information

Article

A Big Network Traffic Data Fusion Approach Based on Fisher and Deep Auto-Encoder

Xiaoling Tao [1,2,*], Deyan Kong [1], Yi Wei [2,*] and Yong Wang [2,3]

[1] Key Laboratory of Cognitive Radio and Information Processing, Guilin University of Electronic Technology, Guilin 541004, China; kcykdy@163.com

[2] Guangxi Colleges and Universities Key Laboratory of Cloud Computing and Complex Systems, Guilin University of Electronic Technology, Guilin 541004, China; wang@guet.edu.cn

[3] Guangxi Key Laboratory of Trusted Software, Guilin University of Electronic Technology, Guilin 541004, China

[*] Correspondence: txl@guet.edu.cn (X.T.); 4128wy@163.com (Y.W.); Tel./Fax: +86-773-2290203 (X.T.)

Academic Editors: Yong Yu, Yu Wang and Willy Susilo
Received: 27 January 2016; Accepted: 7 March 2016; Published: 23 March 2016

Abstract: Data fusion is usually performed prior to classification in order to reduce the input space. These dimensionality reduction techniques help to decline the complexity of the classification model and thus improve the classification performance. The traditional supervised methods demand labeled samples, and the current network traffic data mostly is not labeled. Thereby, better learners will be built by using both labeled and unlabeled data, than using each one alone. In this paper, a novel network traffic data fusion approach based on Fisher and deep auto-encoder (DFA-F-DAE) is proposed to reduce the data dimensions and the complexity of computation. The experimental results show that the DFA-F-DAE improves the generalization ability of the three classification algorithms (J48, back propagation neural network (BPNN), and support vector machine (SVM)) by data dimensionality reduction. We found that the DFA-F-DAE remarkably improves the efficiency of big network traffic classification.

Keywords: big network traffic data; data fusion; Fisher; deep auto-encoder

1. Introduction

Nowadays, to enhance network security, a variety of security devices are used, such as firewall, intrusion detection system (IDS), intrusion prevention system (IPS), antivirus software, security audit, *etc.* Though all kinds of monitoring approaches and reporting mechanisms provide big data for network management personnel, the lack of effective network traffic data fusion has become a stumbling block to solve different issues in network security situation awareness (NSSA) In such circumstances, the research on data fusion as one of the next generation security solutions has enough academic value and comprehensive practical value.

Data fusion in NSSA aims to effectively eliminate the redundancy of big network traffic data by feature extraction, classification, and integration. Thereby network management personnel can realize situational awareness quickly. Therefore, how to build a suitable data fusion algorithm is one of the important issues in NSSA. Feature extraction is the key of the data fusion algorithm because its performance directly affects the result of fusion. The feature extraction, as a preprocessing method to overcome dimension disaster, aims at extracting a few features that can represent the original data from big data by analyzing its internal characteristics. The classic methods include principal components analysis (PCA) [1], linear discriminant analysis (LDA) [2], Fisher score [3], *etc.*

In 2006, the significant technological achievement to effective training tactics for deep architectures [4] came with the unsupervised greedy layer-wise pre-training algorithm that was closed behind supervised fine-tuning. Since then, denoising auto-encoders [5], convolutional neural networks [6], deep belief networks [7], *etc.*, and other deep learning models have been put forward as well. Currently, deep learning theory has been successfully applied to a variety of real-world applications, including face/image recognition, voice search, speech-to-text (transcription), spam filtering (anomaly detection), E-commerce fraud detection, regression, and other machine learning fields.

In this paper, a novel network traffic data fusion approach based on Fisher and deep auto-encoder (DFA-F-DAE) is proposed to reduce the data dimensions and the complexity of computation, and it is helpful for handling big network traffic data validly. The experimental results indicate that, the proposed approach improves the generalization ability of the classification algorithms by data dimensionality reduction. Furthermore, it can reduce the redundancy of big network traffic data. Under the premise of ensuring the classification accuracy, the DFA-F-DAE reduces the time complexity of classification.

The rest of this paper is organized as follows. Section 2 describes related works. Section 3 reviews the concept of Fisher and deep auto-encoder. In Section 4, the data fusion based on Fisher and the deep auto-encoder is proposed. The experimental results and discussion are covered in Section 5. Finally, the conclusion and future work are presented in Section 6.

