A Network Traffic Flow Prediction with Deep Learning Approach for Large-scale Metropolitan Area Network

Weitao Wang*, Yuebin Bai*, Chao Yu*, Yuhao Gu*, Peng Feng*, Xiaojing Wang*, and Rui Wang*†
*School of Computer Science and Engineering
Beihang University, Beijing 100191, China
†Corresponding author, wangrui@buaa.edu.cn

***Abstract*—Accurate and timely internet traffic information is important for many applications, such as bandwidth allocation, anomaly detection, congestion control and admission control. Over the last few years, internet flow data have been exploding, and we have truly entered the era of big data. Existing traffic flow prediction methods mainly use simple traffic prediction models and are still unsatisfying for many real-world applications. This situation inspires us to rethink the internet traffic flow prediction problem based on deep architecture models with big traffic data. In this paper, we propose a novel deep-learning-based internet traffic flow prediction method, which is called SDAPM. It consider the spatial and temporal correlations inherently and internet flow data character. A stacked denoising autoencoder prediction model (SDA) is used to learn generic internet traffic flow features, and it is trained in a greedy layer-wise fashion. Moreover, experiments demonstrate that the SDAPM for traffic flow prediction has effective performance. Our prediction model is in production as part of the traffic scheduling system at China Unicom, one of the largest Internet companies in China, helping improving the network bandwidth utilization.**

***Keywords*-Deep learning; Stacked denoising autoencoder (SDA); Network traffic prediction; Big data.**

I. Introduction

Internet network traffic prediction is crucial to network providers and computer network management in general [1]. The Internet service providers make use of accurate and timely traffic flow information to allocate bandwidth, detect anomaly, control congestion and design networks. It has the potential to help internet mangers make better schedule decisions, alleviate traffic congestion and improve internet operation efficiency. The objective of internet traffic flow prediction is to provide such internet traffic flow information. Internet Traffic flow prediction has gained more and more attention when large scale and geographically distributed applications are on the rise. Many large service providers, such as China Unicom, Baidu and China Mobile Communications Corporation et al., regarded it as a critical element for advanced traffic management systems, advanced network security analysis system, and commercial scheduling operations. For example, some Internet service providers (ISPs) charge for bandwidth by the peak bandwidth that a customer uses. However, most conventional data centers do not have the infrastructure to support flow-level monitoring and scheduling, and thus rely on an accurate prediction of the future traffic to perform short-term / long-term traffic scheduling.

978-1-5386-3416-5/18/$31.00 © 2018IEEE

Internet traffic flow prediction heavily depends on historical and real-time traffic data collected from various internet flow monitoring sources. With the widespread traditional traffic sensors and new emerging traffic sensor technologies, traffic data are exploding, and we have entered the era of big data internet traffic. Internet traffic management and control driven by big data is becoming a new trend [2], [3]. Although there have been already many internet traffic flow prediction systems and models, most of which use shallow traffic models and are still somewhat unsatisfying. This inspires us to reconsider the internet traffic flow prediction model based on deep architecture models with such rich amount of internet traffic data.

Recently, deep learning [4] has drawn a lot of academic and industrial interest, which is a type of machine learning method and has been applied with success in classification tasks, natural language processing, dimensionality reduction, object detection, motion modeling, and so on [5]–[9]. Deep learning is a simple processing structures, which are separated into strongly connected units called artificial neurons (nodes). Neurons are organized into layers, one layer has multiple neurons and any one neural network can have one or more layers, which are defined by the network topology and vary among different network models [10]. Deep learning algorithms use multiple-layer architectures or deep architectures to extract inherent features in data from the lowest level to the highest level, and they can discover huge amounts of structure in the data. As a internet traffic flow process is complicated in nature, deep learning algorithms can represent internet traffic features without prior knowledge, which has good performance for internet traffic flow prediction.

