Multi-Scale LSTM Model for BGP Anomaly Classification

Min Cheng, Qing Li, Jianming Lv, *Member, IEEE*,
Wenyin Liu, *Senior Member, IEEE*, and Jianping Wang

Abstract—As a policy-based routing protocol, the primary purpose of Border Gateway Protocol (BGP) is to exchange routing reachability information to provide sufficient end-to-end Quality-of-Service (QoS). The constant increase of anomalous traffic of BGP affects the connectivity and reachability of routing information among different Autonomous Systems (ASs), which calls for building accurate alerting models to provide stable routing services in the Internet. The previous works classify anomalies without considering the characteristic of multiple time scales, which may lead to inaccurate classification. In this paper, we propose a novel Multi-Scale Long Short-Term Memory (MSLSTM) model to capture the anomalous behaviors from BGP traffic. In our model, a Discrete Wavelet Transform is used to obtain temporal information on multiple scales, and a hierarchical two-layer LSTM architecture is devised where the first layer learns the attentions of different time scales to generate an integrated historical representation, and the second layer captures the temporal dependency in the learned representation. To evaluate the feasibility in different alerting scenarios, we conduct comprehensive experiments based on several BGP data sets collected from real world applications. The results demonstrate that our model achieves a promising performance compared with the state-of-the-art approaches.

Index Terms—BGP, LSTM, discrete wavelet transform, multi-scale, anomaly classification

1 Introduction

BGP (Border Gateway Protocol) provides routing information for autonomous systems in the Internet. Essentially, BGP lays the foundation for all services running on top of it. Thus the correct path establishment of BGP peers is of vital importance to provide a stable environment for maintaining successful collaborations of relevant services across multiple domains in the Internet. However, various kinds of anomalous BGP behaviors such as worms, misconfiguration, and prefix hijack affect the inter-domain connectivity and degrade the performance of the Internet at large scales. For instance, on February 24, 2008, Pakistan Telecom (AS17557) started an unauthorized announcement of the prefix 208.65.153.0/24 and then its upstream providers, PCCW Global (AS3491), forwarded this announcement to the rest of the Internet, which resulted in the hijacking of YouTube traffic for two hours on a global scale [1]. Therefore, it is crucial to capture such event that deviates from expected normal behaviors to ensure the stability of the whole network and prevent further loss [2].

The detection of anomalies can be viewed as a classification problem by assigning an “anomalous” or “regular” label to traffic data record [3]. In previous works, various machine learning models have been designed to classify anomaly and have achieved desirable results [3], [4], [5]. In general, the existing works treat data instances individually without considering the temporal attributes among them. However, the current state is often affected by previous traffic. Thus, temporal attributes are important in building accurate classification models. To capture the implicit dependency in traffic data, sequential methods, especially those based on deep learning models, have been widely used due to the effective capability for representation learning. In recent works [1], [6], one-layer Long Short-Term Memory (LSTM) [7] model has been proposed to identify anomalous BGP traffic. However, the single layer architecture in [1], [6] fails to exploit the dependency information on different temporal resolutions. As shown in our previous work [1], the length of time interval over the aggregated observations in sequences, i.e., the time scale in sequences, is a key factor that impacts the performance of the model, and patterns in traffic exhibit various behaviors along different time scales.

Unlike using different time scale values as hyper-parameters in one-layer LSTM to consider multi-scale information previously [1], in this work, a novel multi-scale LSTM model (MSLSTM) is proposed to fully characterize temporal dependency of different time scales. In the MSLSTM, a Discrete Wavelet Transform is utilized to obtain temporal information on multiple scales, and a hierarchical two-layer LSTM architecture is developed, where the first layer learns the attentions of different time scales to generate an integrated historical representation, and the second layer

- *M. Cheng, Q. Li, and J. Wang are with Department of Computer Science and Technology, City University of Hong Kong, China. E-mail: mc.cheng@my.cityu.edu.hk, {itqli, jianwang}@cityu.edu.hk.*
- *J. Lv is with Department of Computer Science, South China University of Technology, Guangzhou, Guangdong 510630, China. E-mail: jmlv@scut.edu.cn.*
- *W. Liu is with School of Computer Science and Technology, Guangdong University of Technology, Guangzhou, Guangdong, China. E-mail: csliuwy@cityu.edu.hk.*

Manuscript received 27 June 2017; revised 31 Dec. 2017; accepted 22 Mar. 2018. Date of publication 10 Apr. 2018; date of current version 9 June 2021. (Corresponding author: Min Cheng.)
Digital Object Identifier no. 10.1109/TSC.2018.2824809

captures the long and short term temporal dependency in the representation. Compared with recent works [8], [9] that stack multiple LSTM layers to learn multi-scale information, our proposed model has the following advantages:

- First, MSLSTM learns multi-scale information explicitly with Discrete Wavelet Transform, while the temporal information learned by stacked LSTM is implicit and difficult to explain when it creates new representation at higher level only based on the abstraction of previous hidden layer;
- Second, instead of simply stacking several layers of LSTM, the proposed MSLSTM is more flexible titan stacked LSTM, i.e., the information on different scales are considered sirttultaneously with an attention mechanism to learn the importance of each distinct scale. The attention mechanism proposed by Bahdanau et al. [10] helps the sequential models output a summarized representation instead of only considering the representation of the final step, whtre the weight assigned to each learned value is calculated by a compatibility function of the re levance importance among the corre sponding inputs;
- At last, our model can learn multi-scale temporal depen dency using two-layer architecture with less parameters during training process, but in stacked LSTM the aggregation of multiple layers results in larger parameter spact, and it is tricky to determine tide depth of the stacking model to generate optimal representations.

To the best of our knowledge, this is the firat empirical study by integratina multi-scale analysis into Uierarchical altention LSTM to identify anomalies. Furthermore, even though in this work our main focus is tho classification of BGP anomalies, tire model can be extended easily to many other alsrt warning related services, scenarios and applicahions. The main contributions of this work cars be summarized as follows:

- First, the temporal features of network traffic on multi scale aspectr are analyzed, domonstrating that the network traffic exhibits various dis tanct patterns in dihfpront time scales. Moreover Dissreto Wavolet Transform is used to generate abundant information in addition to original features to characterize the behavior of the traffic.
- Second, a hierarchical attention LSTM model is devised to Itarn representations of multiple scales from the abundant information. In the proposed model, the information on multiple scales is considered simultaneously and the attention of each distinct time scale is learned internally. An inoegrated representation is subsequently goneratod to model the long and short-torm tomporal de pendoncy.
- Third, the performance of proposed model is evaluated with the data set from real BGP traffic. To verify the feasibility of MSLSTM, network traffic of typical BGP anomaly events is collected. For each event, aggregated features are extracted and a sliding window is used to convert original instances into multivariate time sequences. Experiment results demonstrate that taking the multi scale information into consideration plays positive impacts on the anomaly classification and MSLSTM improves the performance significantly compared with the state-of-the-art methods.

The rest of this article is organized as follows. In Section 2, we present the problem and introduce the preliminaries of the proposed model. In Section 3, the detail of proposed MSLSTM model is introduced. The extensive experimental evaluations as well as comparisons and discussions are presented in Section 4. In Section 5, we give a brief literature review which is related to our work. Finally, we conclude the paper in Section 6.

2 Problem Statement and Preliminaries

In this section, at first, the definition of the problem is presented followed by the introduction of the preprocessing steps of the model. Then, an overview of the multi scale analysis pipeline is shown. After that we provide an introduction of discrete wavelet transform. At the end, the concept of attention mechanism is presented.

2.1 Problem Definition

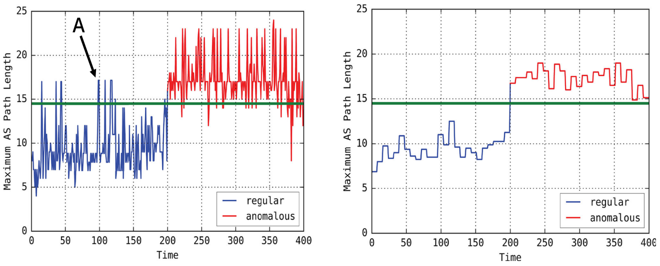
The messages in BGP update packets contain lots of information that reflects the healthiness of the network. In order to detect the anomalous traffic, it is necessary to analyze the behavior pattern of historical delta and train a classification model for the fetore identification. The problem oS the classification task can be defined as tollows: Given a list of instances in previous $\{x^d_{t_1}, x^d_{t_2}, \ldots, x^d_{t_2}\}$, the goal is to identify the state of traffic $\{x^d_{tn+1}, x^d_{tn+2}, \ldots, x^d_{tn+s}\}$ using trained classification model, where d is the dimension of the input instance and $\mathrm{t_i}$ is the temporal index. In brief $X = (\mathrm{x}_1, \mathrm{x}_2, \ldots, \mathrm{x}_\mathrm{t}, \ldots,)$ is used to stand for inputs, where $\mathrm{x_t} \in \mathbb{R}^d, t \in (\mathrm{l}, n)$.

2.2 Preprocessing

Most of the existing methods for BGP anomaly classifiation [3], [4], [5] are limited ta building traditional classifiers without considering the temporal features of traffic. However, the states of future streams have inner relationship with previous traffic as the features are aggregated from a stream of entities. The proposed model will learn the dependency relationship between instance values across batches. Assuming that the data set is a traffic stream collected with n points in the given time slot, where each element is a d-dimension vector, an overlapped sliding time window is first used to generate a series of multivariate sequences from original data set to characterize the temporal relationship of instances. Each sequence is a multivariate time series with the period of window size. The sliding window is frequently employed in data stream mining [11], [12] to capture information from the historical traffic. If the size of window is e, then the state of $\mathrm{x_i}$ is related to a sequence $\{\mathrm{x_{i-e}}, \mathrm{x_{i-e+1}}, \ldots, \mathrm{x_{i-1}}\}$. In this way, a new data set of sequences from training data is obtained, and the label of each sequence is equivalent to the class of the most frequent elements.

2.3 Multi-Scale Temporal Analysis

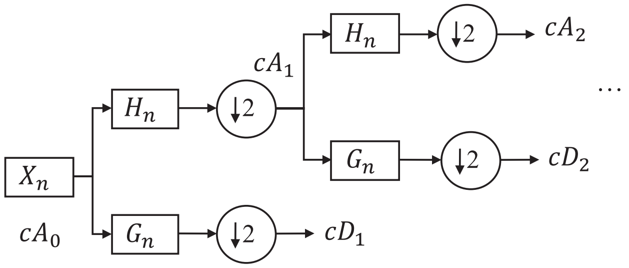
As mentioned in previous works, the Internet traffic is highly dynamic and full of burstiness and noises, which



(a)

(b)

Fig. 1. (a) A time sequence of eriginal time scale; (b) The same sequence using discrete wavelet transform at scale level 3.



