the SAE is further trained using supervised learning to capture the complex relationships of the network flows. Finally, extensive experiments are conducted on two synthetic datasets based on two representative network flow datasets, MAWI [10] and DARPA 99 [11]. The results demonstrate that the ISAE outperforms the traditional one in terms of classification accuracy.

Peng Li is with Dalian University of Technology; Zhikui Chen and Jing Gao are with Key Laboratory for Ubiquitous Network and Service Software of Liaoning Province and Dalian University of Technology; Laurence T. Yang and Qingchen Zhang are with St. Francis Xavier University; M. Jamal Deen is with McMaster University.

Digital Object Identifier:
10.1109/MNET.2018.1800078



(a) The traditional stacked auto-encoder

(b) The basic auto-encoder

FIGURE 1. The deep stacked auto-encoder paradigm: a) a standard auto-encoder. Each hidden layer is initialized as a basic building block auto-encoder. After initializing, it is fine-tuned by supervised back-propagation computation; b) an auto-encoder that is trained by unsupervised back-propagation computation.

The main contributions of this work can be summarized in three aspects.

- The traditional SAE cannot learn the effective relationship over the unbalanced multi-source network flows well. To solve this problem, an ISAE is proposed to classify the multi-source network flows by stacking the basic Bayesian auto-encoders.
- The traditional auto-encoder cannot handle the uncertainty contained in the network flow. To address this problem, the Bayesian auto-encoder is introduced to model uncertainty of the network flows by using the Bayesian probability model to predict the parameters of each basic auto-encoder.
- To evaluate the performance of the ISAE, extensive experiments are carried out on the two synthetic datasets based on the representative network flow datasets, MAWI and DARPA 99. The results demonstrate the effectiveness of the ISAE.

The rest of the article is organized as follows. In the following section, a summary of the SAE and the Bayesian theory, as preliminaries of the ISAE is provided. Then a description of the proposed improved model is given. The experiment results are then provided. Finally, the conclusions are stated.

Preliminaries

In this section, the SAE and the Bayesian theory are described briefly as the preliminaries of the ISAE model. Specifically, the SAE is used as the basic architecture in the improved model, while the Bayesian probability theory is used to train the initialized parameters of each hidden layer.

Stacked Auto-Encoder

The SAE, one of the deep learning models, has been widely used in many domains, such as traffic control and computer vision [10]. It can learn the hierarchical features of the input by stacking several basic auto-encoders to transform the input to the output with the least distortion. The SAE is characterized by unsupervised learning of data. It captures the intrinsic representations of the data by executing the two-stage stochastic gradient algorithm, that is, unsupervised pre-training and supervised fine-tune training. The former is used to initialize the weights and biases of each hidden layer to prevent the deep learning model from converging to the local extremum. The latter is used to further train the weights and biases to obtain the final features of the input data. A typical SAE is shown in Fig. 1.

As shown in Fig. 1, given the dataset $\{(x^{(1)}, y^{(1)}), ..., (x^{(k)}, y^{(k)})\}$, each hidden layer h^i is obtained from the corresponding auto-encoder AE^i. In the each AE^i, the input feature I^i is encoded by the following expression:

$$a^i = f(W_e^i I^i + b_e^i), \quad (1)$$

where a^i is the activation, f is the nonlinear function, that is, the sigmoid function and the ReLU function, b_e^i is the encoder bias vector, and W_e^i is the encoder weight that is used to initialize the weight of the hidden layer h^i.

Afterward, the activation is decoded to the output feature in the following form:

$$O^i = f(W_d^i a^i + b_d^i), \quad (2)$$

where O^i is the activation, f is also the nonlinear function, b_e^i is the decoder bias vector, and W_d^i is the decoder weight.

After the unsupervised pre-training of each hidden layer, the deep architecture stacked by the pre-trained layers will be trained on the labeled samples. The training is performed in a supervised fashion to minimize the total loss function, such as the squared error function and the cross-entropy cost function.

Although the SAE has achieved state-of-the-art performance, there are no clear measures to deal with the uncertainties in the deep learning model. In this work, the Bayesian method is designed to train the parameters, which takes the uncertainty of the model into consideration.

Although the SAE has achieved state-of-the-art performance, there are no clear measures to deal with the uncertainties in the deep learning model. In this work, the Bayesian method is designed to train the parameters, which takes the uncertainty of the model into consideration.

