Neurocomputing 149 (2015) 415–425

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Anomaly detection in traffic using L1-norm minimization extreme learning machine

Yibing Wang [a], Dong Li [a], Yi Du [b], Zhisong Pan [a,*]

[a] College of Command Information System, PLA University of Science and Technology, Nanjing 210007, China
[b] Telecommunication Network Technology Management Center, Beijing 100840, China

ARTICLE INFO

Article history:
Received 5 August 2013
Received in revised form
3 April 2014
Accepted 7 April 2014
Available online 10 September 2014

Keywords:
Traffic classification
Anomaly detection
Extreme learning machine
Support vector machine
L1-norm minimization

ABSTRACT

Machine learning algorithms are widely used for traffic classification and anomaly detection nowadays, however, how to fast and accurately classify the flows remains extremely challengeable. In this paper, we propose an extreme learning machine (ELM) based algorithm called L1-Norm Minimization ELM, which fully inherits the merits of ELM, and meanwhile, exhibits the sparsity-induced characteristics which could reduce the complexity of learning model. At the evaluation stage, we preprocessed the raw data trace from trans-Pacific backbone link between Japan and the United States, and generated 248 features datasets. The empirical study shows that L1-ELM can achieve good generalization performance on the evaluation datasets, while preserving the fast learning and little human intervened advantages that ELM has.

1. Introduction

Continued development of the Internet has spawned a large number of web-based applications, which have brought tremendous network traffic and anomalous activities. Fast and accurate classification of type of traffic and abnormal behaviors on the backbone network becomes more and more important for the Internet Service Providers (ISPs) and network administrators. For example, ISPs need to monitor the constitutions of different applications so as to prioritize traffic of the QoS-sensitive application, prevent and locate harmful activities and take additional steps for other reasons, say, politics [1].

To date there mainly exists four effective fashions on traffic classification and anomaly detection: (1) transport layer port number based method, which is easy to implement. However, it becomes increasingly unreliable for the emergence of all kinds of new applications types, e.g. P2P application, which uses randomly assigned ports number that are not registered with IANA [2,3]; (2) Deep packet inspection (DPI), a more effective technique, which inspect the packet payload with existing patterns [4,2,5]. While it has higher detection accuracy, this method demands much more resources and bandwidth than ISPs and network administrators could afford and could be blocked from encryption easily. It should be deployed to the border routers instead of backbone network; (3) host behavior based method, which lies on the host system to contrast the audit records and security logs against archived host profiles. It would raise an alert once they were found unmatched [4,6]; (4) traffic flow features based methods, which prevails recently and have been proven promising for the following perspectives [7,8]. Firstly, flow features could be easily derived from packets header statistics alone, which avoid privacy issues, legal constraints and resource-intensive requirements. Secondly, in the face of obstacles of encryptions and so on, flow based features could almost fully characterize the different applications by using the machine learning methodologies in the pattern recognition field. Thirdly, many efficient machine learning algorithms [9–19] have already been applied to dealing with traffic classification problems, which enriched the theoretical foundations while providing comprehensive applications and analysis.

There are many state-of-the-art methodologies applied in the traffic classification field. Erman et al. [11] used two unsupervised clustering algorithms, K-Means and DBSCAN, to demonstrate how cluster analysis can be used to effectively identify groups of traffic that are similar using only transport layer statistics. Kim et al. [1] conducted an evaluation of three traffic classification approaches: port based, host behavior based and flow features based. After comparing seven commonly used machine learning algorithms with the other two kinds of traffic classification methods, they found that Support Vector Machine (SVM) algorithm achieved the highest accuracy on every trace and application. Williams et al. [17] evaluated Naiive Bayes, C4.5, Bayesian Network and Naiive Bayes Tree algorithms to show that, although classification accuracy between the algorithms is similar, computational performance can differ significantly.

* Corresponding author.
E-mail address: hotpzs@hotmail.com (Z. Pan).

http://dx.doi.org/10.1016/j.neucom.2014.04.073
0925-2312/

Machine learning methodologies are the main solutions to the flow based traffic classification. However, different from classical data mining scenarios, traffic classification needs more attentions in terms of its underlying characteristics. A conventional definition of flow is 5-tuple (source IP, source port, destination IP, destination port, protocol), which uniquely identifies a data stream that two hosts communicate with each other in a certain time period.

As depicted above, the state-of-the-art machine learning algorithms may achieve good accuracy, e.g. SVM, on flow based traffic classification. However, these approaches are so computationally expensive that they can be hardly put into practical use unless we are content with sacrificing accuracy. Extreme learning Machine (ELM) [20,21] is a rapidly developing learning theory proposed for generalized single-hidden layer feed-forward networks (SLFNs) with distinguishing characteristics of (1) fast learning speed compared to traditional gradient-based algorithms, (2) good generalization performance on predicting multi-class labels or regression values, (3) free of human-intervened tuning parameters with randomly generated hidden node parameters (e.g. random input weights and hidden biases). In recent years, ELM theory has made considerable progress, which inspired us a lot when in the face of traffic classification, since all of these characteristics of ELM could meet the need of the tremendous growth of the infrastructure of modern Internet. Therefore, we propose an algorithm which extends the original ELM theory and framework with L1-norm minimization of the output weights vector. In this paper, our main contributions can be summarized below:

- Proposing an L1-norm minimization extreme learning machine algorithm to exploit the intrinsic data patterns of network.
- Employing an effective preprocessing including flow features extraction from raw network data trace and labeling the flows.
- Generating the labeled anomaly detection data sets with ground truth from WIDE project.