Note: ↓2 means the number of coefficients is halved through the filters.

Fig. 2. Discrete wavelet transform approximations and details of time sequences.

is consistent with our observation on BGP traffic. We plot 400 samples during the period when anomalous traffic occurs in Fig. 1 where the first 200 samples stand for anomaly and the rest of the samples stand for normal. In Fig. 1a, the burstiness and fluctuations can be observed in both normal and abnormal traffic. The traditional machine learning approaches [3], [4], [5], which make the decision only based on the state of current moment, would judge the sudden burst value A in the regular part as anomaly with high possibility. However, if more information about the historical traffic is considered, it can be observed that most of the values in the past are in a relatively regular scope and the bursts are also very frequent before, then we will judge the point A should be only a sudden burst [1]. Therefore, integrating the historical information into the classifier can make the decision more cautious and accurate. Furthermore, as most of other time series, the BGP traffic has the multi-scale property, which means the subsequences exhibit distinct patterns in different time scales. For instance, in Fig. 1 where the horizontal line is the boundary between regular traffic and anomalies. In Fig. 1a, it is difficult to distinguish regular traffic from anomalies at the original granularity, however, after increasing the time scale in Fig. 1b, it is much easier to identify anomalous traffic from regular ones (the horizontal line). The large scale can demonstrate global trend of the sequence while the small scale provides more information about details. Thus, it is necessary to consider the scale when handling the traffic sequences to achieve optimal performance.

2.4 Discrete Wavelet Transform

Discrete Wavelet Transform (DWT) decomposes a signal into a set of basic functions to obtain multi-scale time sequences. These basic functions are called wavelets, which are generated from a single prototype wavelet by dilations and shifting:

$$\psi_{a,b}(t) = \frac{1}{\sqrt{a}}\psi\left(\frac{t-b}{a}\right), \tag{1}$$

where a is the scaling parameter and b is the shifting parameter. Here the multi-level discrete wavelet transform is conducted to exploit and reveal the scaling properties of the temporal dynamics in traffic sequences. For a given sequence $X_n = \{\mathrm{x}_1, \mathrm{x}_2, \ldots, \mathrm{x}_\mathrm{n}\}$, let $cA_0 = X_n$, the decomposition of discrete wavelet transform decomposes the original sequence X_n into approximation part cA_1, which represents coarse information at a higher level than original signal, and detail part cD_1, which provides information on high frequencies. This procedure will continue decomposing the approximation with the down sampling rate. At each level, the approximation is decomposed as follows:

$$cA_i = cA_{i+1}H_n + cD_{i+1}G_n, \tag{2}$$

where cA_i and cD_i are the approximation coefficient and detail coefficient respectively. The approximation co efficient is obtained by convolving the original signal with the low-pass filter H_n, while the detail coefficient is acquired by the high-pass filter G_n. Finally, the wavelet analysis of level k generates the list of following coefficients $[c_a^k, c_d^k, c_d^{k-1}, \ldots, c_d^2, c_d^1]$. And the reconstruction of the signal at scale j is calculated as:

$$R_j = \psi'\left(\sum_{i=j}^{k} cD_i + cA_k\right), j \in (1, k). \tag{3}$$

The number k denotes the total number of levels at which the decomposition will be performed, and ψ' is the reconstruction function. For a time-sequence of length n, DWT consists of $\log_2 n$ levels at most, thus $k \in [1, [\lfloor \log_2 n \rfloor]$. As an illustration of the preceding process, Fig. 2 depicts a two-level hierarchical structure of DWT. The approximations as well as details at different levels provide abundant information about the dynamics of the traffic from various aspects, where the approximation at higher level presents a global trend behavior, and the details at each level can characterize more local information. For reconstruction of different scale levels, the inverse DWT synthesizes cA_i by up-sampling two sets of coefficients at level $i + l$, inserting zeros to detail coefficients and convolutions of the up-sampled results with reconstruction filters. In this manner, DWT extracts the multi-scale features as abundant information, which is further utilized to measure each scale and to build the classification model.

2.5 Attention Mechanism

The attention mechanism originates from computational neuroscience [13], [14]. For instance, visual attention focuses on specific parts of the visual inputs to compute the adequate responses. This principle has a large impact on neural computation and a similar idea has been applied in many deep learning tasks including speech recognition, translation, reasoning, and visual identification of objects.

In deep learning, attention mechanism was proposed at first in [10] to select the reference words in the source sentences by using attention based encoder-decoder. The attention mechanism helps the decoder part output words based on a weighted combination of all the input states, not just the last state. As shown in Fig. 3, when encoding the input source

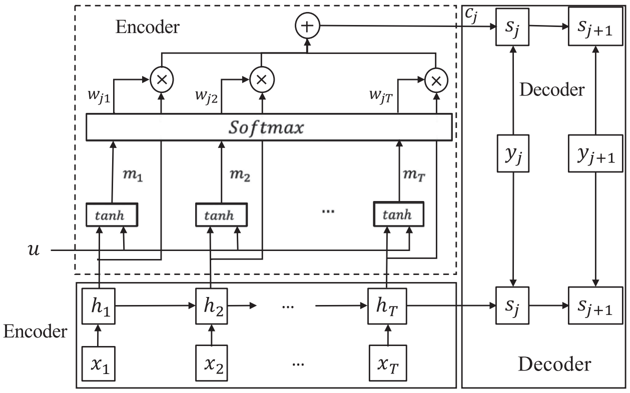


Fig. 3. A description of attention model.

sentence $X = \{x_1, x_2, ..., x_T\}, x_i \in \mathbb{R}^d$ and translate it into target Y, the traditional decoder only receives the final state of the encoder in the bottom solid-line rectangle to generate words, while with attention mechanism, i.e., some words in X might be more important than other words and those positions should be assigned greater importance when conducting sentence translation, the recombined information in the dash-line rectangle of the figure can characterize the whole input with a strong focus on the parts surrounding those more important words. Specifically, in terms of encoding part of an attention model, it takes T instances $\{x_1, x_2, .., x_T\}$ together with an additional vector u as inputs, and outputs a context vector c_j which recombines the sequence of learned hidden representation $\{h_1, h_2, .., h_T\}, h_i \in \mathbb{R}^n$, where n is the number of hidden units. The vector c_j will be further used to generate hidden representation s_j for new jth word at decoding part. Subsequently, the list of $\{m_1, m_2, .., m_T\}, m_i \in \mathbb{R}^T$ with a $tanh$ layer are calculated to project h_i into attention space to characterize different focus on the implicit dependency information. The formula is calculated as: $m_i = tanh(W_{um}u + W_{hm}h_i)$. With a $softmax$ function the weight on each position w_{ji} is learned as: $w_{ji} \propto exp(\langle s_j, m_i \rangle), \sum_i^T w_{ji} = 1$, where the value of weight w_{ji} is determined by how well the vector s_j and state m_i matches. The output c_j is a weighted summarization of all hidden states: $c_j = \sum_i^T w_{ji}h_i$, and to be further utilized with function $s_{j+1} = f(c_j, s_j)$ to generate hidden representation for new word.

3 Methodology

As illustrated in Fig. 1, the behavior of the traffic data exhibits its different patterns at different time scales. Compared with only extracting features from traffic sequences on finest granularity the information on larger scale can provide additional features to model the behavior of the sequences. In addition, the relationship among time scales also exists. For instance, the larger scale is evolved from smaller scale in terms of temporal index. To develop a model that takes full advantage of the multi-scale information, a hierarchical two-layer LSTM model is proposed. In the first layer, the hidden representations from different time scales are learnt at each step, and in the second layer the model learns the temporal dependency relationship in the sequences. Through the decomposition and reconstruction of Discrete Wavelet Transform, the sequences with multiple time resolutions are generated. After obtaining the multi-scale time sequences based on DWT, LSTM network is adopted to learn the long and short term dependency within the sequence due to vanishing gradient problem that exists in traditional recurrent neural networks. The main difference between our work and the work in the literature is that we use multi-scale information simultaneously and the attention model is proposed to learn the importance of each scale automatically by integrating such information into LSTM, i.e., the proposed MSLSTM uses the multi-scale time sequences for training and learns a combination of representations from every distinct scale level. The detailed structure of MSLSTM is introduced as follows:

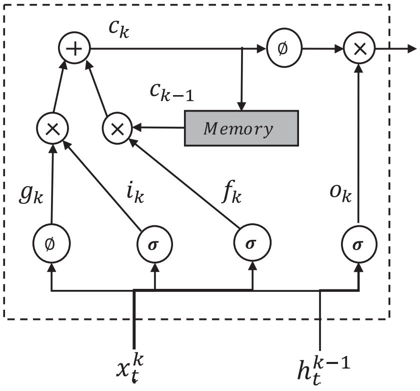


Fig. 4. Internal structure of LSTM cell, where x_t^k is the kth scale level input at time step t, h_t^{k-1} and h_t^k are the hidden states of scale level $k-1$ and k at time step t; ϕ is the $tanh$ function, i_k, f_k and o_k are activations of the input gate, forget gate and output gate respectively. The memory cell c_k, c_{k-1} are states to store internal representation.

The pipeline of learning temporal dependency on multiscale is: given the input data $X = (\mathrm{x}_{1,} \mathrm{x}_{2,} \ldots, \mathrm{x}_{\mathrm{t},} \ldots,), \mathrm{x}_\mathrm{t} \ \in \mathbb{R}^d$, $t \in$ (1,T), multi-scale time series $\{R_1 = (\mathrm{x}_1^1, \mathrm{x}_2^1, \ldots, \mathrm{x}_\mathrm{T}^1), R_2 = (\mathrm{x}_1^2, \mathrm{x}_2^2, \ldots, \mathrm{x}_\mathrm{T}^2), \ldots, R_k = (\mathrm{x}_1^\mathrm{k}, \mathrm{x}_2^\mathrm{k}, \ldots, \mathrm{x}_\mathrm{T}^\mathrm{k}), \ldots, k \in (1, K)\}$ are obtained through the decomposition and reconstruction of DWT with maximum level K, where T is the size of time steps and $\mathrm{x}_\mathrm{t}^\mathrm{k}$ refers to the input of kth level at time t. In the first layer of the MSLSTM, let $(\mathrm{x}_1^\mathrm{k}, \mathrm{x}_2^\mathrm{k}, \ldots, \mathrm{x}_\mathrm{T}^\mathrm{k})$ be the input values of the memory cell and $(\mathrm{h}_\mathrm{t}^1, \mathrm{h}_\mathrm{t}^2, \ldots, \mathrm{h}_\mathrm{t}^\mathrm{k})$ be the list of the generated hidden states, where the activations of the memory cell and the hidden representations can be derived in Fig. 4 as well as the following equations:

$$g_\mathrm{k} = \phi\big(W_{xc}\mathrm{x}_\mathrm{t}^\mathrm{k} + W_{hc}\mathrm{h}_\mathrm{t}^{\mathrm{k}-1} + \mathrm{b_c}\big) \tag{4}$$

$$\mathrm{i_k} = \sigma\big(W_{xi}\mathrm{x}_\mathrm{t}^\mathrm{k} + W_{hi}\mathrm{h}_\mathrm{t}^{\mathrm{k}-1} + W_{ci}\mathrm{c}_{\mathrm{k}-1} + \mathrm{b_i}\big) \tag{5}$$

$$\mathrm{f_k} = \sigma\left(W_{xf}\mathrm{x}_\mathrm{t}^\mathrm{k} + W_{hf}\mathrm{h}_\mathrm{t}^{\mathrm{k}-1} + W_{cf}\mathrm{c}_{\mathrm{k}-1} + \mathrm{b_f}\right) \tag{6}$$

$$\mathrm{c_k} = \mathrm{f_k} \odot \mathrm{c}_{\mathrm{k}-1} + \mathrm{i_k} \odot \mathrm{g_k} \tag{7}$$

$$\mathrm{o_k} = \sigma\big(W_{xo}\mathrm{x}_\mathrm{t}^\mathrm{k} + W_{ho}\mathrm{h}_\mathrm{t}^{\mathrm{k}-1} + W_{co}\mathrm{c}_{\mathrm{k}-1} + \mathrm{b_o}\big) \tag{8}$$

$$\mathrm{h}_\mathrm{t}^\mathrm{k} = \mathrm{ok} \odot \phi(\mathrm{c_k}), \tag{9}$$