Bayesian Theory

The Bayesian theory is an effective classification approach based on statistics and probability theory [11]. It has been adopted across a wide spectrum of applications, such as medical diagnostics, image processing, and autonomous driving. In the Bayesian theory, there are two representative methods, that is, the naive Bayesian model [12] and the Bayesian network [13]. The former associates a sample with a particular class of membership by summarizing the probabilities of membership for each feature with each feature being equally valuable and independent. The latter is a kind of probabilistic graphical model in which the variables and their conditional dependence are modeled as a directed acyclic graph.

Generally, the Bayesian theory is expressed mathematically in the following form:

$$P(\alpha|\beta)=\frac{P(\beta|\alpha)P(\alpha)}{P(\beta)}, \qquad (3)$$

where α and β are random events. In the above equation, there are the following terms:

- $P(\alpha)$ is the prior probability indicating the likelihood of the hypothesis α before observing any data.
- $P(\alpha \mid \beta)$ is the posterior probability indicating the likelihood of the hypothesis α under the condition β. Usually, it is the probability of a learning model given the observed data.
- $P(\beta \mid \alpha)$ is the likelihood of the hypothesis β under the condition α. Usually, it is the probability of the observed data given the learned model.
- $P(\beta)$ is the marginal probability, which is the same for all possible hypotheses.

There are two probability rules in the Bayesian theory: the product rule and the sum rule. The probability product rule is used to compute the joint probability of event α and event β under a certain condition λ. It is expressed mathematically in the following form:

$$P(\alpha, \beta \mid \lambda) = P(\alpha \mid \beta, \lambda)P(\beta \mid \lambda). \qquad (4)$$

The probability sum rule is utilized to compute the marginal probability of the event α. It is expressed mathematically as follows:

$$P(\alpha|\lambda)=\sum_{\beta}P(\alpha,\beta|\lambda)=\sum P(\alpha|\beta,\lambda)P(\beta|\lambda). \qquad (5)$$

In this work, the Bayesian theory is used to predict the final weights and biases of the basic auto-encoder to initialize each hidden layer of the SAE, considering the uncertainty of the deep learning model.

The Improved Stacked Auto-Encoder

In this section, the ISAE is designed to learn the hierarchical representation of the network flow, which aims to address the uncertainty caused by the model and inputs. Specifically, there are two phases in the ISAE: the greedy layer-wise pre-training and the fine-tuning training. In the pre-training phase, the parameters of each hidden layer are first trained on the network flow data in an unsupervised learning method based on the probability theory layer by layer. Then the parameters are further adjusted to capture the final features of the network data flow in the supervised learning method.

Neural Network as the Probability Model

A typical deep learning model is composed of the input layer, which is used to obtain the object features, the hidden layers, which capture the multi-layer features, and the output layer, which outputs the object label. Specifically, the first layer receives the features of the network flow in our problem. The last layer outputs the probable label of the class for the input network flow. Each middle layer of the deep learning model represents one-layer features. In the deep learning model, each node of those layers except the input layer connects to each node of the previous layer. Each node receives the sum of the weighted activation of each node of the previous layer. Then the sum is mapped to the activation by the nonlinear function.

More specifically, given M objects $\{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \dots, (x^{(M)}, y^{(M)})\}$ and a three-layer model, each activation a_i^1 of the hidden layer is the nonlinear mapping of the sum of the weighted input feature. It can be computed by Eq. 1 with the parameters w_{ij}^1 and b_i^1. Parameter w_{ij}^1 is the weight of each connection where i and j are the indices of the hidden node and the input feature, respectively. Parameter b_i^1 is the bias of a hidden layer that always outputs the constant one. In the output layer, each activation a_i is computed in the same way.

In contrast, the activation a_i of each node in the output layer is modeled as the probability describing membership that the input object belongs to the class i, given the hypothesis of model and the input object. In the modeling process, the output probability needs to meet two conditions:

- The value of the output probability should fall in the interval [0,1].
- The sum of all the output is always equal to one.

To meet these conditions, the softmax function is applied to the output layer, which makes the deep learning neural network model behave as the probability model. Thus, the output of the model is interpreted as the probability $p(C \mid x, W, H)$, in which C is the class label, x is the input object, W is the weight matrix of the deep learning model, and H is the architecture of the deep learning model. Thus, the forward process of the probability deep learning model is as follows:

- Given the input object vector, the activation vector of each layer is computed layer by layer.
- After obtaining the last layer activation vector, the output probability vector applies softmax to the last layer activation vector.