2. Related Work

Network security issues are more prominent with each passing day, which has become a key research topic which needs to be dealt with urgently [8]. In 1999, Bass proposed the concept of NSSA [9]. Its main goal is to obtain the macro level of information from multiple network security information by extracting, refining, and fusing. Then it can help administrators to deal with various kinds of security problems in the network. Soon after, Bass proposed a framework of intrusion detection based on multi-sensor data fusion, and pointed out that the next generation network management system and intrusion detection system will interact in the unified model. Thus it can fuse the data into information to help the network administrators make decisions. Since the objects of NSSA mostly are data information, the research of data fusion in NSSA [10,11] has gradually become a developmental trend.

Data fusion technology dated from 1970s, it was mainly engaged in the military area. As the technology is developing in a high speed, data fusion technology gradually extended to civilian areas, and has been widely employed in urban mapping [12], forest-related studies [13], oil slick detection and characterization [14], disaster management [15], remote sensing [16] and other fields. Of course, all sorts of data fusion approaches were proposed. Li *et al.* [17] proposed a fusion mechanism MCMR based on trust, which considered historical and time correlation and draws up situation trust awareness rule on historical trust and current data correlation. Papadopoulos *et al.* [18] used a data fusion method to present SIES, a scheme that solves exact SUM queries through a combination of homomorphic encryption and secret sharing. A distributed data fusion technique is provided by Akselrod *et al.* [19] in multi-sensor multi-target tracking. A few examples which introduce Fisher into data fusion are as follows. Zeng *et al.* [20] proposed a sensor fusion framework based adaptive activity recognition and dynamic heterogeneous, and they incorporated it into popular feature transformation algorithms, e.g., marginal Fisher's analysis, and maximum mutual information in the proposed framework. Chen *et al.* [21] introduced the finite mixture of Von Mises-Fisher (VMF) distribution for observations that are invariant to actions of a spherical symmetry group. The approach reduced the computation time by a factor of 2. Yong Wang [3] described an interpolation family that generalizes the Fisher scoring method and proposed a general Monte Carlo approach in dimensionality reduction.

Recently, deep learning has attracted wide attention again since the efficient layer-wise unsupervised learning strategy is proposed to retrain this kind of deep architecture. Deep learning focuses on the deep structure of neural networks, with the purpose of realizing a machine which has cognitive capabilities similar to those of the human brains. In 2006, Hinton *et al.* proposed deep belief nets (DBN) [4], which were composed of multiple logistic belief neural networks and one restricted Boltzmann machine (RBM). In recent years, deep learning has been successfully applied to various applications, such as dimensionality reduction, object recognition, and natural language processing. For example, Bu *et al.* [22] proposed to fuse the different modality data of 3D shapes into a deep learning framework, which combined intrinsic and extrinsic features to provide complementary information so better discriminability could be reached. It is better to mine the deep correlations of different modalities. Gu *et al.* [23] used the quasi-Newton method, conjugate gradient method and the Levenberg-Marquardt algorithm to improve the traditional BP neural network algorithm, and eventually got converged data, as well as improved traffic flow accuracy. Furthermore, in [24], speech features were used as input into a pre-trained DBN in order to extract BN features, though the DBN hybrid system outperforms the BN system. Although varieties of deep learning algorithms have been applied in the field of data fusion, there are a few studies of auto-encoder algorithms in the data fusion field. Felix *et al.* [25] investigated blind feature space de-reverberation and deep recurrent de-noising auto-encoders (DAE) in an early fusion scheme. Then they proposed early feature level fusion with model-based spectral de-reverberation and showed that this further improves performance. A sparse auto encoder (SAE) has proven to be an effective way for dimension reduction and data reconstruction in practices [26].

3. Preliminaries

3.1. Fisher Score

Classical Fisher Score is a well-known method to establish a linear transformation that maximizes the ratio of between-class scatter to average within-class scatter in the lower-dimensional space. The Fisher Score [27] is a classical algorithm widely engaged in statistics, pattern recognition, and machine learning. In statistical pattern recognition, the Fisher Score is used to reduce the dimension of a given statistical model, by searching for a transform. F is the class-to-class variation of the detected signal divided by the sum of the within-class variations of the signal, and F is defined as follows [28].

$$F = \frac{\sigma_{between}}{\sigma_{within}} \tag{1}$$

where $\sigma_{between}$ is the class-to-class variation, and σ_{within} is the within-class variation.

$$\sigma_{between} = \sum \frac{(\bar{x}_i - \bar{x})n_i}{(k-1)} \tag{2}$$

where n_i is the number of measurements in the ith class, $\bar{x}_i$ is the mean of the ith class, $\bar{x}$ is the overall mean, and k is the number of classes.

$$\sigma_{within} = \frac{\sum(\sum(\bar{x}_{ij} - \bar{x})^2) - (\sum(\bar{x}_i - \bar{x})n_i)}{(N-k)} \tag{3}$$

where $\bar{x}_{ij}$ is the ith measurement of the jth class, and N is the total number of sample profiles.