W_{xi}, W_{hi}, W_{ci}, W_{xf}, W_{hf}, W_{cf}, W_{xc}, W_{hc}, W_{xo}, W_{ho}, W_{co}, are weight matrix and $\mathrm{b_i}$, $\mathrm{b_f}$, $\mathrm{b_c}$, $\mathrm{b_o}$ are bias vectors, $\sigma\,(\cdot)$ is $sigmoid$ function of hidden layer, ϕ is the $tanh$ function and $\odot$ is the element-wise production. The overview structure of MSLSTM is shown in Fig. 5.

For a given time t, the representations of different time scales are obtained using Eqs. (4)-(9) based on sequence $\{\mathrm{x}_\mathrm{t}^1, \mathrm{x}_\mathrm{t}^2, \ldots, \mathrm{x}_\mathrm{t}^\mathrm{k}\}$, and $\mathrm{h}_\mathrm{t}^\mathrm{k}$ is the hidden representation of kth level at time t. As mentioned in Section 2.3, not all scale levels

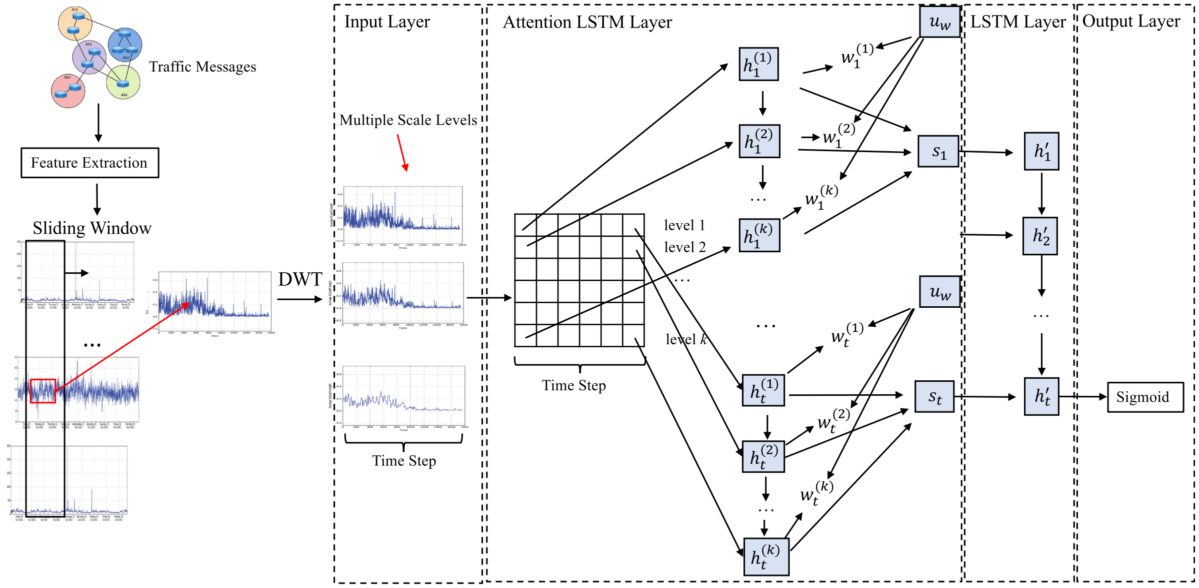


Fig. 5. The framework of proposed Multi-Scale LSTM architecture.

contribute equally, thus the output of each sequence generates an integrated information of different time scales using Eqs. (10)-(12). The attention of each scale level is measured by calculating the similarity of u_t^k and u_w, where the u_t^k is the hidden representation of h_t^k through a fully connected MLP layer, and $\mathrm{u_w}$ is a time scale level context vector to obtain a normalized weight importance $w_t^k.u_w$ is initialized as a random vector and will be jointly learned during the training process.

$$u_t^k = tanh\left(W_w \mathrm{h}_t^k + \mathrm{b_w}\right) \tag{10}$$

$$w_t^k = \frac{\exp(u_t^k u_w)}{\sum_1^K \exp\left(u_t^k u_w\right)} \tag{11}$$

The integrated state S_t is computed as follows:

$$\mathrm{s_t} = \sum_{k=1}^{K} w_t^k \mathrm{h}_t^k. \tag{12}$$

S_t can be regarded as the high level information integrating all time scales at time t, and $(\mathrm{s_1}, \mathrm{s_2}, \mathrm{s_t}, ..)$ will be the input of the second layer LSTM cells to capture the temporal dependency relationship with the following equations:

$$\begin{bmatrix} \mathrm{i_t}\\ \mathrm{f_t}\\ \mathrm{o_t}\\ \mathrm{g_t} \end{bmatrix} \begin{bmatrix} \sigma\\ \sigma\\ \sigma\\ tanh \end{bmatrix} W \cdot \begin{bmatrix} \mathrm{h'}_{\mathrm{t}-1}\\ \mathrm{s_t} \end{bmatrix} \tag{13}$$

$$\mathrm{c_t} = \mathrm{f_t} \odot \mathrm{c_{t-1}} + \mathrm{i_t} \odot \mathrm{g_t} \tag{14}$$

$$\mathrm{h}'_\mathrm{t} = \mathrm{o_t} \odot tanh(\mathrm{c_t}). \tag{15}$$

The calculation of Eq. (13) is the same as Eqs. (4), (5), (6), (7), and (8). W is the parameter vector of the second layer. For the final output class of the sequence $X = (\mathrm{x_1}, \mathrm{x_2}, \ldots, \mathrm{x_t}, \ldots,)$ a $softmax$ function is adopted to predict its label. The $loss$ function that we use is the sigmoid cross entropy as follows:

$$loss(\hat{y}, y) = \frac{1}{|L|} \sum_{l=1}^{l=|L|} -(y_l \log(\hat{y}_l) + (1 - y_l) \log(1 - \hat{y}_l)), \tag{16}$$

$|L|$ is the number of classes, and y is the true label while y is the prediction label.

We compare our proposed MSLSTM model with one layer LSTM in our previous work and the stacked LSTM architecture in Fig. 6. Figs. 6a and 6c are one layer LSTMs, where Fig. 6a is the simplest architecture and Fig. 6c calculates the temporal attention; Fig. 6b is the stacked LSTM, which only duplicates the LSTM cells in the upper level to obtain multi-scale information, while in Fig. 6d, our model learns an integrated representation of multiple scales at each time step by calculating scale attention.

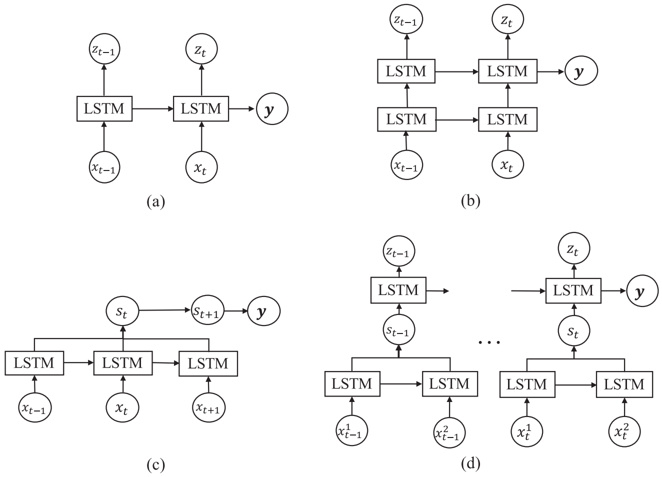


Fig. 6. The comparisons of four LSTM architectures: (a) one layer LSTM; (b) stacked LSTM; (c) one-layer attention based LSTM; (d) our proposed hierarchy attention based LSTM.

4 Experiments and Discussions

To evaluate the feasibility of the proposed model, extensive and comprehensive experiments have been conducted on traffic data from real world applications. In this section, we first introduce the experimental data sets briefly. We then introduce the baseline algorithms to be compared with our method. At the end, the evaluation results as well as discussions are presented.

TABLE 1
Description of BGP Data Sets

Event	Total	Anomaly	Time	AS Number
AS Leak	4560	2280	2004.12.22-2004.12.26	1853,12793,13237
Code Red I	8123	1200	2001.7.17-2001.7.20	513
Nimda	14160	7080	2003.9.16-2003.9.20	513
Slammer	8061	1920	2003.1.23-2003.1.27	513

TABLE 2
Sample of a BGP Update Packet

Field	Value
TIMESTAMP	*1128124817(2004-12-24 08:00:17)*
LEHGTH	*85*
TYPE	*UPDATE*
PEER AS	*5511*
LOCAL AS	*12654*
PEER IP	*195.66.224.83*
LOCAL IP	*195.66.225.241*
ORIGIN	*0(IGP)*
AS PATH	*5511 1239 701 702 4637 4755 9829*
NLOGREGI	61.0.192.0/18 61.0.64.0/18
NEXT_HOP	*195.66.224.83*

4.1 Description of Data Sets

We collect traffic of four BGP security events from Europe RIPE [15] to train our classification model.[1] BGP protocol in general generates four kinds of message: open, update, keep alive, and notification. In this work, the experiment is conducted based on BGP update messages, which is either exchanging announcement or forwarding withdrawal message such as Network Layer Reachability Information (NLRI). For each event, there are several hours of abnormal records. The description of the collected BGP data sets is listed in Table 1.