In this paper, we propose a deep-learning-based internet traffic flow prediction method (SDAPM). Herein, a stacked denoising autoencoder (SDA) model is used to learn generic traffic flow features, and it is trained in a layerwise greedy fashion. According to our knowledge, it is the first time that the SDA approach is used to represent internet traffic flow features for prediction. The spatial and temporal correlations as well as internet flow data character are inherently considered in the modeling. In addition, it demonstrates that the proposed method for internet traffic flow prediction has superior performance.

Internet traffic flow prediction has been long regarded as a key functional component in some internet applications. Over the past few decades, a number of traditional linear models for traffic flow prediction have been developed to assist in internet traffic management and control. The evolution of traffic flow can be considered a temporal and spatial process. The traffic flow prediction problem can be stated as follows. Let X_i^t denote the observed internet traffic flow quantity during the tth time interval at the ith port of node X in a internet network. Given a sequence $\{X_i^t\}$ of observed internet traffic flow data, $i = 1, 2, \cdots, m, t = 1, 2, \cdots, T$, it is interested in predicting the internet traffic flow of node X at time interval $(t + \Delta)$ for some prediction horizon Δ. Therefore, the internet traffic flow prediction problem is precisely defined as following:

Definition *1:* Given a internet node X' ith port internet traffic flow data $\{X_i^t\}$, $i = 1, 2, \cdots, m, t = 1, 2, \cdots, T$, how to predict the internet traffic flow of node X at time interval $(t + \Delta)$ for some prediction horizon Δ and port i?

The rest of this paper is organized as follows. Section II reviews the studies on short-term traffic flow prediction. Section III presents the deep learning approach with SDA as building blocks for internet traffic flow prediction. Section IV the experimental results and analysis are discussed. Concluding remarks are described in Section V.

II. Related works

As early as 1970s, researchers have used Poisson (Poisson) model to describe the characteristics of the traffic data of the Public Switched Telephone Network (PSTN) [11]. And then it is known that internet network traffic has a Short related (Short-range Dependence, SRD) features. In 1990s, the network traffic has a self-similar phenomena is proposed for the first time by Leland et al. [12]. Since then, an extensive variety of models for traffic flow prediction have been proposed by researchers from different areas, such as transportation engineering, statistics, machine learning, control engineering, and economics. Previous prediction approaches can be grouped into three categories, i.e., parametric techniques, nonparametric methods, and simulations. Parametric models include time-series models, Kalman filtering models, etc. Nonparametric models include k-nearest neighbor (k-NN) methods, artificial neural networks (ANNs), etc. Simulation approaches use traffic simulation tools to predict traffic flow.

A widely used technique to the problem of traffic flow prediction is based on time-series methods. In time-series methods, the most widely used model is the autoregressive integrated moving average (ARIMA) model, which was first used to predict short-term traffic flow [13]. Later, Granger et al. proposed a autoregressive fractionally integrated moving average model, called FARIMA, which captures the characters of longmemory time series, is also widely used in traffic prediction [14]. Periyanayagi et al. [15] proposed a time series model called S-ARMA, using Swarm intelligence and ARMA, for the network traffic prediction in wireless sensor networks. Many variants of ARIMA were proposed to improve prediction accuracy, such as spacetime ARIMA [16], and seasonal ARIMA (SARIMA) [17],KohonenARIMA (KARIMA) [18]. Except for the ARIMA-like time-series models, other types of time-series models were also used for traffic flow prediction [19].

Since time-series methods are traditional linear models, which maybe not very suitable for dealing with the stochastic and nonlinear nature of traffic flow, researchers have paid much attention to nonparametric methods in the traffic flow forecasting field. Davis and Nihan used the k-NN method for short-term traffic forecasting and argued that the k-NN method performed comparably with but not better than the linear time-series approach [20]. Chang et al. presented a dynamic multi-interval traffic volume prediction model based on the k-NN nonparametric regression [21]. El Faouzi developed a kernel smoother for the autoregression function to do short-term traffic flow prediction, in which functional estimation techniques were applied [22]. Sun et al. used a local linear regression model for short-term traffic forecasting [23]. A Bayesian network approach was proposed for traffic flow forecasting in [24]. An online learning weighted support vector regression (SVR) was presented in [25] for short-term traffic flow predictions.