3.2. Deep Auto-Encoder

An auto-encoder (AE) is a professional neural network composed of three layers, including an input layer, hidden layer (because its values are not observed in the training set), and an output layer. The output of the second layer acts as a compact representation or "code" for the input data. The function of AE is much like principal component analysis (PCA) but AE works in a non-linear fashion. Auto-encoders are unsupervised learning algorithms that attempt to reconstruct visible layer data in the reconstruction layer. The idea of AE was extended to several other variants such as deep AE [29], sparse AE, denoising AE [5] and contractive AE [30]. All of these ideas have been formalized and successfully applied to various applications, and have even taken an important part of deep learning. An AE is shown in Figure 1.

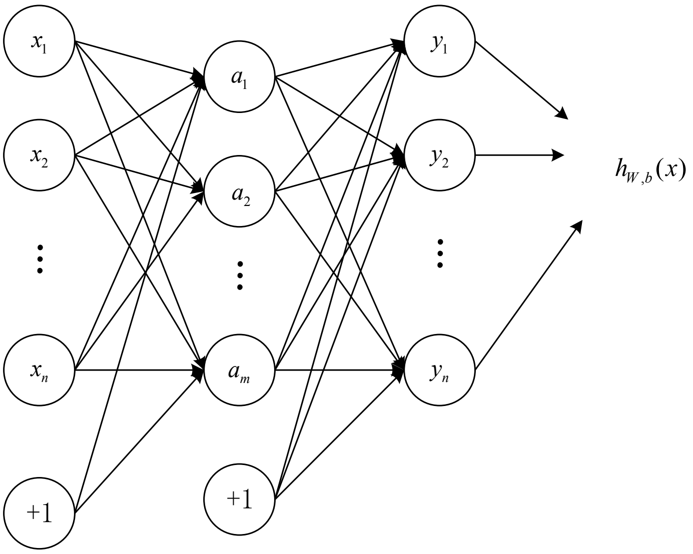

Figure 1. The architecture of auto-encoder.

Suppose a set of unlabeled training samples $x = (x_1, x_2, \cdots, x_i)$, $i \in (1, 2, \cdots, n)$, where $x_i \in \Re^n$. AE neural network is an unsupervised learning algorithm that utilizes backpropagation, setting the target values to be equal to the inputs. This means $y_i = x_i$. AE attempts to learn a function $h_{W,b}(x) \approx x$, *i.e.*, it is attempting to learn an approximation to the identity function. In Figure 1, the circles labeled "+1" are called bias units, and correspond to the intercept term.

In our scheme, we choose $f(\bullet)$ to be the sigmoid function:

$$f(z) = \frac{1}{1+\exp(-z)} \tag{4}$$

l denotes the number of layers in our network. $W_{ij}^{(l)}$ denotes the parameter (or weight) associated with the connection between unit j in layer l, and unit i in layer $l+1$. Also, $b_i^{(l)}$ is the bias associated with unit i in layer $l+1$. s_l denotes the number of nodes in layer l (not counting the bias unit).

$a_i^{(l)}$ denotes the activation (meaning output value) of unit i in layer l. For $l = 1$, we also make use of $a_i^{(1)} = x_i$ to denote the ith input. Given a fixed setting of the parameters (W, b), our neural network defines a hypothesis $h_{W,b}(x)$ that outputs a real number. Particularly, the calculation is given by:

$$h_{W,b}(x) = a_i^{(l+1)} = f(z_i^{(l)}) = f(\sum_{j=1}^{m} W_{ji}^{l} + b_j^{l}) \tag{5}$$

where m is the number of hidden nodes.