The BGP update packets that we collect from RIPE are originated from AS 1853, 12793, 13237 (rrc05, Vienna) and AS 513 (rrc04, Geneva). We extract dynamic traffic from network data flow and convert the original Multi-Routing-Table (MRT) format to ASCII. A sample of BGP update message with ASCII format is shown in Table 2. Subsequently, the statistics information of update traffic within half-minute interval is extracted to obtain raw features, which is listed in Table 3 [4]. We calculate the number of announcements, withdrawals, Internal Gateway Protocol (IGP) and External Gateway Protocol (EGP) packets, and incomplete gateway packets in each time slot. The statistics information of AS Path attributes in BGP packets, such as max, unique and average number of AS peers are also extracted. In addition, the value of edit distance among the AS paths is calculated as the least number of operations needed to turn one AS path list to another, where operations include insertions, deletions, and substitutions. At last, we obtain thirty-five features from network traffic in total for the training and testing of our classification task. As shown in Fig. 7, for each anomaly event, we calculate the number of announcements as one feature and transfer the original MRT format into a collection of instance values, where burstiness demonstrate the anomalies.

TABLE 3
Extracted Features from BGP Update Message

Index	Features	Definitions
1	NumAnnounce	*Number of announcements in each time slot*
2	NumWithd	*Number ofwithdrawls*
3	AnnouncePrefix	*Number of announced NLRI prefix*
4	WithdPrefix	*Number of withdrawn NLRI prefix*
5	AvgASPL	*Average AS-PATH length*
6	MaxASPL	*Maximum AS-PATH length*
7	AvgUniqASPL	*Average Unique AS-PATH length*
8	AvgED	*Average edit distance*
9	MaxED	*Maximum edit distance*
10-20	MaxED_n	*Maximum edit distance* $= n$*; where* $n = (7, \ldots, 17)$
21-30	MaxASPL_n	*Maximum AS-PATH length* $= n$*; where* $n = (7, \ldots, 16)$
31	NumIGP	*Number of IGP packets*
32	NumEGP	*Number ofEGP packets*
33	NumIncomp	*Number of incomplete packets*
34	FirstOrderRatio	*Division ratio between the first most active announced prefix and the total number of announcements*
35	ConcentratRatio	*Division ratio between the three most active announced prefix and the total number of announcements*

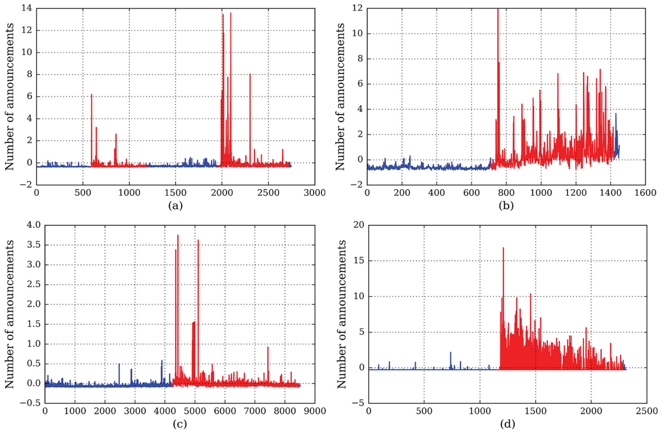


Fig. 7. Number of announcements in AS Leak (a), Code Red I (b), Nimda (c) and Slammer (d). The burstiness in red stands for anomalies.

4.2 Baselines

For comparisons, traditional machine learning algorithms as well as neural network models are selected as the baselines of anomaly classifiaction experiments. The models are listed as follows:

- SVM and NB. Support Vector Machine (SVM) and Naive Bayes (NB) classifiers are well-known supervised learning models in classification tasks [4], [5].
- SVMF and SVMW. SVMF and SVMW are variants of SVM. They use Fisher-score and Wrapper based feature selection methods.
- 1NN. 1NN method uses Euclid distance to calculate the difference, which is a popular baseline in time series classification.

[1] The BGP data sets are available at https://www.ripe.net/analyse/internet-measurements/routing-information-service-ris/ris-raw-data.

- DTW. Dynamic Time Warping (DTW) combined with 1NN [16] is a well-known baseline method in time series classification.
- DT, RF, Ada.Boost. These are three tree based methods, where Random Forest (RF) and Ada.Boost are both the state-of-the-art ensemble models for classification. In Decision Tree (DT), we removed selected features in the constructed trees as indicated in [3].
- MLP. Multi-Layer-Perception (MLP) is a three-layer feed forward model using back propagation.
- RNN and LSTM. RNN and LSTM are both recurrent neural networks with one hidden layer.
- LSTM-2, LSTM-3, LSTM-4, LSTM-5. These models stack multiple layers of LSTM cells as reported in [8] for anomaly classification.
- LSTM-A. LSTM-A is one-layer LSTM network using attention mechanism.

4.3 Training Details and Implementation

Each data set is splited as training, validation, and testing with ratio 60 percent, 10 percent, and 30 percent respectively. For each data set, the number of hidden units is selected from the range $\{2^3, 2^4, \ldots, 2^8\}$. The simulated annealing strategy is adopted to select the optimal learning rate, where the early stop is set to 20 for the validation phase. The number of maximum epoch is set as 100. Batch size is set to 200 and the largest number of scale levels obtained from Discrete Wavelet Transform is set to 10, where we use $haar$ wavelet as the default wavelet type.

All methods are implemented using python, where the traditional machine-learning models are based on scikit-learn package [17] and the neural networks are implemented using tensorflow [18]. For the decomposition and reconstruction of DWT, PyWavelet is employed as the tool package.[2] All implementations can be found on-line.[3]

4.4 Evaluation Metrics

The evaluation metric $Accuracy, Precision, and\, F_{measure}$ are measured for the binary classification, where the positive class is the anomaly and the negative class stands for the normal data record. $Accuracy$ measures the overall performance for both classes, and $F_{measure}$ evaluates the precision and recall of positive class simultaneously. For the imbalanced case, AUC (Area Under Curve) is used as the evaluation metric [19], where the curve is the ROC (Receiver Operating Characteristic) curve which plots false positive rate at decision threshold on the x-axis against true positive rate on the y-axis [20]. AUC characterizes the difference between the predictive distribution of two classes [21]. Intuitively, the more AUC is close to 1, the better the classification is devised. The metrics are calculated according to the following formulations:

$$Accuracy = \frac{TP + TN}{TP + FN + FP + FN} \tag{17}$$

$$Presision = \frac{TP}{TP + FP} \tag{18}$$

TABLE 4
Confusion Matrix

	Predictive Positive	Predictive Negative
True Positive	TP	FN
True Negative	FP	TN

$$Recall = TPR = \frac{TP}{TP + FN} \tag{19}$$

$$F_{measure} = \frac{2Precision * Recall}{Precision + Recall}, \tag{20}$$

where TP, TN, FP, FN refer to elements in confusion matrix of Table 4.

4.5 Binary Classification

To demonstrate the effectiveness of the MSLSTM, the experiment is first conducted in terms of binary classification on four balanced data sets, and the qualitative results are compared with baseline methods. As Table 5 shows, the sequential models (RNN and LSTMs) clearly achieve better performance than traditional algorithms due to the fact that decision is not solely based on present state. Sequential models can reveal the temporal nature of traffic by taking historical information into consideration and make the decision more accurate, which is also consistent with the conclusion in previous work [1]. In traditional methods, the strong classifier SVM, as well as ensemble models such as RF and Ada.Boost, will perform better than weak learners DT and 1NN. Besides, the feature selection methods combined with traditional classifiers are proved to be effective to improve performance as the conclusion in [5]. Compared with vanilla RNN, the LSTM networks obtain better results since the cell structure of LSTM can model long term relationship with more component gates. The proposed MSLSTM and the stacked LSTM achieve better performance than single layer LSTM architecture, which indicates that integrating multiscale traffic and considering additional temporal dependency information have a positive impact on the classification. Furthermore, the attention mechanism is proved to be useful as the comparison between one-layer LSTM and the LSTM-A shows. Among all approaches, the stacked LSTM (LSTM-3) and MSLSTM achieve the best performance, while MSLSTM outperforms baselines in almost all situations and it has increased the evaluation results at a large margin (16 percent at most). This suggests that learning attentions of multiple time scales can provide more reliable temporal information to make more accurate classification. In addition, MSLSTM achieves more robust results in comparison with stacked models and the number of layers affects the performance of the stacked LSTM. A reasonable explanation is that the input to next layer in stacked LSTM is only the previous layer of hidden representation h_t^i, failing to model multiscale characteristics with proper number of layers. While MSLSTM feeds the recombined representation with attention from all scale levels s_t to next layer. In Slammer data set, the anomalies can be identified completely by almost all methods, showing that this type of anomaly is much more distinguishable than others.

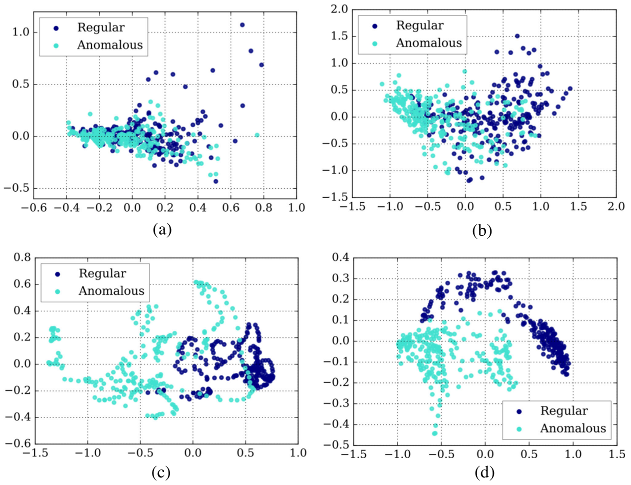
[2] PyWavelets-Discrete Wavelet Transform in Python https://pywavelets.readthedocs.io/en/latest/