To obtain adaptive models, some works explore hybrid methods, in which they combine several techniques. Tan et al. proposed an aggregation approach for traffic flow prediction based on the moving average (MA), exponential smoothing (ES), ARIMA, and neural network (NN) models. The MA, ES, and ARIMA models were used to obtain three relevant time series that were the basis of the NN in the aggregation stage[26]. Zargari et al. developed different linear genetic programming, multilayer perceptron, and fuzzy logic (FL) models for estimating 5-min and 30-min traffic flow rates [27]. In addition to the methods aforementioned, the Kalman filtering method [28], stochastic differential equations [29], the online change-point-based model [30], the type-2 FL approach [31], the variational infinite-mixture model [32], simulations [33], and dynamic traffic assignment [34], were also applied in predicting short-term traffic flow.

Over the last few years, internet flow data have been exploding, and we have truly entered the era of big data. Yisheng Lv et al. [35] proposed a deep learning model called stacked autoencoders (SAEs), using freeway traffic flow data of the Caltrans for the traffic prediction in highway system. In [36], Wenhao Huang et al. propose a deep architecture that consists of a deep belief network (DBN) and a multitask regression layer to forecast traffic flow. The Multilayer Perceptron (MLP) model and Stacked Autoencoder (SAE) proposed by Tiago Prado Oliveira et al. [37] is used to predict internet traffic and they considered that MLP can work even better than SAE. In addition, in [38], the marginalized denoising autoencoder (mDAE) was put forward by Chen, etc., this new algorithm model based on DAE will be marginalized in reconstruction error handling, which can not only reduce reduce the computational overhead but also improve classification/prediction accuracy. In general, literature shows promising results when using Neural Networks (NNs), which have good prediction power and robustness. Although the deep architecture of NNs can learn more powerful models than shallow networks, existing NN-based methods for traffic flow prediction usually only have one hidden layer. It is hard to train a deep-layered hierarchical NN with a gradient-based training algorithm. Recent advances in deep learning have made training the deep architecture feasible since the breakthrough of Hinton et al. [39], and these show that deep learning models have superior or comparable performance with state-of-the-art methods in some areas.

In summary, a large number of prediction algorithms have been developed due to the growing need for realtime flow information in internet networks. However, it is difficult to say that one method is clearly superior over other methods in any situation. One important reason is that the proposed models are developed with a small amount of separate specific internet traffic data, and the accuracy of internet flow prediction methods is dependent on the data features embedded in the collected spatiotemporal internet traffic data.

III. INTERNET FLOW PREDICTION MODEL DESIGN

In this section, we will introduce the SDA model, which is a kind of stack autoencoders, and also is a very popular deep learning model. It is used to building autoencoders of blocks to create a deep network [40]. Firstly, the SDA and introducing necessary notations is described. Afterwards, this section also presents the detailed derivations of the internet flow prediction model. What is more, a few concrete examples are included for illustration. Finally, the internet flow prediction algorithm is designed.

A. *Autoencoder*

Autoencoder is proposed by Rumelhart [41], which is used to learn the compressed and distributed characteristics expression for given the data set. Autoencoder is neural network with three layer structure, which consists of an input layer, one hidden layer and one output layer. Fig.1 shows an illustration of an autoencoder, the scale of input layer must be the same as the hidden layer, but output layer can be any size. Given a set of training samples $S = \{x^{(1)}, x^{(2)}, X^{(3)}, ...\}$, where $x^{(i)} \in R^{(w)}$, for an input $x^{(i)}$, denoted as $\boldsymbol{x}$, the encoding process is that the $x^{(i)}$ is expressed from input layer to the hidden layer, and then the decoding process is that the expressed information is backward expressed from the hidden layer to input layer. Thus, the encoding and decoding process can be described as the following equations:

$$\boldsymbol{h} = f(\boldsymbol{x}) = s_f(\boldsymbol{wx} + \boldsymbol{p}) \quad (1)$$

$$\boldsymbol{y} = g(\boldsymbol{h}) = s_g(\tilde{\boldsymbol{w}}\boldsymbol{h} + \boldsymbol{q}) \quad (2)$$