Suppose that given a fixed training set of n training examples. The definition of the overall cost function is as follows:

$$J(W,b) = [\frac{1}{n}\sum_{i=1}^{n}(\frac{1}{2}\,||\,h_{W,b}(x_i) - y_i\,||^2)] + \frac{\lambda}{2}\sum_{k=1}^{l-1}\sum_{i=1}^{s_l}\sum_{j=1}^{s_l+1}\left(W_{ji}^{(l)}\right)^2 \tag{6}$$

where λ is weight decay parameter. The first term in the definition of $J(W, b)$ is an average sum-of-squares error term. The second term is a regularization term (also called a weight decay term) that tends to decrease the magnitude of the weights, and helps to prevent overfitting.

3.3. Fine-Tune

In order to minimize $J(W, b)$ according to the function of W and b, we initialized every parameter $W_{ij}^{(1)}$ and every $b_i^{(1)}$ to a small random value closed to 0. Then we use a Fine-tune algorithm, for instance, batch gradient descent (BGD). Gradient descent is likely to lead to local optima because $J(W, b)$ is a non-convex function. However, gradient descent usually works quite well in practice. Eventually, noted that it is important to initialize the parameters randomly, rather than to all 0's. The random initialization avoids symmetry breaking.

One iteration of gradient descent updates the parameters $W^{(l)}$, $b^{(l)}$ as follows:

Firstly, compute the error term:

$$\delta^{(l)} = ((W^{(l)})^T \delta^{(l+1)}) f'(z^{(l)}) \tag{7}$$

Secondly, compute the desired partial derivatives:

$$\nabla_{W^{(l)}} J(W, b) = \delta^{(l+1)} (a^{(l)})^T \tag{8}$$

$$\nabla_{b^{(l)}} J(W, b) = \delta^{(l+1)} \tag{9}$$

Thirdly, update $\Delta W^{(l)}$, $\Delta b^{(l)}$:

$$\Delta W^{(l)} := \Delta W^{(l)} + \nabla_{W^{(l)}} J(W, b) \tag{10}$$

$$\Delta b^{(l)} := \Delta b^{(l)} + \nabla_{b^{(l)}} J(W, b) \tag{11}$$

Finally, reset $W^{(l)}$, $b^{(l)}$:

$$W^{(l)} = W^{(l)} - \alpha[(\frac{1}{m} \Delta W^{(l)} + \lambda W^{(l)})] \tag{12}$$

$$b^{(l)} = b^{(l)} - \alpha[\frac{1}{m} \Delta b^{(l)}] \tag{13}$$

where α is the learning rate.

4. Data Fusion Approach Based on Fisher and Deep Auto-Encoder (DFA-F-DAE)

The machine-learning methods generally are divided into two categories: supervised and unsupervised. In the supervised methods, the training data is fully labeled and the goal is to find a mapping from input features to output classes. On the contrary, unsupervised methods devote itself to discovering patterns in unlabeled data such that the traffic with similar characteristics is grouped without any prior guidance from class labels. The unsupervised methods need to be further transformed into a classifier for the online classifying stage. In general, the supervised methods are more precise than the unsupervised. Instead, unsupervised methods have some significant advantages such as the elimination of requirements for fully labeled training data sets and the ability to discover hidden classes that might represent unknown applications. Furthermore, unlabeled data is not only cheap but also requires experts and special devices. It is not practical to use the traditional feature extraction method to deal with it. Therefore, in this paper, combined the robustness of traditional feature extraction method (Fisher) with the unsupervised learning advantages of deep auto-encoder, we propose a novel network traffic data fusion approach. In particular, Fisher score, as a high-efficiency filter-based supervised feature selection method, according to the feature selection criteria of the minimum intra-cluster distance and the maximum inter-cluster distance, evaluates and sorts the features by the internal properties of single feature. The architecture of DFA-F-DAE is shown in Figure 2.