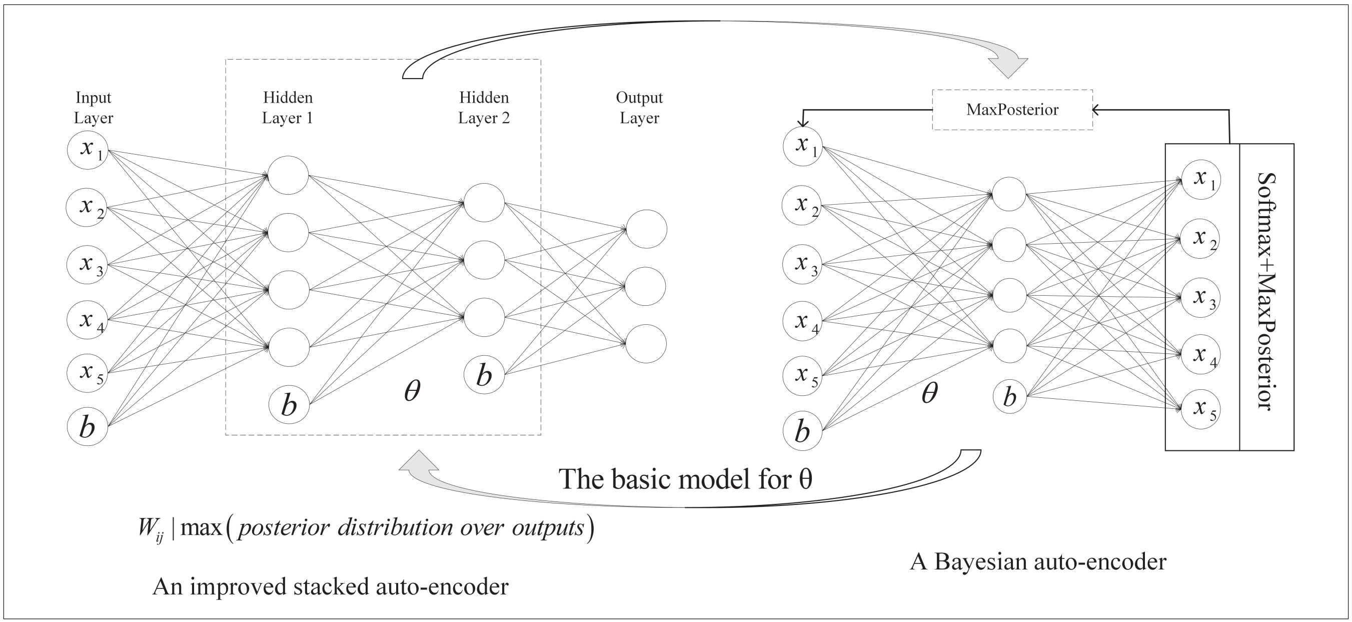


FIGURE 2. The improved deep stacked model and the Bayesian auto-encoder.

The Learning of the Probability Neural Network

The learning of the network aims to find a set of weights to fit the dataset such that the deep learning model can produce the correct classification label when fed a test object. In supervised learning, the traditional method is to define a loss function, such as the squared error function and the cross-entropy cost function. This function measures the difference between the real values of the training objects and the output value produced by the neural network with corresponding training objects. Then the weights and biases of the neural network are adjusted to minimize the loss of the neural network.

Unlike the traditional learning method, the Bayesian probability training method is used to maximize the likelihood that each training sample is assigned with the correct class label. In detail, the weights and biases of the neural network model are assigned random numbers that are subjected to a prior probability distribution such as the standard normal distribution. Then the posterior probability distribution of the weights and biases of the network is constructed as Eq. 3 based on the prior distribution and observed training objects by using the Bayesian theory [14, 15]. Finally, updating the parameters of the network is obtained by maximizing the posterior probability distribution over the parameter space.

The Bayesian training method is to find the set of weights and biases of the model over the parameter space. In other words, the weights and biases with the largest probability are used as the final parameters. The Bayesian training method can prevent overfitting of the model and deal with the uncertainty of the model and dataset naturally in terms of the probability.

The Improved Stacked Auto-Encoder

As shown in Fig. 2, the ISAE is designed to learn the hierarchical features of the network flow, which naturally takes the uncertainty into consideration. Similar to the traditional SAE, it is composed of several basic auto-encoders by stacking one after another. It is also trained in two phases: the layer-wise pre-training and the fine-tuning training. Here, each basic auto-encoder is interpreted as the probability neural network. In the unsupervised layer-wise learning, the auto-encoder is trained by using the probability learning algorithm described previously. After that, the ISAE is further trained by the standard stochastic back-propagation algorithm. The Bayesian auto-encoder and ISAE have the same complexity as the auto-encoder and the traditional SAE, since they have the same model architectures.