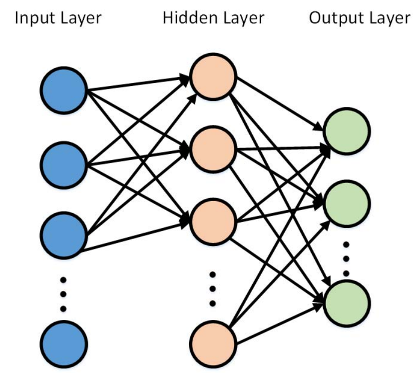


Figure 1: An example of autoencoder.

In equation (1), s_f is called encoded activation function, s_g is called decoding the activation function. In general, encoded activation function and decoding the activation function is replaced by sigmoid function $1/(1 + exp(-x))$ or identity function, and usually let the weight matrix $\tilde{w}$ as w^T. Thus the parameter of an autoencoder θ is equated as $\theta = \{\boldsymbol{w}, \boldsymbol{p}, \boldsymbol{q}\}$.

The output of output layer $\boldsymbol{y}$ can be viewed as the predictive value of the input $\boldsymbol{x}$, and in order to get the exactly θ, Autoencoder let $\boldsymbol{y}$ approach $\boldsymbol{x}$ as soon as possible by minimizing reconstruction error $L(\boldsymbol{x},\boldsymbol{y})$. When s_f is a identity function, the $L(\boldsymbol{x},\boldsymbol{y})$ is as following:

$$L(\boldsymbol{x},\boldsymbol{y}) = \| \boldsymbol{x} - \boldsymbol{y} \|^2 \quad (3)$$

When s_f is a sigmoid function, the $L(\boldsymbol{x},\boldsymbol{y})$ is as following:

$$L(\boldsymbol{x},\boldsymbol{y}) = -\sum_{i=1}^{n}[x_i ln(y_i) + (1 - x_i)ln(1 - y_i)] \quad (4)$$

Then, for given sample training set S, the whole loss function is:

$$J_{AE}(\theta) = \sum_{x \in S} L(\boldsymbol{x}, s_g(s_f(\boldsymbol{x}))) \quad (5)$$

The encoding process, decoding process and minimizing reconstruction error process constitutes a unsupervised learning of Autoencoder, as is shown in Fig.2.

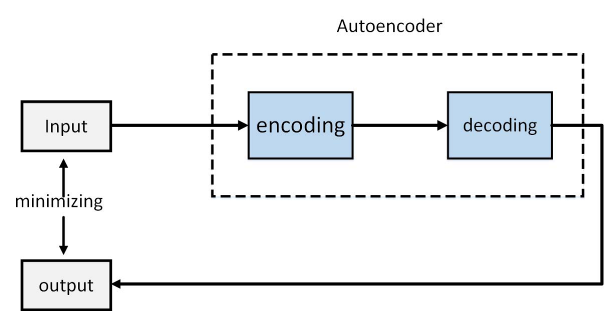


Figure 2: The unsupervised learning process of an autoencoder.

B. Denoising Autoencoder

In order to further improve the generalization ability and robust of autoencoder, some corruption is added to input data. In this paper, for each input data $\boldsymbol{x}$ of given sample training set S, isotropic Gaussian noise ε is added into $\boldsymbol{x}$, that is $\tilde{\boldsymbol{x}} = \boldsymbol{x} + \varepsilon$, $\varepsilon \sim N(0, \mu^2 I)$, where $\mu^2 = \sigma^2/(10^{SNR/10})$, σ^2 is the the variance of S with expectation 0, SNR is given signal to noise ratio. In addition, in the hereinbelow Stacked Denoising Autoencoder(SDA), the output data $\tilde{\boldsymbol{x}}$ of each SDA is added by binary masking noise, that is, each component of the input vector $\tilde{\boldsymbol{x}}$ is 0 with the same probability p, and stays the unchanged with probability $1-p$. The processing of data Denoising Autoencoder is shown in Fig.3.