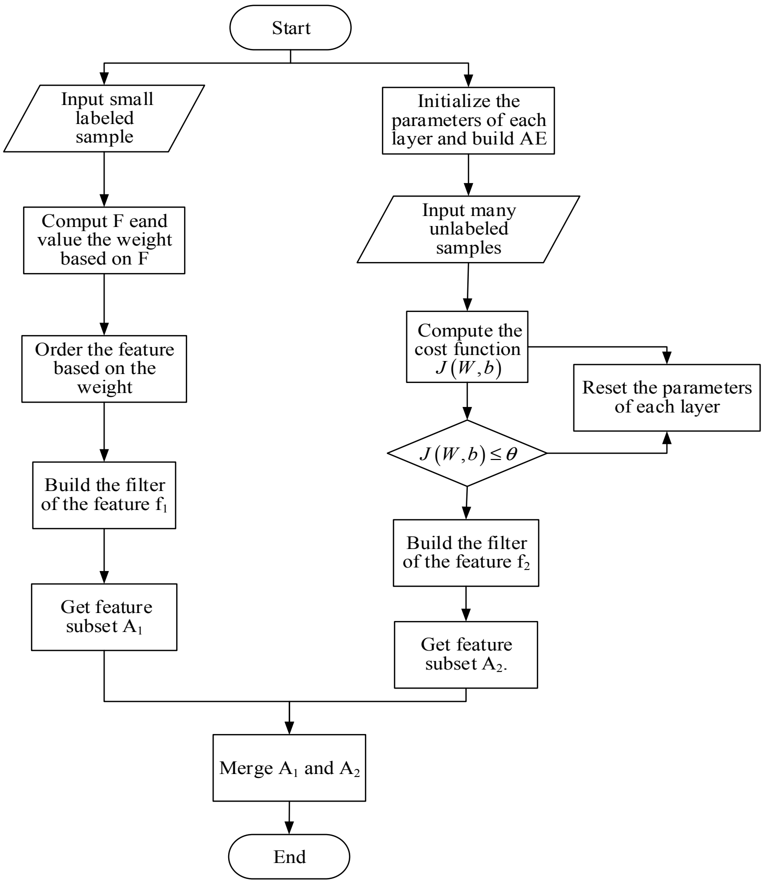
Start
Input small labeled sample
Comput F eand value the weight based on F
Order the feature based on the weight
Build the filter of the feature f1
Get feature subset A1
Initialize the parameters of each layer and build AE
Input many unlabeled samples
Compute the cost function J(W,b)
Reset the parameters of each layer
J(W,b) ≤ θ
Build the filter of the feature f2
Get feature subset A2.
Merge A1 and A2
End

Figure 2. The architecture of DFA-F-DAE.

The DFA-F-DAE aims to fuse network traffic data by two approaches (Fisher and DAE). The details are below.

Fisher:

I Input small labeled set sample.
II Use the Formula (1) to compute F and value the weight based on F.
III Order the feature based on the weight.
IV Build the filter of the feature f_1 and get feature subset A_1.

DAE:

I Initialize the parameters of each layer and build the model of AE.
II Input a large number of unlabeled samples.
III Set up the threshold value θ, then compute the cost function according to Formula (6).
IV If $J(W,b) \leqslant \theta$, the process continues. However, if $J(W,b) > \theta$, reset the parameters of each layer until $J(W,b) \leqslant \theta$.
V Build the filter of the feature f_2 and get feature subset A_2.

In the end:

Merge A_1 and A_2.

5. The Experiment Design and the Result Analysis

Below we present the datasets, experimental environment, and experimental results. The classifier is used in our experiments for the evaluation criteria.

5.1. Dataset

The 1999 DARPA IDS data set [31], or KDD99 for short, is well known as standard network security dataset, which was collected at MIT Lincoln Labs. The attacks types were divided into four categories: (1) Denial-Of-Service (DOS): Denial of service; (2) Surveillance or Probe (Probe): Surveillance and other probing; (3) User to Root (U2R): unauthorized access to local super user (root) privileges; (4) Remote to Local (R2L): unauthorized access from a remote machine. The experiment used repeated sampling to randomly extract 10,000 flows from KDD99 as train-set. Besides, the test-set contains of 500,000 flows. Since the Normal type is easy to be mistaken, we increased the proportion of Normal type in the test-set, and the other types are randomly selected. The composition of data set is shown in Table 1.

Table 1. The composition of data set.

Attacks Types	Train-Set		Test-Set	
	Number	Percentage	Number	Percentage
Normal (0)	2146	21.46%	348,413	69.6826%
Probe (1)	2092	20.92%	19,395	3.879%
DOS (2)	5164	51.64%	131,605	26.321%
U2R (3)	25	0.25%	25	0.005%
R2L (4)	573	5.73%	562	0.1124%

5.2. Experimental Environment

Experimental environment: Matlab version 8.0.0 (The MathWorks Inc., Natick, MA, USA) and Weka version 3.7.13 (University of Waikato, Waikato, New Zealand) were used as the tools in data processing and analysis in the experiments. The configuration information of node is as shown in Table 2.

Table 2. Node configuration information.