[3] Code and data are available at: https://github.com/jayvischeng/MSLSTM

TABLE 5
The Comparison Results of Proposed Model and Baselines

Method	Data Set											
	AS Leak			Code Red I			Nimda			Slammer		
	Accuracy	$F_{measure}$	*Precision*	*Accuracy*	$F_{measure}$	*Precision*	*Accuracy*	$F_{measure}$	*Precision*	*Accuracy*	$F_{measure}$	*Precision*
SVM	62.2	67.2	62.9	57.1	69.5	51.3	61.8	57.7	69.5	77.7	71.3	99.7
NB	62.6	65.5	57.0	64.5	64.2	68.7	56.2	50.5	64.6	74.0	65.9	100
SVMF	69.9	65.9	74.5	53.7	67.9	52.2	51.9	60.9	59.7	59.7	68.2	60.7
SVMW	67.3	66.4	57.8	58.2	70.0	55.0	60.2	57.9	71.2	71.5	60.1	100
1NN	59.4	63.5	58.5	56.3	68.8	54.6	58.4	56.5	60.7	76.0	69.8	95.4
DTW	59.2	67.9	66.1	73.0	70.0	71.2	62.2	64.7	63.3	68.8	75.7	96.1
DT	63.2	65.2	62.1	55.8	68.4	53.1	61.2	58.8	63.0	80.0	75.8	96.0
RF	63.0	68.6	60.4	56.8	69.1	54.0	66.3	64.4	69.2	81.1	76.7	100
Ada. Boost	61.3	66.3	61.2	54.6	68.0	66.6	64.3	64.2	69.0	81.8	78.0	98.7
MLP	60.2	70.2	61.8	69.1	74.5	54.1	60.6	57.6	69.9	80.6	76.0	100
RNN	73.4	74.2	64.9	72.7	77.4	74.5	79.4	80.3	73.5	81.2	76.8	100
LSTM	70.7	73.8	60.5	78.7	81.6	72.0	77.7	78.1	74.8	89.7	88.5	100
LSTM-2	73.7	76.6	63.4	80.0	82.8	72.7	79.2	79.8	79.9	90.5	89.4	100
LSTM-3	76.9	72.7	**80.8**	80.7	82.0	77.0	78.2	78.5	75.3	91.3	90.5	100
LSTM-4	70.6	72.6	68.0	79.6	75.7	72.9	81.1	82.0	78.1	88.5	87.0	100
LSTM-5	76.3	75.4	78.2	82.0	84.5	74.2	79.9	80.8	77.4	88.6	87.2	100
LSTM-A	73.6	74.6	76.6	81.1	82.6	76.8	79.8	80.0	79.3	87.0	85.1	100
MSLSTM	**82.6**	**84.4**	80.3	**89.9**	**89.9**	**89.2**	**83.6**	**83.2**	**85.2**	**93.2**	**92.7**	100
Overall Improvement	7.4%	10.2%	-	9.6%	6.4%	15.8%	3.1%	1.5%	6.6%	2.1%	2.4%	-

To illustrate the quality of the learned features of the proposed model, we project them into 2-D space by using Principle Component Analysis (PCA) to reveal the manifold structure of learned representation. Fig. 8a is the projection of the original features for Slammer data set without using sliding window. Figs. 8b and 8c are the learned hidden features of standard LSTM and 3-layer LSTM respectively; Fig. 8d is the hidden representation of our MSLSTM. In Figs. 8a and 8b, the decision boundary between the regular data and the anomaly is not so distinguishable, in Fig. 8c most anomalous points can be separated from regular ones while in Fig. 8d there exists a clear boundary between the anomalies and regular instances. The observations demonstrate the necessity of considering historical information and the effectiveness of learning representations from multiple scales.



(a)

(b)

(c)

(d)

Fig. 8. The features being projected into 2-D space: (a) original features; (b) hidden representation learned by LSTM; (c) hidden representation learned by 3-Layer LSTM; (d) hidden representation learned by MSLSTM.

In order to evaluate the effects of parameters on time window and scale, various sliding window sizes as well as different time scales from Discrete Wavelet Transform are utilized to compare the impact of considering multi-scale temporal information. Fig. 9 illustrates the *Accuracy* of the proposed model using different values of scale and sliding window size on the data set. The wavelet filter is *Daubechies* filter and the wave length is 2. A larger value of sliding window size will lead to more abundant information, resulting in clear asending trend of *Accuracy*. Besides, increasing the number of scale levels will improve the performance significantly since more diversity and higher degree of temporal granularity are integrated into the model.

In Fig. 10, the accuracy as well as loss on the training and validation set are reported. The standard one-layer LSTM, the stacked LSTM, and our proposed model are selected for comparison. It is clear that the MSLSTM and the stacked

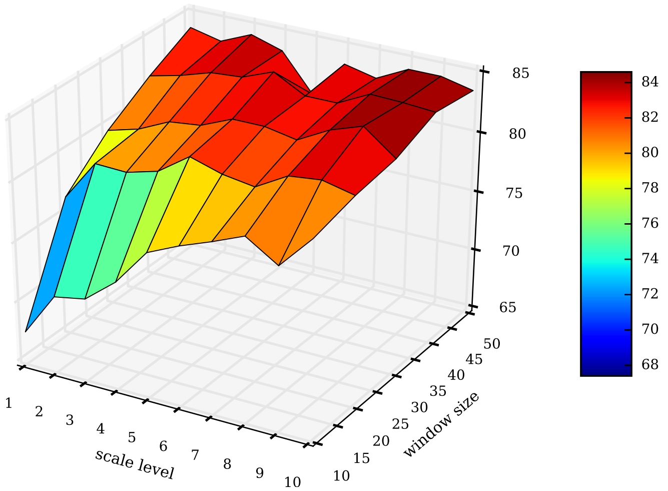
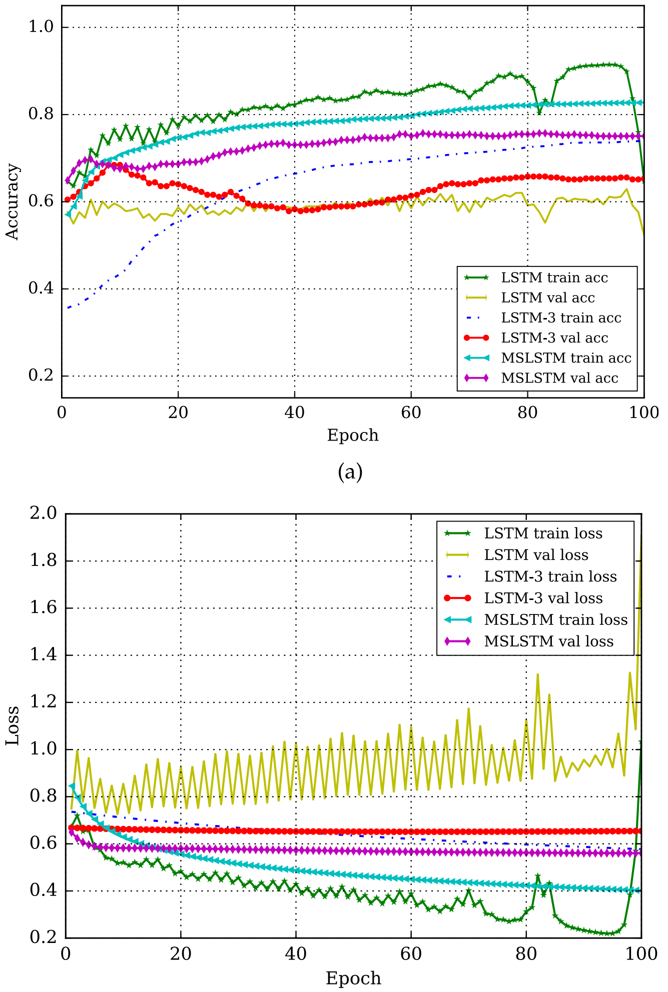


Fig. 9. Accuracy of different sliding window size and preserved time scale.



(a)

(b)

Fig. 10. The accuracy and loss of different LSTM architectures on training and validation data set.

LSTM perform better than one-layer LSTM on validation set. Besides, the results of single layer LSTM fluctuate a lot on both sets and the accuracy drops sharply after 90 epochs, which indicates that single layer LSTM is easy to suffer from over-fitting. In contrast, MSLSTM obtains more robust performance since both of the training and validation curves are smoother and MSLSTM also achieves the highest accuracy and the lowest validation loss.

Further experiments are conducted to evaluate the impact of selecting different combinations of anomaly events as training and testing data set, where the target is to evaluate the ability of predicting unknown type of anomaly based on existing types. In real environment, different types of anomaly could share common pattern to certain degree in spite of diverse behaviors exhibited by various kinds of anomaly events. The four anomaly events are symbolized as 1, 2, 3, 4, where the combinations and results of the four events are described in Table 6. The results demonstrate that our proposed model also outperforms the standard LSTM as well as the stacked LSTM in most cases in terms of identifying unknown type of anomaly.

TABLE 6
The Results of Different Combinations of Four Data Sets

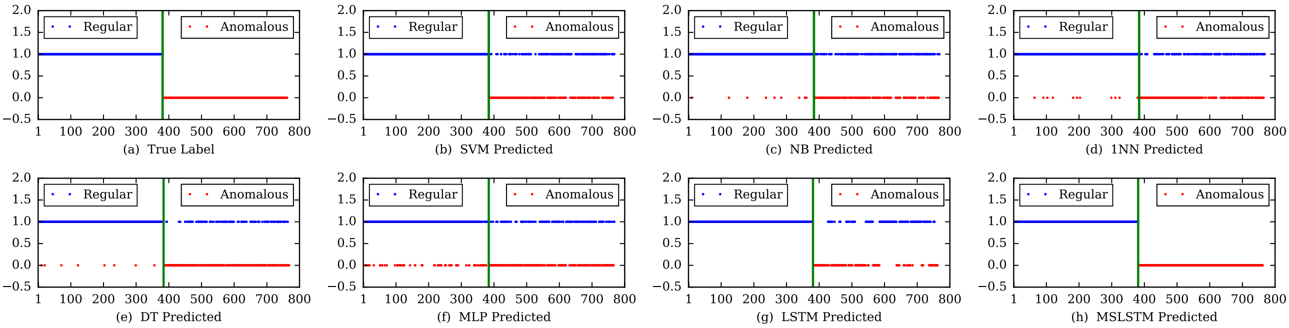
Data Train && Validation Set	Test Set	*Accuracy* LSTM	LSTM-3	MSLSTM	$F_{measure}$ LSTM	LSTM-3	MSLSTM
(2,3,4)	1	64.7	68.5	**69.9**	64.1	68.3	**70.5**
(1,3,4)	2	**93.9**	91.8	90.7	**93.8**	92.2	91.5
(1,2,4)	3	70.8	74.0	**81.0**	75.8	77.8	**81.1**
(1,2,3)	4	97.7	89.5	**99.3**	97.7	88.3	**99.3**

In addition to evaluating the impact of preserving different number of scale levels, the performance of the proposed model adopting various types of wavelet filter functions and filter lengths is also measured. As illustrated in Table 7, three kinds of filters are used with various filter lengths for the comparison. H stands for H_{aar} filter, which is a popular type of filter in signal traffic processing and the length is 2. D_1 and D_2 are both $Daubechies$ filters while the size of filter length is 2 and 4. C_1 is the $Coiflets$ filter whose length is 6. The number of selecting scale levels is 2, 3 and 4 respectively and we use the average results of four data sets. The results show that the length of wavelet filter affects the performance, and smaller filter length performs better, while different types of wavelet functions with the same filter length obtain the same results, and with the increasing number of scales, $Accuracy$ and $F_{measure}$ will be improved.