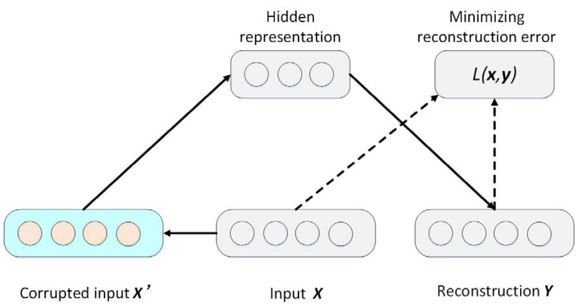


Figure 3: The corrupted process in Denoising Autoencoder.

C. Stacked Denoising Autoencoder Prediction Model

SDA is make up of DAEs by stacking, and the output of the curerent hidden layer as the input of the next hidden layer, which forms a deep learning network [42]. More cleary, considering stacked denoising autoencoder prediction model(SDAPM) with l layers, the first layer is an input layer to input training data, the last second layer is prediction layer, which is used to predict short-term internet traffic flow; the last layer is output layer to output predicted data; and other layers called hidden layers are SDAs, which are used to feature expression. The construction of SDAPM is shown as Fig.4.

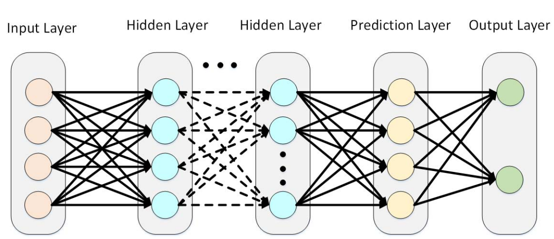


Figure 4: The construction of a SDAPM. The hidden layers ate autoencoders to feature expression, and the prediction layer is a nonlinear prediction function.

The traditional prediction layer of SDA is logistic regression model, which is used to solve multi-value classification problem. However,in SDAPM, the internet traffic flow prediction problem actually is a nonlinear regression problems, thus we put a sigmoid function $f(x) = sigmoid(\boldsymbol{Wx} + \boldsymbol{b})$ into the prediction layer to predict traffic flow and this function is also used to supervise the predicted traffic flow and fine-tune the model.

D. Prediction Algorithm Design

In this subsection, we will introduce how to obtain SDAPM, which is consist of two parts: Data Preprocessing Algorithm and SDAPM Generation Algorithm. Data Preprocessing Algorithm is used to process the original sample data to the expected format; and the SDAPM Generation Algorithm is used to train the SDAPM.

The sample date of internet traffic flow usually contains some missing data entry, and we need to fill the missing data. Moreover, in order to make the model better and faster convergence, it is needed to make a linear transformation of the sample data and map the results to $[0, 1]$. Finally, a isotropic Gaussian noise is added into the sample data to improve the generalization ability and adaptive ability of the model. The preprocess procedure is stated as algorithm 1. When the preprocessed sample data is ready, the core task is how to train the forecast model of SDAPM.

Firstly, we need to set some parameters of SDAPM, which contains pre-train epochs pte, training epochs te, the learning rate of pre-train phase $ptlr$, the learning rate of train phase tlr, the batch input size bts.

Secondly, initialize SDAMP, which is make up of for parts: determine the input dimension and label dimension, split the sample date, set the number of layers of the model and the number of neurons per layer, and set binary masking noise per layer. Where determining the input dimension and label dimension is that determine input dimension containing how long previous time intervals data, and the input data should contain how many label data items.

Thirdly, pre-training the hidden layers of SDAMP, which contains the following steps:

1) Take the first layer as the input layer;

2) Train each hidden layer as an autoencoder taking the front layer's output as the input;

3) Determine the weight value of per hidden layer $\boldsymbol{W}$ and constant vector $\boldsymbol{b}$.