Information	
CPU	Intel i7-3770@ 3.40 GHz
Memory	16 GB
Hard Drive	256 G SSD
Operating System	Windows 7 64-bit
Java Environment	JDK 1.7.0
Matlab	version 8.0.0
Weka	version 3.7.13

5.3. Experimental Results

In order to verify the validity of the proposed DFA-F-DAE, our experimental evaluation considers two standards of classification: one is classification accuracy which symbolizes the effect of classification, while another is classification time which symbolizes the efficiency of classification.

5.3.1. Classification Accuracy under Different Dimensionalities

In order to choose proper dimensionalities by the DFA-F-DAE, we measure the performance of three classification algorithms (J48, BPNN, and SVM) under different dimensionalities. Matlab 8.0.0

and Weka 3.7.13 were used as the tools in data processing and analysis. The classification accuracy under different dimensionalities (J48, BPNN, and SVM) is described in Figure 3.

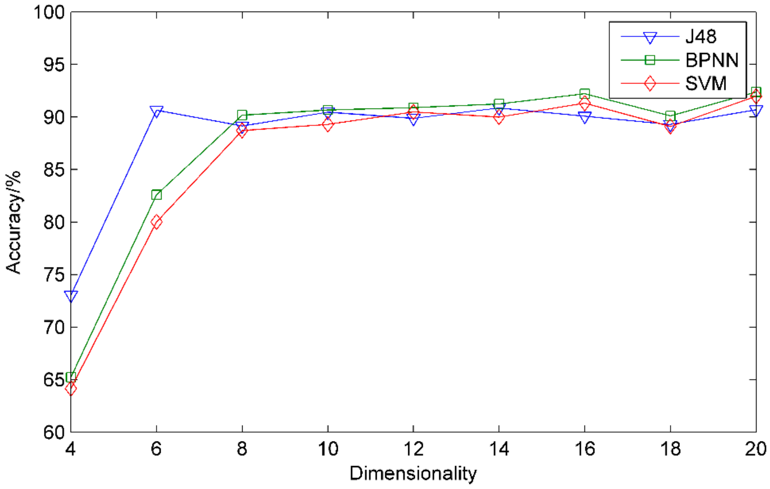

Figure 3. The classification accuracy under different dimensionalities.

It is clear that with the increasing of the number of dimensionalities, the classification accuracy performs better and better, and all the curves tend to be stable after eight dimensionalities. In addition, undesired accuracy is exhibited under smaller dimensionalities, which means that one cannot reduce the dimension infinitely. What is more, all the algorithms (J48, BPNN, and SVM) perform well under more than eight dimensionalities. So the result indicates that the DFA-F-DAE improves the generalization ability of the three algorithms by data dimensionality reduction.

5.3.2. Classification Time

For proving the effectiveness of the proposed DFA-F-DAE in terms of classification efficiency, we apply it in big network traffic data classification. The experiment compared classification times of three algorithms, which are respectively J48, BPNN, and SVM. Table 3 describes the classification time for different scale sets, which the test set increases by a factor of 1/2/3/4/5, and classification time contains of the time before fusing (B-time) and the time after fusing (A-time).

Table 3. The comparison of classification time.

Algorithm		J48	BPNN	SVM
500,000	B-time A-time	7.8 s 2.71 s	43.23 s 3.42 s	51.04 s 2.73 s
1,000,000	B-time A-time	15.68 s 5.12 s	86.32 s 6.91 s	104.47 s 5.75 s
1,500,000	B-time A-time	24.96 s 7.86 s	130.48 s 10.08 s	158.81 s 7.98 s
2,000,000	B-time A-time	30.73 s 10.47 s	169.65 s 14.12 s	214.39 s 10.78 s
2,500,000	B-time A-time	39.83 s 13.81 s	217.42 s 17.46 s	256.76 s 15.52 s
3,000,000	B-time A-time	47.9 s 16.58 s	263.03 s 20.41 s	319.53 s 18.41 s

From Table 3, it can be seen that the classification times of three algorithms after data fusion all show a sharp decline, when compare the classification times before data fusion. This is because DFA-F-DAE reduces the dimensionalities, furthermore reduces the time complexity of classification. Note that classification time of BPNN and SVM decreased more severely than that of J48, since that the dimensionalities of the test-set have a great influence on the nonlinear computation, whereas BPNN and SVM need a large number of nonlinear computation. Obviously, the DFA-F-DAE remarkably improves the efficiency of big network traffic classification.