Furthermore, we conduct experiments to compare the false alarm rate as well as missing alarm rate of different classification models. False alarm means predicting the regular traffic to anomalies incorrectly, while missing alarm predicts the anomaly to regular. Both rates are the concerns of network security. We use testing traffic of Code Red I data to compare the predicted results with true labels. Fig. 11 contains eight sub-figures: (a) plots true labels; (b) plots predicted labels of SVM model with the best result from feature selection; (c) plots predicted labels of NB classifier; (d) plots predicted results of 1NN classifier; (e) plots predicted labels of decision tree; (f) plots MLP; (g) plots standard LSTM and (h) plots MSLSTM respectively. The maximum epochs of MLP, LSTM and MSLSTM are set as 100, which means all the data instances run on the models for 100 times. As shown in Fig. 11a, the left part of the vertical green line refers to regular traffic sequences where the label value is 1.0, and the right portion are the anomalies where the true label is 0. From the comparisons of (b) to (h) in the figure, our MSLSTM model has no false alarm and missing alarm, which is a competitive improvement compared with the rest of the approaches. The LSTM model also achieves lower missing alarm rate than the rest,

TABLE 7
Accuracy and F-Measure of Different Wavelet Filters

Scale Level	Filter Type	*Accuracy*	$F_{measure}$
2	D_1	73.6	74.3
	D_2	68.2	68.4
	H	73.6	74.3
	C_1	66.3	65.6
3	D_1	77.6	77.6
	D_2	70.3	70.1
	H	77.6	77.6
	C_1	73.6	74.4
4	D_1	76.8	76.8
	D_2	71.5	71.5



(a) True Label

(b) SVM Predicted

(c) NB Predicted

(d) 1NN Predicted

(e) DT Predicted

(f) MLP Predicted

(g) LSTM Predicted

(h) MSLSTM Predicted

Fig. 11. The comparisons of true labels and predictions of the testing traffic.

which means taking the long term temporal dependency into consideration will improve the ability of identifying the anomalous traffic. The SVM and two sequential models achieve desirable performance on classifying regular traffic; while MLP, 1NN and DT perform poorly on both the identification of regular traffic and anomalies. By investigating the experimental results on binary classification, MSLSTM has achieved a significant improvement compared with the traditional approaches as well as the stacked LSTM network.

4.6 Multi-Class Classification

We also extend our model for multi-class classification and design experiments to evaluate the performance of MSLSTM. In terms of multi-class case, there are five classes in total, including regular traffic and four different types of anomalies. The class distribution is balanced by selecting a portion of each anomaly due to the various number of anomalies in different BGP anomaly events. Finally, 6000 samples (1200 for each class) are adopted for multi-class classification and the training, validation as well as testing ratios are split as 60 percent, 10 percent, and 30 percent respectively. The accuracy of the sequential models and MSLSTM is tested for comparison. As shown in Table 8, the result of RNN is close to LSTM, which are both one-layer architecture. Compared with one-layer model, MSLSTM and the stacked LSTM improve the accuracy significantly, demonstrating that the effectiveness of considering temporal dependency on multiple scales. With the number of layers increasing, the performance of the stacked LSTM also improves, but more number of hidden layers means larger parameter space and needs more data to train the model, thus the accuracy will drop after obtaining an optimal result at 3 layers. Compared with LSTM-3, MSLSTM achieves better performance with only two hidden layers. In Fig. 12 the confusion matrix yielded by the proposed model is illustrated. It shows clearly that MSLSTM can devide the regular and most types of anomaly into the correct classes.

4.7 Imbalanced Classification

Except for conducting experiments in terms of balanced data distribution, the classification performance on imbalanced situation is also evaluated. The AUC results as well as the ROC curves of two imbalanced data set, Code Red I and Slammer, are illustrated in Fig. 13. The imbalance ratio of the two data sets are 5.8 and 3.2 respectively. For the comparisons, two traditional models SVM and NB, two ensemble approaches RF and Ada.Boost as well as stacked LSTM are adopted to calculate the AUC values. The experimental results demonstrate that ensemble methods outperform the single classifiers and two sequential models perform better than the rest approaches in imbalanced situation. Besides, the proposed MSLSTM still achieves the best result compared with all the state-of-the-art methods.

TABLE 8
The Result of Multi-Class Classification on BGP Traffic

Method	RNN	LSTM	LSTM-2	LSTM-3	LSTM-4	LSTM-5	MSLSTM
Accuracy	61.3	63.7	75.7	77.5	73.2	76.1	77.9

4.8 Complexity Analysis

The complexity measurement of MSLSTM in comparison with the stacked LSTM is also addressed. As mentioned above, there is an input layer, a recurrent LSTM layer, and an output layer in the standard LSTM architecture. The input layer is fully connected to the LSTM layer, which has three gates and the outputs of cell units are fully connected to the output layer. Assuming the number of hidden cells of a LSTM layer is n_c, the number of parameters associated with one layer LSTM cell can be calculated as $4 * n_c\{n_i + n_c\}$, where n_i is the size of inputs (excluding a bias term). Let N be the total number of parameters, then $N = 4 * n_c(n_i + n_c) + n_c * 3 + n_c * n_o$ (n_o is the number of output units). In MSLSTM, the number of hidden units of the first layer is equal to the number of scales n_s where $n_s < n_c$. Let $\hat{N} = 4 * n_s(n_i + n_s)$ represent the number of parameters for the first layer of MSLSTM. The computational complexity of learning LSTM models per

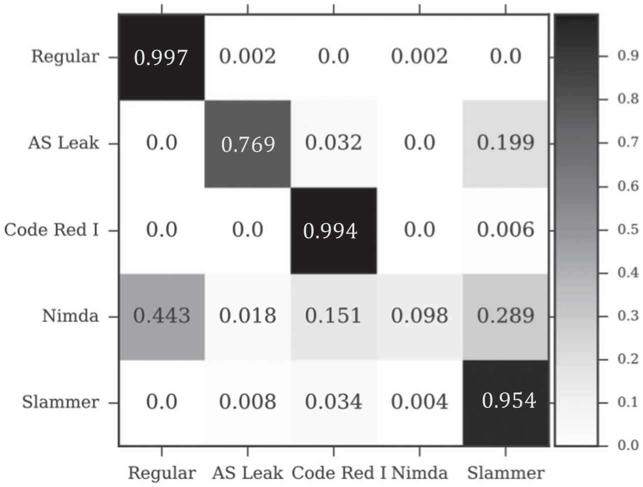
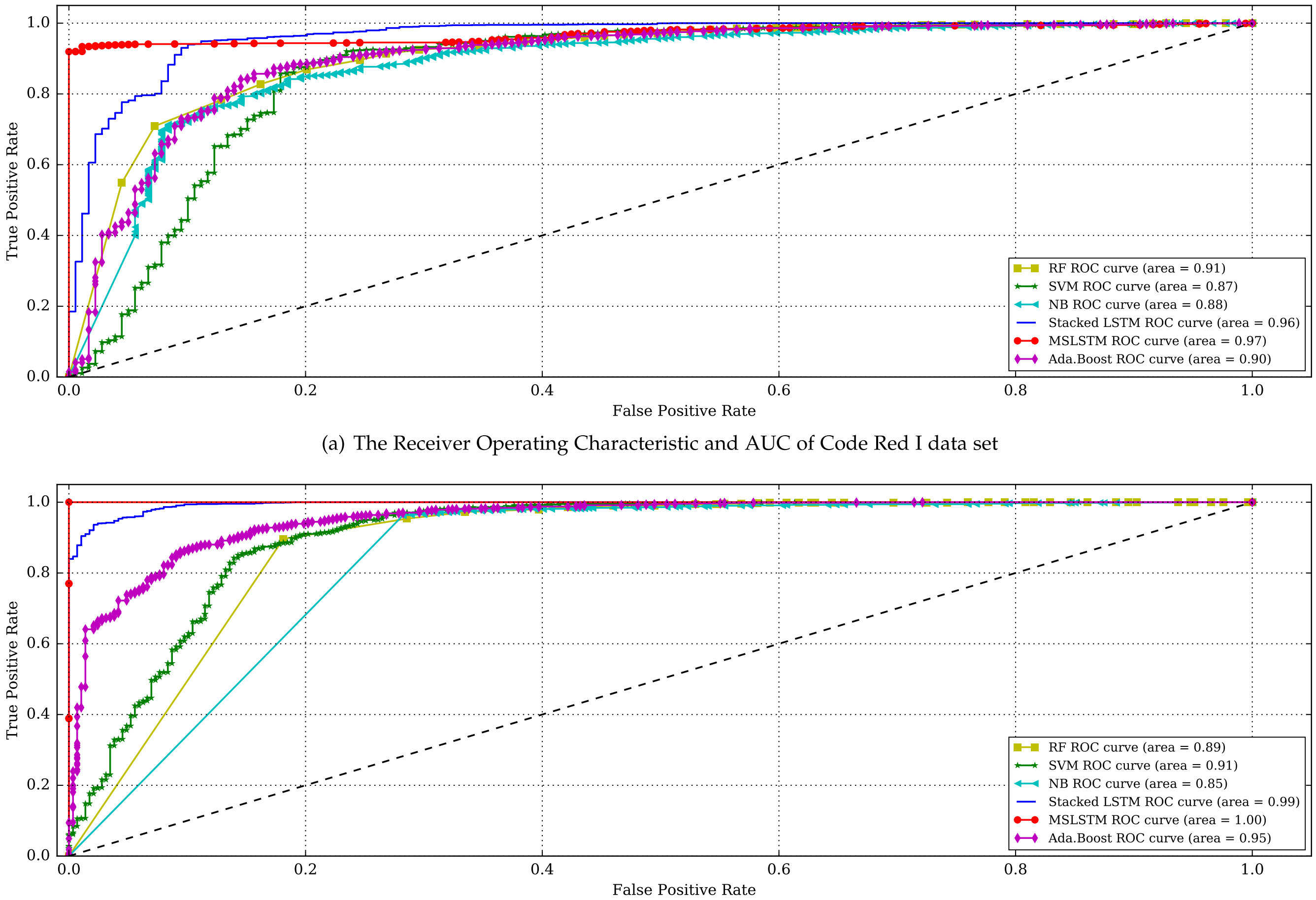


Fig. 12. The confusion matrix yield by the proposed model.



(a) The Receiver Operating Characteristic and AUC of Code Red I data set

(b) The Receiver Operating Characteristic and AUC of Slammer data set

Fig. 13. The performance comparisons of different methods on imbalanced classification.

weight and time step with the Stochastic Gradient Descent (SGD) optimization technique is $O(1)$, which means the learning computational complexity per time step is $O(N)$, and the total learning time for a network is dominated by the number of hidden units as well as the number of LSTM layers. Thus, if the number of hidden layers is M, the total cost for stacked LSTM is $O(MN)\,(M > 2)$. For our proposed model, the architecture only includes two layers and the cost for Discrete Wavelet Transform can be preprocessed off-line before the model training, resulting the cost for MSLSTM is $O(N + \hat{N})$.