The remainder of this paper is organized as follows. After reviewing the network traffic flows in Section 2, we describe the algorithm L1-ELM in Section 3. Section 4 describes the preprocessing details, and, we evaluate our methods in Section 5. Section 6 concludes this paper.

2. Sparse representation and its application on ELM

The hidden layer neurons have powerful generalization abilities. However, due to the randomness of the input weights, many neurons may be closely correlated. Regularization is very necessary to prevent the model from over-fitting and improve the generalization capability.

L1-norm regularization has been extensively applied for its sparsity-induced capability, when training samples have high dimensionality. Nevertheless, it draws great attentions in the optimization research field. As the L1-norm regularization will lead the problem to a non-smooth and non-differentiable constrained optimization one, the problem will become much more challengeable to solve. The state-of-the-art methodologies for solving the L1-norm optimization problem can be categorized into four categories [22]. The first methodology solves the problem as a non-smooth optimization problem through sub-gradient based algorithms. The second methodology approximates the L1-norm term with a smooth formulation, so the smooth optimization algorithms can solve the problem directly. The third methodology reformulates the problem into a smooth constraint smooth optimization problem by introducing extra variables. The fourth methodology casts the problem as a smooth objective function optimization problem with a L1-ball constraint, which is applied in this paper, and it can be formulated as

$$\min_{\mathbf{x}:\|\mathbf{x}\|_1 \le z} : F(\mathbf{x}) \tag{1}$$

where $F(\cdot)$ is a smooth loss function and $z \in \mathbb{R}^+$. The building block of this methodology is to apply Euclidean projection onto the L1-ball [23,24]. In the optimization process of the L1-ELM proposed in this paper, through casting Euclidean projections as root finding problems associated with specific auxiliary function, this problem can be solved in linear time via bisection [25].

As of the popularity of ELM theories, many works have been done fully utilizing the advantages of ELM. Parallel computing in a distributed environment is effective when in the face of computation-intensive scenarios. Li et al. [26] use MapReduce model [27] to parallelize the ELM computation across large-scale clusters of machines. Benoît et al. [28] proposed a feature selection method for nonlinear models with ELM. There also exists some works that have some common ideas with ours. Decherchi et al. [29] address the implementation of the powerful ELM model on reconfigurable digital hardware. To obtain the sparse hidden neurons, they introduce a new optimization problem with hinge loss function and L1-norm regularization. The optimization problem is

$$\min_{\mathbf{w}} : \sum_{i=1}^{N} L(y_i, h(\mathbf{x}_i)\cdot\mathbf{w}) + \lambda\|\mathbf{w}\|_1 \tag{2}$$

where $h(\mathbf{x}_i)$ is the ith row of matrix H, which will be explained in the next section. $(\mathbf{x}_i, y_i)$ is the ith training sample and its corresponding target, and $L(\cdot)$ is the hinge loss function. After solving this problem, sparse hidden neurons are selected in accordance with nonzero terms in the optimal $\mathbf{w}$. When the selections are done, the original complete ELM process has to be finished afterwards. The main differences between their work and L1-ELM lie in: firstly, their contribution does not change the ELM theory itself, yet L1-ELM can obtain the sparse neurons and output weights vector in the meantime of training process, i.e. the forms and meanings of the two optimization problems are totally different. Secondly, the optimization process in [28] applies a conventional convex optimization algorithms, say, simplex method or interior point methods [30], while L1-ELM could achieve the convergence rate of $O(1/k^2)$, w.r.t. L1-norm regularization.

Miche et al. [31] proposed a methodology named optimally pruned extreme learning machine (OP-ELM), which is based on the original ELM algorithm. They firstly construct a SLFN using the ELM algorithm; then, OP-ELM apply the MRSR [32] algorithm to obtain a ranking of the hidden neurons. MRSR is an extension of the least angle regression (LARS) algorithm [33], and it is a variable ranking method, rather than directly selecting variables with a LASSO [34] solution. At the third step, OP-ELM prunes less useful neurons through leave-one-out validation. As we discussed above, the intrinsic mechanisms between OP-ELM and L1-ELM are different, although they both construct a SLFN with sparse hidden neurons. OP-ELM prunes less useful neurons by leave-one-out validation after ranking the neurons through modified LARS algorithm, while L1-ELM can obtain the sparse neurons after the objective function is optimized immediately. Thereafter, we compare these two methodologies with the convergence time and accuracy in the experiment.

Since Nesterovs method [35] is the one of the optimal first-order black-box methods for smooth convex optimization, its convergence rate can achieve $O(1/k^2)$, where k is the number of iterations. Efficient Euclidean Projection [25], which plays a building block role in the L1-ELM, makes L1-ELM achieve the convergence rate of $O(1/k^2)$, although the objective function is non-smooth other than smooth.