6. Conclusions

In recent years, a few methods for data fusion have been proposed by utilized the machine learning approach, such as Dempster–Shafer (D-S), principal components analysis (PCA), *etc.* Although these methods have shown their promising potential and robustness, there are still several challenges such as the curse of dimensionality because datasets are often of high dimension. The architecture of DFA-F-DAE has been proven useful to overcome this drawback that how to reduce dimensionality and generalization error. The experimental study shows that the proposed architecture outperforms traditional methods in terms of the classification time and classification accuracy. Our future work is studying the influence of DFA-F-DAE on the classification results, which is an interesting research topic is to realize the data fusion of big data by MapReduce.

Acknowledgments: This work is supported by the National Natural Science Foundation of China (61163058 and 61363006), Guangxi Key Laboratory of Trusted Software (No. KX201306), and Guangxi Colleges and Universities Key Laboratory of Cloud Computing and Complex Systems (No. 14104).

Author Contributions: The work presented here was a collaboration of all the authors. All authors contributed to designing the methods and experiments. Deyan Kong performed the experiments. Xiaoling Tao, Yi Wei, and Yong Wang analyzed the data and interpreted the results. Xiaoling Tao and Yi Wei wrote the paper. All authors have read and approved the final manuscript.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Li, X.; Pang, Y.; Yuan, Y. L1-Norm-Based 2DPCA. *IEEE Trans. Syst. Man Cybern.* **2010**, *40*, 1170–1175.
2. Lu, G.; Zou, J.; Wang, Y. Incremental complete LDA for face recognition. *Pattern Recognit.* **2012**, *45*, 2510–2521. [CrossRef]
3. Wang, Y. Fisher scoring: An interpolation family and its Monte Carlo implementations. *Comput. Stat. Data Anal.* **2010**, *54*, 1744–1755. [CrossRef]
4. Hinton, G.E.; Salakhutdinov, R.R. Reducing the dimensionality of data with neural networks. *Science* **2006**, *313*, 504–507. [CrossRef] [PubMed]
5. Wang, D.; Tan, X. Label-Denoising Auto-encoder for Classification with Inaccurate Supervision Information. In Proceedings of the Pattern Recognition (ICPR), Stockholm, Sweden, 24–28 August 2014; pp. 3648–3653.
6. Baccouche, M.; Mamalet, F.; Wolf, C.; Garcia, C.; Baskurt, A. Sequential Deep Learning for Human Action Recognition. In *Human Behavior Understanding*; Springer: Berlin/Heidelberg, Germany, 2011; Volume 7065, pp. 29–39.
7. Tamilselvan, P.; Wang, Y.; Wang, P. Deep Belief Network Based State Classification for Structural Health Diagnosis. In Proceedings of the Aerospace Conference, Big Sky, MT, USA, 3–10 March 2012; pp. 1–11.
8. Liu, Z.; Yang, S. A Hybrid Intelligent Optimization Algorithm to Assess the NSS Based on FNN Trained by HPS. *J. Netw.* **2010**, *5*, 1076–1083. [CrossRef]
9. Tim, B. Intrusion detection systems and multi-sensor data fusion: Creating cyberspace situational awareness. *Commun. ACM* **2000**, *43*, 99–105.
10. Kokar, M.; Endsley, M. Situation awareness and cognitive modeling. *IEEE Intell. Syst.* **2012**, *27*, 91–96. [CrossRef]
11. Parvar, H.; Fesharaki, M.; Moshiri, B. Shared Situation Awareness System Architecture for Network Centric Environment Decision Making. In Proceedings of the Second International Conference on Computer and Network Technology (ICCNT), Bangkok, Thailand, 23–25 April 2010; pp. 372–376.