Experiments

In this section, extensive experiments are performed on two representative datasets, MAWI [10] and DARPA 99 [11], to evaluate the performance of the ISAE, while the traditional SAE is used for comparison.

Data Preprocessing

The MAWI and DARPA 99 datasets are generated by the MAWI Working Group and Lincoln Laboratory of MIT, respectively. These two datasets are composed of various network flows, such as FTP, SSH, TELNET, MAIL, DNS, and HTTP. Details of MAWI and DARPA 99 are shown in Table 1.

As shown in Table 1, the number of each class of the network flow is unbalanced. For example, in DARPA 99, the number of DNS objects is 25,735,411, which is more than that of FTP objects. Similarly, in MAWI, the number of TELNET objects is 353, which is much less than that of other classes of network flow. This imbalance of the network flow will cause the deep learning model trained on the separate dataset to produce low average classification accuracy. In other words, the model can better capture the features of the network flow with a large number, since the parameters of the model are adjusted to fit the network flow with a large number with high probability. Thus, two synthetic datasets, SYNDA-

Name	FTP	SSH	TELNET	MAIL	DNS	HTTP	OTAHEARS
MAWI	3395	19,016	353	31,410	9,601,134	155,511	10,163,022
DARPA99	8867	72,094	463,643	173,530	25,735,411	474,282	1,633,475

TABLE 1. Statistics on MAWI and DARPA 99.

Name	FTP	SSH	TELNET	MAIL	DNS	HTTP
SYNDATA1	12,262	12,262	12,262	12,262	12,262	12,262
SYNDATA2	10,422	14,101	10,422	14,101	10,422	14,101

TABLE 2. Statistics on SYNDATA-1 and SYNDATA-2.

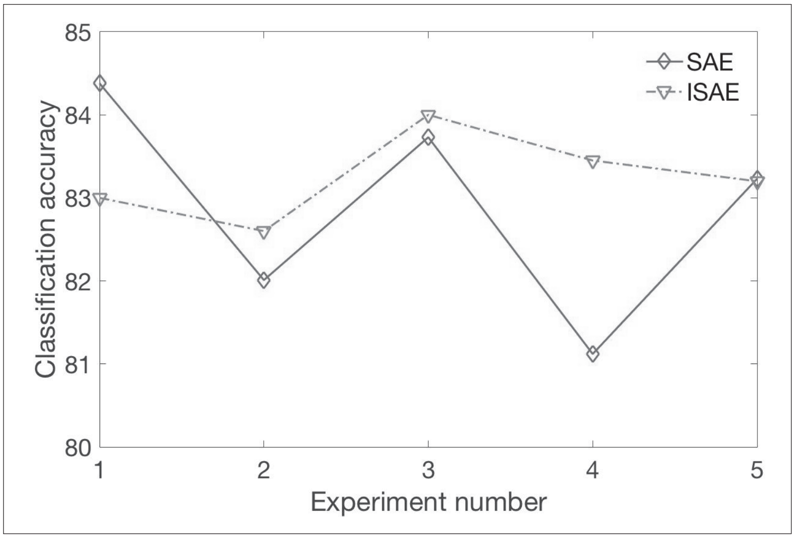


FIGURE 3. The results on SYNDATA-1.

TA-1 and SYNDATA-2, are generated based on MAWI and DARPA 99. In SYNDATA-1, each class of the network flow is of the same number, while the distribution of the network flow is still unbalanced in SYNDATA-2. Detailed descriptions of two synthetic datasets are given in Table 2.

The Results on Two Synthetic Datasets

In this experiment, the architecture with two layers is adopted to learn the features of the network flow, and the results are shown in Figs. 3 and 4.

Figure 3 shows the average classification accuracy produced by SAE and ISAE on SYNDDATA-1. In this experiment, 20 percent of objects of each class of the network flows are extracted as the test dataset, while the remainder is used to train the two models for capturing the hierarchical intrinsic representation of the network flows. There are two observations that can be made from Fig. 3. First, the ISAE can produce similar classification results to the traditional stacked auto-encoder. Specifically, the expectation of the results produced by the ISAE is 83.2 percent, which is slightly higher than that of the traditional SAE, 82.9 percent , indicating better performance of the ISAE based on the Bayesian posterior distribution. Second, the ISAE performs more stably than the traditional SAE. Specifically, the variance of the improved model is 0.27, which is much smaller than 1.73 of the traditional model. The reason is that the ISAE considers the uncertainty contained in the dataset by using the Bayesian theory, which can naturally handle the uncertainty.