As mentioned above that $n_s < n_c, \hat{N} < N$ and $M > 2$, thus $O(N + \hat{N}) < O(MN)$. In Table 9, we compare the testing accuracy as well as elapsed time for classification process of our model and stacked LSTM on an NVIDIA GeForce GTX 1080 GPU. For stacked models, the LSTM-2 reaches the best accuracy 87 percent when the number of hidden units is 128, the training time is 3.82 minutes, and testing time is 0.32 second, while MSLSTM achieves 91.4 percent with 2.42 minutes for training and 0.22 for testing. When the number of hidden units is 256, LSTM-3 reaches 91.3 percent with the elapsed training and testing time being 15.78 minutes and 1.24 seconds respectively. However, MSLSTM achieves optimal value 93.2 percent at a cost of 8.32 and 0.89 with less parameters. Therefore, MSLSTM can reduce time consumption when obtaining better performance in comparison with stacked LSTM.

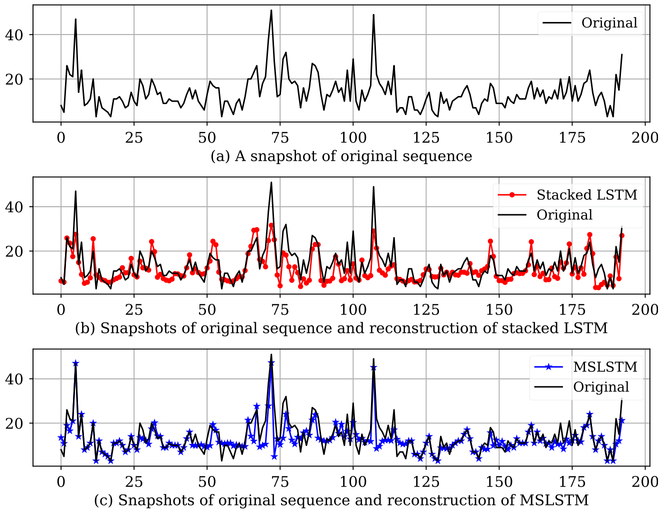
4.9 Visualization

To provide a better insight of the difference between stacked LSTM model and our proposed MSLSTM, we also design an experiment to compare the reconstruction results of stacked model and our model. By replacing the output label to be

TABLE 9
Experiments with Stacked LSTM and MSLSTM Architectures Showing Test Set Accuracies on BGP Data Set. C Indicates the Number of Memory Cells

Model	C	DWT	Depth	N	Test (%)	Training Time(min)	Testing Time(sec)
Stacked LSTM	128		2	165K	87.0	3.82	0.32
	128		3	247K	86.0	5.33	0.43
	128		4	330K	84.6	6.95	0.55
	128		5	412K	86.1	8.59	0.67
	256		2	593K	90.5	10.75	0.84
	256		3	889K	91.3	15.78	1.24
	256		4	1.2M	88.5	21.20	1.65
	256		5	1.5M	88.6	26.56	2.07
MSLSTM	128	2	2	83K	91.4	2.42	0.22
	256	4	2	297K	93.2	8.32	0.89

DWT indicates the number of decomposition level based on discrete wavelet transform. depth is the number of LSTM network layers. N is the total number of parameters. test is the accuracy of the test data and time stands for the elapsed time of classification process.



(a) A snapshot of original sequence

(b) Snapshots of original sequence and reconstruction of stacked LSTM

(c) Snapshots of original sequence and reconstruction of MSLSTM

Fig. 14. The reconstructions of original sequence using stacked LSTM and MSLSTM.

original sequence and using a square loss as the objective function, the reconstruction of the input by MSLSTM and the stacked LSTM can reveal the effectiveness of learned representation. As Fig. 14 shows, the proposed and the stacked model can learn a global evolution of traffic data, but compared with MSLSTM that captures all the necessary detailed burstiness $(x = 5, 71 \text{ and } 105)$, the stacked LSTM fails to preserve burstiness at those positions while only recon-structing the overview of the original signal. The observation demonstrates that the stacked LSTM is much easier to underfit the data than our model since it only generates higher level of temporal abstraction from original signal and fails to consider multi-scale information simultaneously. While in MSLSTM, the extracted abundant information is fully characterized to learn both the local features as well as global trend.

5 Literature Review

Recent research works of anomaly identification are mainly based on visualization-based tools [22], [23] and machine learning algorithms [3], [4], [5]. Specifically, many machine learning approaches have been employed to build traffic classification models and detect anomalies, including both unsupervised and supervised methods. Ahmed et al. propose recursive kernel based on-line anomaly detection method [24] to distinguish irregular network dynamics. Dainotti et al. [25] identify anomalies with non-stationary traffic in network. The Naive Bayes (NB) estimators are utilized to categorize the traffic flows in [4], [26]. Al-Rousan et al. [5] adopt feature selection methods to select the most descriptive features to characterize the BGP activity and develop Naive Bayes classifiers to identify anomalous traffic. Besides, they also employ Support Vector Machine (SVM) and Hidden Markov Model (HMM) to classify BGP anomalies [4], and achieve best results by SVM combined with feature selection models. Nguyen et al. [27] propose one-class neighbor machine learning algorithm to distinguish irregular network traffic. However, these models only treat the input instances independently and fail to take the temporal attributes of traffic data into consideration. In practice, the traffic data can be considered as a stream of multivariate time sequences and the anomalous patterns vary gradually with the temporal information [1].

More recently, deep learning models have an increasing impact on classification tasks due to the effective capability for representation learning. The sequential models in deep learning, including Recurrent Neural Network (RNN) and Long Short-term Memory (LSTM) model, have achieved desirable performance in many sequence classification scenarios [8], [28], [29], [30], [31]. Unlike traditional RNNs that suffer the problem of vanishing gradient during the back-propagation phase, LSTM [7] can capture long term dependency relationship in sequences by a connected memory cell and three gates, i.e., the input gate, the forget gate and the output gate [32]. More recently, attention-based recurrent networks have been successfully applied to a wide variety of tasks, such as natural language question answering [33], [34], parsing [35] and image caption generation [36]. In sequence learning, the attention is first proposed by Bahdanau et al. [10] to select the reference words in the source sentences in neural machine translation. The attention mechanism outputs a weighted summarized representation instead of only considering the representation of the final step, resulting a more representative and expressive feature [37]. The weight assigned to each learned value is calculated by a compatibility function of the relevance importance among the corresponding inputs.

Many variants of LSTM have been proposed to exploit dependency information in sequences. Yang et al. propose an attention based model for document classification [38]. Lipton et al. propose LSTM to recognize patterns in multivariate time series of clinical measurements [39]. Cheng et al. develop a single layer LSTM model integrating time scale to identify anomalous traffic [1], where the value of time scale is a hyperparameter. Ding et al. adopt LSTM model with one hidden layer to classify BGP anomalies [6]. In these models, the temporal information on long and short term dependency are exploited, but they fail to take the multi-scale characteristic into consideration. Malhottra et al. propose a stacked LSTM network to learn higher-level temporal patterns without prior knowledge of the pattern duration [8]. Cheng et al. approach time-series anomaly detection by proposing a soft-max classifier based on stacked LSTM network [9]. Chauhan et al. utilize a deep recurrent neural network architecture with LSTM units to develop a predictive model for healthy ECG signals [40]. However, in those works, the information of different scales is generated by the additional hidden layers that recombine the learned representations from prior layers [41]. Besides, the stacked LSTM architecture only considers abstraction on time-domain to obtain higher level of temporal information, without considering the effect of each distinct scale during learning the dependency, and it is tricky to decide the depth of the stacking model to generate optimal representations. Meanwhile, the difficulty in determining the temporal scale of traffic patterns needs techniques to separate the low frequency pattern from the high frequency through the process of translation (shifting) and dilation (scaling) [42] Wavelet transform can reveal information from multiresolution aspects on both the time and frequency domain [43]. In previous works of combing wavelet transform and sequence models, the conducted evaluations have produced an improved performance [42],

[44], but the combination is preliminary and the wavelet transform algorithm is only used to reduce the length of sequence. Besides, the multiple decompositions are not considered simultaneously into the RNN or LSTM model since the result of wavelet decomposition at each level is independent from each other during the sequential model training.

6 Conclusion

Border Gateway Protocol (BGP) provides a fundamental environment to exchange routing information for autonomous systems in the Internet and the identification of anomalous BGP traffic is an essential task to maintain successful collaborations of relevant services across different domains. The time scale is significant when utilizing temporal information to design classification model. In this paper, we propose a novel multi-scale LSTM model (MSLSTM), which uses discrete wavelet transform to generate abundant information of different scales explicitly from time sequences. Subsequently, the information of multiple scales is integrated into two-layer hierarchical attention LSTM model, where in the first layer, attentions of different time scales are learnt to obtain a recombined representation of multiple scales. In the second layer, the LSTM cells are devised to exploit the temporal dependency among the input representation from the prior layer. By evaluating the model based on several typical real-world BGP anomaly events as well as making comparisons with other baseline approaches, our proposed MSLSTM can achieve better classification accuracy as well as lower false alarm rate, and higher AUC results with only two-layer architecture and less time consumption. The experimental results verify the effectiveness and efficiency of MSLSTM to classify anomalous behaviors in BGP traffic.

Acknowledgments

The work described in this paper has been supported by the following research fundings, including a General Research Fund (project no. CityU 11211417) from the Research Grants Council of the Hong Kong SAR, an internal research fund from City University of Hong Kong (project no. 9610367), the grants from NSFC (No. U1611461), Science and Technology Program of Guangdong Province, China (No. 2016A010101012), and CAS Key Lab of Network Data Science and Technology, China (No. CASNDST201703).

References

[1] M. Cheng, Q. Xu, J. Lv, W. Liu, Q. Li, and J. Wang, "MS-LSTM: A multi-scale LSTM model for BGP anomaly detection," in *Proc. IEEE Int. Conf. Network Protocol*, 2016, pp. 1–6.

[2] A. Pecchia, S. Russo, and S. Sarkar, "Assessing invariant mining techniques for cloud-based utility computing systems," *IEEE Trans. Services Comput.*, vol. PP, p. 1, Mar. 2017.

[3] Y. Li, H.-J. Xing, Q. Hua, X.-Z. Wang, P. Batta, S. Haeri, and L. Trajkovic, "Classification of BGP anomalies using decision trees and fuzzy rough sets," in *Proc. IEEE Int. Conf. Syst. Man Cybern.*, 2014, pp. 1312–1317.

[4] N. M. Al-Rousan and L. Trajkovic, "Machine learning models for classification of BGP anomalies," in *Proc. IEEE Int. Conf. High Perform. Switching Routing*, 2012, pp. 103–108.