12. Gamba, P. Human settlements: A global challenge for EO data processing and interpretation. *Proc. IEEE* **2013**, *101*, 570–581. [CrossRef]
13. Delalieux, S.; Zarco-Tejada, P.J.; Tits, L.; Jimenez-Bello, M.A.; Intrigliolo, D.S.; Somers, B. Unmixing-based fusion of hyperspatial and hyperspectral airborne imagery for early detection of vegetation stress. *Sel. Top. Appl. Earth Obs. Remote Sen.* **2014**, *7*, 2571–2582. [CrossRef]
14. Fingas, M.; Brown, C. Review of oil spill remote sensing. *Mar. Pollut. Bull.* **2014**, *83*, 9–23. [CrossRef] [PubMed]
15. Dell, A.F.; Gamba, P. Remote sensing and earthquake damage assessment: Experiences, limits, perspectives. *Proc. IEEE* **2012**, *100*, 2876–2890. [CrossRef]
16. Dalla, M.M.; Prasad, S.; Pacifici, F.; Gamba, P.; Chanussot, J.; Benediktsson, J.A. Challenges and opportunities of multimodality and data fusion in remote sensing. *Proc. IEEE* **2015**, *103*, 1585–1601. [CrossRef]
17. Li, F.; Nie, Y.; Liu, F.; Zhu, J.; Zhang, H. Event-centric situation trust data aggregation mechanism in distributed wireless network. *Int. J. Distrib. Sens. Netw.* **2014**, *2014*, 585302. [CrossRef]
18. Papadopoulos, S.; Kiayisa, A.; Papadias, D. Exact in-network aggregation with integrity and confidentiality. *Comput. Inf. Syst.* **2012**, *24*, 1760–1773. [CrossRef]
19. Akselrod, D.; Sinha, A.; Kirubarajan, T. Information flow control for collaborative distributed data fusion and multisensory multitarget tracking. *IEEE Syst. Man Cybern. Soc.* **2012**, *42*, 501–517. [CrossRef]
20. Zeng, M.; Wang, X.; Nguyen, L.T.; Mengshoel, O.J.; Zhang, J. Adaptive Activity Recognition with Dynamic Heterogeneous Sensor Fusion. In Proceedings of the 6th International Conference on Mobile Computing, Applications and Services (MobiCASE), Austin, TX, USA, 6–7 November 2014; pp. 189–196.
21. Chen, Y.; Wei, D.; Neastadt, G.; DeGraef, M.; Simmons, J.; Hero, A. Statistical Estimation and Clustering of Group-Invariant Orientation Parameters. In Proceedings of the 18th International Conference on Information Fusion (Fusion), Washington, DC, USA, 6–9 July 2015; pp. 719–726.
22. Bu, S.; Cheng, S.; Liu, Z.; Han, J. Multimodal feature fusion for 3D shape recognition and retrieval. *IEEE MultiMed.* **2014**, *21*, 38–46. [CrossRef]
23. Gu, Y.; Wang, X.; Xu, J. Traffic data fusion research based on numerical optimization BP neural network. *Appl. Mech. Mater.* **2014**, *513–517*, 1081–1087. [CrossRef]
24. Yu, D.; Seltzer, M.L. Improved bottleneck features using pretrained deep neural networks. *Interspeech* **2011**, *237*, 234–240.
25. Felix, W.; Shigetaka, W.; Yuuki, T.; Schuller, B. Deep Recurrent De-noising Auto-encoder and Blind De-Reverberation for Reverberated Speech Recognition. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Florence, Italy, 4–9 May 2014; pp. 4623–4627.
26. Coates, A.; Ng, A.Y.; Lee, H. An analysis of single-layer networks in unsupervised feature learning. *J. Mach. Learn. Res.* **2011**, *15*, 215–223.
27. Chen, B.; Wang, S.; Jiao, L.; Stolkin, R.; Liu, H. A three-component Fisher-based feature weighting method for supervised PolSAR image classification. *Geosci. Remote Sens. Lett.* **2015**, *12*, 731–735. [CrossRef]
28. Marney, L.C.; Siegler, W.C.; Parsons, B.A. Tile-based Fisher-ratio software for improved feature selection analysis of comprehensive two-dimensional gas chromatography–time-of-flight mass spectrometry data. *Talanta* **2013**, *115*, 887–895. [CrossRef] [PubMed]
29. Lange, S.; Riedmiller, M. Deep Auto-Encoder Neural Networks in Reinforcement Learning. In Proceedings of the 2010 International Joint Conference on Neural Networks (IJCNN 2010), Barcelona, Spain, 18–23 July 2010; pp. 1–8.
30. Muller, X.; Glorot, X.; Bengio, Y.; Rifai, S.; Vincent, P. Contractive auto-encoders: Explicit invariance during feature extraction. In Proceedings of the 28th International Conference on Machine Learning (ICML-11), Bellevue, WA, USA, 28 June–2 July 2011; pp. 833–840.
31. KDD Cup 1999 Data. Available online: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html (accessed on 11 October 2015).