Figure 4 illustrates the average classification accuracy produced by the two models, that is, SAE and ISAE, on SYNDATA-2. Similarly, 20 percent of objects of each class of the network flows are extracted to test two models, while the remainder is used as the training dataset. Figure 4 shows that the classification accuracy on SYNDATA-2 is lower than that on SYNDATA-1. The drop in accuracy is caused by the imbalance of training objects. In other words, there are classes that have larger numbers of objects than others, as shown in Table 2. The classes of the larger number have higher probability to train the parameters of the models, causing the two models to produce low classification accuracy for the classes of small numbers. Another observation is that the improved model performs better than the traditional model in most cases on this unbalanced dataset, since the improved model can effectively handle the uncertainty caused by the imbalance of the dataset.

The results in Figs. 3 and 4 demonstrate that the improved model yields better performance in terms of the average classification accuracy, indicating that it can effectively learn the features of the network flow.

Conclusion

In this article, an ISAE is introduced to capture the complex relationship over the network flow by stacking several basic Bayesian auto-encoders. An important property of the proposed model is the use of the Bayesian posterior distribution to handle the uncertainty of data in a natural way. Also, a supervised back-propagation algorithm is used to fine-tune the parameters of the ISAE. Finally, extensive experiments are carried out on two synthetic datasets based on the representative network flow datasets. The results show the potential for feature learning of network flows. In the future, tensor decomposition will be explored to promote the efficiency of the ISAE.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grants No. 61672123 and 61602083, in part by the Major Science and Technology Planning Project of Guangdong Province under Grants No. 2015B010110006, in part by the Doctoral Scientific Research Foundation of Liaoning Province No. 20170520425, and in part by the Dalian University of Technology Fundamental Research Fund under Grant No. DUT15RC(3)100. M. Jamal Deen gratefully acknowledges the support of the Canada Research Chair program.

References

[1] Y. Wang *et al.*, "Internet Traffic Classification Using Constrained Clustering," *IEEE Trans. Parallel and Distributed Systems*, vol. 25, no. 11, 2014, pp. 2932–43.

[2] V. K. Naik *et al.*, "Online Resource Matching for Heterogeneous Grid Environments," *Proc. Fifth IEEE Int'l. Symp. Cluster Computing and the Grid*, vol. 2, 2005, pp. 607–14.

[3] N. Abbas *et al.*, "Mobile Edge Computing: A Survey," *IEEE Internet of Things J.*, vol. 5, no. 1, 2018, pp.450–65.

[4] Y. Zhang *et al.*, "Home M2M Networks: Architectures, Standards, and QoS Improvement," *IEEE Commun. Mag.*, vol. 49, no. 4, Apr. 2011, pp. 44–52.

[5] X. Liu *et al.*, "Privacy-Preserving Outsourced Calculation Toolkit in the Cloud," *IEEE Trans. Dependable and Secure Computing*, 2018. DOI: 10.1109/TDSC.2018.2816656.

[6] C. J. D' Orazio, K. K. R. Choo, and L. T. Yang, "Data Exfiltration from Internet of Things Devices: IoS Devices as Case Studies," *IEEE Internet of Things J.*, vol. 4, no. 2, 2016, pp. 524–35.

[7] W. Xiong *et al.*, "Anomaly Secure Detection Methods by Analyzing Dynamic Characteristics of the Network Traffic in Cloud Communications," *Info.Sciences*, vol. 258, 2014, pp. 403–15.

[8] A. L. Buczak and E. Guven, "A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection," *IEEE Commun. Surveys & Tutorials*, vol. 18, no. 2, 2016, pp. 1153–76.

[9] Z. M. Fadlullah *et al.*, " State-of-the-Art Deep Learning: Evolving Machine Intelligence Toward Tomorrow's Intelligent Network Traffic Control Systems," *IEEE Commun. Surveys & Tutorials*, vol. 19, no. 4, 2017, pp. 2432–55.

[10] P. Borgnat *et al.*, "Seven Years and One Day: Sketching the Evolution of Internet Traffic," *Proc. 2009 IEEE (INFOCOM*, 2009, pp. 711–19.