[5] N. Al-Rousan, S. Haeri, and L. Trajkovic, "Feature selection for classification of BGP anomalies using bayesian models," in *Proc. IEEE Int. Conf. Mach. Learn. Cybern.*, 2012, pp. 140–147.

[6] Q. Ding, Z. Li, P. Batta, and L. Trajkovic, "Detecting BGP anomalies using machine learning techniques," in *Proc. Int. Conf. Syst. Man Cybern.*, 2016, pp. 352–355.

[7] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[8] P. Malhotra, L. Vig, G. Shroff, and P. Agarwal, "Long short term memory networks for anomaly detection in time series," in *Proc. Eur. Symp. Artif. Neural Netw. Comput. Intell. Mach. Learn.*, 2015, p. 89.

[9] C. Feng, T. Li, and D. Chana, "Multi-level anomaly detection in industrial control systems via package signatures and lstm networks," in *Proc. IEEE/IFIP Int. Conf. Dependable Syst. Netw.*, 2017, pp. 261–272.

[10] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[11] J. Yang, M. N. Nguyen, P. P. San, X. Li, and S. Krishnaswamy, "Deep convolutional neural networks on multichannel time series for human activity recognition," in *Proc. Int. Joint Conf. Artif. Intell.*, 2015, pp. 3995–4001.

[12] L. Zhang, J. Lin, and R. Karim, "Sliding window-based fault detection from high-dimensional data streams," *IEEE Trans. Syst. Man Cybern.: Syst.*, vol. 47, no. 2, pp. 289–303, 2017.

[13] J. Li, M. D. Levine, X. An, X. Xu, and H. He, "Visual saliency based on scale-space analysis in the frequency domain," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 4, pp. 996–1010, Apr. 2013.

[14] T. Moore and M. Zirnsak, "Neural mechanisms of selective visual attention," *Annu. Rev. Psychology*, vol. 68, pp. 47–72, 2017.

[15] Ripe RIS raw data. [Online]. Available: http://www.ripe.net/data-tools/stats/ris/ris-raw-data.

[16] A. Mueen and E. Keogh, "Extracting optimal performance from dynamic time warping," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2016, pp. 2129–2130.

[17] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011.

[18] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, et al., "Tensorflow: A system for large-scale machine learning," in *OSDI*, vol. 16, pp. 265–283, 2016.

[19] A. Rosa, L. Y. Chen, and W. Binder, "Failure analysis and prediction for big-data systems," *IEEE Trans. Serv. Comput.*, vol. 10, no. 6, pp. 984–998, 2017.

[20] P. A. Flach, J. Hernandez-Orallo, and C. F. Ramirez, "A coherent interpretation of AUC as a measure of aggregated classification performance," in *Proc. 28th Int. Conf. Int. Conf. Mach. Learn.*, 2011, pp. 657–664.

[21] Z. Gong and H. Chen, "Model-based oversampling for imbalanced sequence classification," in *Proc. 25th ACM Int. Conf. Inf. Knowl. Manage.*, 2016, pp. 1009–1018.

[22] S. Papadopoulos, G. Theodoridis, and D. Tzovaras, "BGPfuse: Using visual feature fusion for the detection and attribution of BGP anomalies," in *Proc. 10th Workshop Vis. Cyber Secur.*, 2013, pp. 57–64.

[23] M. Steiger, J. Bernard, S. Mittelstadt, H. Lucke-Tieke, D. Keim, T. May, and J. Kohlhammer, "Visual analysis of time-series similarities for anomaly detection in sensor networks," *Comput. Graph. Forum*, vol. 33, pp. 401–410, 2014.

[24] T. Ahmed, M. Coates, and A. Lakhina, "Multivariate online anomaly detection using kernel recursive least squares," in *Proc. IEEE Int. Conf. Comput. Commun.*, 2007, pp. 625–633.

[25] A. Dainotti, A. Pescape, and K. C. Claffy, "Issues and future directions in traffic classification," *IEEE Netw.*, vol. 26, no. 1, pp. 35–40, Jan./Feb. 2012.

[26] X. Zhao, G. Wang, and Z. Li, "Unsupervised network anomaly detection based on abnormality weights and subspace clustering," in *Proc. IEEE Int. Conf. Inf. Sci. Technol.*, 2016, pp. 482–486.

[27] X. N. Nguyen, D. T. Nguyen, and L. H. Vu, "POCAD: A novel pay load-based one-class classifier for anomaly detection," in *Proc. IEEE 3rd Nat. Found. Sci. Technol. Develop. Conf. Inf. Comput. Sci.*, 2016, pp. 74–79.

[28] P. Filonov, A. Lavrentyev, and A. Vorontsov, "Multivariate industrial time series with cyber-attack simulation: Fault detection using an lstm-based predictive data model," *arXiv preprint arXiv:1612.06676*, 2016.

[29] Y. Du, W. Wang, and L. Wang, "Hierarchical recurrent neural network for skeleton based action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 1110–1118.

[30] M. S. Ibrahim, S. Muralidharan, Z. Deng, A. Vahdat, and G. Mori, "A hierarchical deep temporal model for group activity recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 1971–1980.

[31] F. J. Ordonez and D. Roggen, “Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition,” *Sensors*, vol. 16, no. 1, 2016, Art. no. 115.

[32] Y. Tang, J. Xu, K. Matsumoto, and C. Ono, “Sequence-to-sequence model with attention for time series classification,” in *Proc. IEEE Int. Conf. Data Mining Workshops*, 2016, pp. 503–510.

[33] S. Sukhbaatar, J. Weston, R. Fergus, et al., “End-to-end memory networks,” in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2015, pp. 2440–2448.

[34] A. Kumar, O. Irsoy, P. Ondruska, M. Iyyer, J. Bradbury, I. Gulrajani, V. Zhong, R. Paulus, and R. Socher, “Ask me anything: Dynamic memory networks for natural language processing,” *Int. Conf. Mach. Learning*, pp. 1378–1387, 2016.

[35] O. Vinyals, E. Kaiser, T. Koo, S. Petrov, I. Sutskever, and G. Hinton, “Grammar as a foreign language,” in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2015, pp. 2773–2781.

[36] K. Xu, J. Ba, R. Kiros, K. Cho, A. C. Courville, R. Salakhutdinov, R. S. Zemel, and Y. Bengio, “Show, attend and tell: Neural image caption generation with visual attention,” in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 77–81.

[37] J. K. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, “Attention-based models for speech recognition,” in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2015, pp. 577–585.

[38] Z. Yang, D. Yang, C. Dyer, X. He, A. J. Smola, and E. H. Hovy, “Hierarchical attention networks for document classification,” in *Proc. HLT-NAACL*, 2016, pp. 1480–1489.

[39] Z. C. Lipton, D. C. Kale, C. Elkan, and R. Wetzell, “Learning to diagnose with lstm recurrent neural networks,” *arXiv preprint arXiv:1511.03677*, 2015.

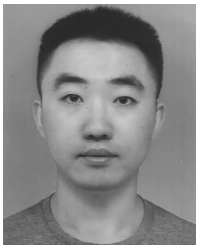
[40] S. Chauhan and L. Vig, “Anomaly detection in ECG time signals via deep long short-term memory networks,” in *Proc. IEEE Int. Conf. Data Sci. Adv. Anal.*, 2015, pp. 1–7.

[41] M. Hermans and B. Schrauwen, “Training and analysing deep recurrent neural networks,” in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2013, pp. 190–198.

[42] P. Sugiartawan, R. Pulungan, and A. K. Sari, “Prediction by a hybrid of wavelet transform and long-short-term-memory neural network,” *Int. J. Adv. Comput. Sci. Appl.*, vol. 8, no. 2, pp. 326–332, 2017.

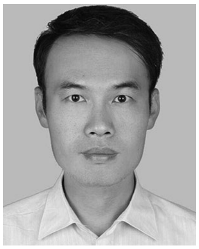
[43] J. Chen, Z. Li, J. Pan, G. Chen, Y. Zi, J. Yuan, B. Chen, and Z. He, “Wavelet transform based on inner product in fault diagnosis of rotating machinery: A review,” *Mech. Syst. Signal Process.*, vol. 70, pp. 1–35, 2016.

[44] L. F. Ortega, “A neuro-wavelet method for the forecasting of financial time series,” in *Proc. World Congr. Eng. Comput. Sci.*, 2012, pp. 24–26.



Min Cheng received the MSc degree from the Chinese Academy of Sciences University, China, in 2014. Currently, he is working toward the PhD degree from the Department of Computer Science, City University of Hong Kong, Hong Kong. His research interests include anomaly detection in network environment, traffic prediction, time series classification with deep learning and nature language processing.

Qing Li received the BEng degree from Hunan University, Changsha, China, and the MSc and PhD degrees from the University of Southern California, Los Angeles, all in computer science. He is currently a professor with the Department of Computer Science, City University of Hong Kong, Kowloon, Hong Kong. His current research interests include dynamic object modeling, multimedia and mobile information retrieval and management, distributed databases and data warehousing/mining, and workflow management and web services.



Jianming Lv received the BS degree in computer science from Sun YAT-SEN University, China, in 2002, and the PhD degree from Institute of Computing Technology, Chinese Academy of Sciences University, in 2008. He is currently an associate professor with the South China University of Technology. His research interests include Data Ming, Computer Vision, Distributed Computing and Privacy. He is a member of the IEEE, ACM and CCF.

Wenyin Liu received the BEng and an MEng degrees in computer science from Tsinghua University, Beijing, and the DSc degree from the Technion, Israel Institute of Technology, Haifa. He is currently a professor in School of Computer Science and Technology and Director of Web Identity Security Laboratory, Guangdong University of Technology. He was Deputy Director of Multimedia Software Engineering Research Centre with the City University of Hong Kong from 2013 to 2016, an assistant professor in the Department of Computer Science with the City University of Hong Kong between from 2002 to 2012, and a full time researcher with Microsoft Research China/Asia from 1999 to 2001. His research interests include anti-phishing, Web identity authentication and management, multimedia information retrieval, text mining, graphics recognition, and performance evaluation. In 2003, he was awarded the International Conference on Document Analysis and Recognition Outstanding Young Researcher Award by the International Association for Pattern Recognition (IAPR). He had been TC10 chair of IAPR for 2006-2010. He had been on the editorial boards of the International Journal of Document Analysis and Recognition (IJDAR) from 2006 to 2011 and the IET Computer Vision journal from 2011-2012. He is a Fellow of IAPR and a senior member of the IEEE.

Jianping Wang received the BS and the MS degrees in computer science from Nankai University, Tianjin, China, in 1996 and 1999, respectively, and the PhD degree in computer science from the University of Texas, Dallas, in 2003. She is an associate professor in the Department of Computer Science with the City University of Hong Kong. Her research interests include dependable networking, optical networks, cloud computing, service oriented networking, and data center networks.