[11] H. B. Baravati *et al.*, "A New Data Mining-Based Approach to Improving the Quality of Alerts in Intrusion Detection Systems," *Int'l. J. Computer Science and Network Security*, vol. 17, no. 8, 2017, pp. 194–98.

[12] W. Liu *et al.*, "A Survey of Deep Neural Network Architectures and Their Applications," *Neurocomputing*, vol. 234, 2017, pp. 11–26.

[13] J. He and A. Kolovo, "Bayesian Maximum Entropy Approach and Its Applications: A Review," *Stochastic Environmental Research and Risk Assessment*, vol. 32, no. 4, 2018, pp. 859–77.

[14] D. J. C. MacKay, "Bayesian Methods for Neural Networks: Theory and Applications," 1997; http://www.inference.phy.cam.ac.uk/mackay/cpi short.pdf.

[15] S. F. Gull and J. Skilling, "Quantified Maximum Entropy: Memsys5 Users' Manual, 1999; http://www.maxent.co.uk/documents/MemSys5_manual.pdf.

Biographies

Peng Li received his B.E. degree in electronic and information engineering from Dezhou University, China, in 2012. He is currently working toward a Ph.D. degree in software engineering at Dalian University of Technology, China. His research interests include deep learning and big data.

Zhikui Chen received his B.E. degree in mathematics from Chongqing Normal University, China, in 1990 and his Ph.D. degree in solid mechanics from Chongqing University in 1998. He is currently a professor at Dalian University of Technology. His research interests include the Internet of Things and big data.

Laurencen T. Yang received his B.E. and B.S degrees in computer science and technology, and applied physics from Tsinghua University, Beijing, China, and his Ph.D. degree in computer science from the University of Victoria, British Columbia, Canada. He is currently a professor at St. Francis Xavier University, Antigonish, Nova Scotia, Canada. His research interests include parallel and distributed computing, embedded and ubiquitous/pervasive computing, and big data. His research has been supported by the National Sciences and Engineering Research Council and the Canada Foundation for Innovation.

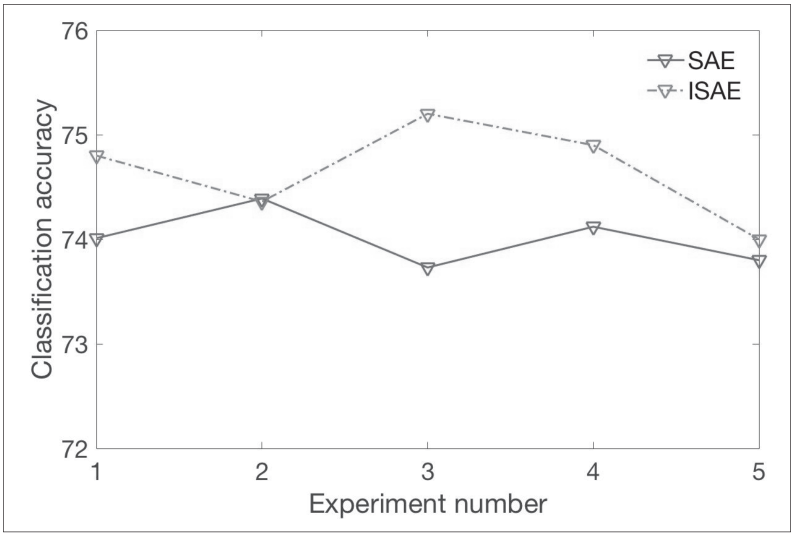


FIGURE 4. The results on SYNDATA-2.

Jing Gao received her B.E. degree in computer science and technology and her Ph.D. degree in computer software and theory from Harbin Institute of Technology, China, in 2008 and 2015, respectively. She is currently an assistant professor with the School of Software Technology, Dalian University of Technology. Her current research interests include multi-modal data mining and deep learning.

Qingchen Zhang received his Ph.D. degree in software engineering from Dalian University of Technology in 2015. He is currently an assistant professor at St. Francis Xavier University. His research interests include cloud computing, deep learning, and big data.

M. Jamal Deen completed his Ph.D. degree in electrical engineering and applied physics at Case Western Reserve University, Cleveland, Ohio, in 1985. His research interests include nano/optoelectronics, nanotechnology, data analytics, and their emerging applications in health and environment. He is a Distinguished University Professor and the Senior Canada Research Chair of information technology at McMaster University, Hamilton, Ontario, Canada.