Fourthly, fine-tune SDAMP, means that fine-tune per hidden layer $\boldsymbol{W}$ and constant vector $\boldsymbol{b}$ to minimize validation loss value in validation set and the test loss value in test set, which mainly contains the following steps:

1)computer prediction traffic flow value by prediction layer;

2) Minimize average cost value by computing the difference between prediction value and the label value with BP method;

3) Store the current parameters set of SDAMP, find the best parameter set per hidden layer and prediction layer by computing validation loss value and the test loss value.

The pseudo code of SDAPM Generation is shown as Alg.2.

IV. Experiments

In this section, we will introduce internet traffic flow data set collected, and then the performance evaluation indicators are introduced to comparison and analysis. Finally, the experiment

Algorithm 1 Data Preprocessing Algorithm

Input: the sample data set *X*, SNR=30dB
Output: the preprocessed sample data set $\tilde{X}$

Step 1) Uniformization the sample data
for each data item x in *X*
    $x' = \frac{x - x_{min}}{x_{max} - x_{min}}$
end for
Step 2) Dealing with missing data
for each data item x' in *X'*
    if x' is missing data
        find some non-missing data x'' near with x'
        $x' = \frac{1}{n}\sum_{i=1}^{n} x_i^{''}$
    end if
end for
Step 3) Add the gaussian noise
for each data item x' in *X'*
    $\sigma^2 += x^{'} * x^{'}$
end for
$sigma = \sigma^2 / 10^{SNR/10}$
for each data item x' in *X'*
    $x' += \varepsilon$, $\varepsilon \sim N(0, sigma)$
    $x \ = x'$
end for
return $\tilde{X}$

Algorithm 2 SDAPM Generation Algorithm

Input:
the preprocessed data set *X*, the pre-train epochs *pte*, training epochs *te*, the learning rate of pre-train phase *ptlr*, the learning rate of train phase *tlr*, the batch input size *bts*
Output: the trained SADPM model, sdapm

Step 1) Initialize SDAMP
Determine the input dimension x_{in} and label dimension x_{la} for each x in *X*
The sample data set *X* is divided into training set X_1, validation set X_2 and test set X_3
Set the number of layers of the model nl and the number of neurons nnl_i per layer dA in nl
Add the probability p_i of binary masking noise per layer dA in nl
Initialize weight matrices ***W*** and bias vectors layers ***b***
Step 2) Pre-training the hidden layers of SDAMP
for each layer dA of SDAMP
    for epoch i in xrange(*pte*)
        for each input ***x*** in training set X_1
            greedy training hidden layer dA with gradient descent method
        end for
    end for
end for
Step 3) Fine-tune SDAMP
for each input ***x*** in training set X_1
    computer prediction value ***y***
    minimize average cost value avc = ***y***− ***x*** with BP method
    computer validation loss value $valoss$ in validation set X_2
    if $valoss$ is current minimum
        computer mean relative error in test set X_3
    end if
end for
return SADPM model parameters of $sdapm$

results are described in detail and shows the superiority of our model.

A. Data Description

The experiment data used to the proposed deep architecture model was collected from internet network in Tianjin,China, owned by China United Network Communications. The internet traffic data are aggregated 15-min interval for each network port. In this paper, the internet traffic flow data collected in the last two months of the year 2015 were used for the experiments. Fig.5 shows a typical network's internet traffic flow over 15-min interval for some days in the sample data.

B. Performance Indicators

In this paper, we use three performance indicators to evaluate the effectiveness of the proposed model, that are the mean absolute error(MAE), the mean relative error(MRE), and the relative mean square error(RMSE). The three indicator are defined as

$$MAE = \frac{1}{n}\sum_{i=1}^{n} \mid x_i - \hat{x_i} \mid \tag{6}$$

$$MRE = \frac{1}{n}\sum_{i=1}^{n} \frac{\mid x_i - \hat{x_i} \mid}{f_i} \tag{7}$$

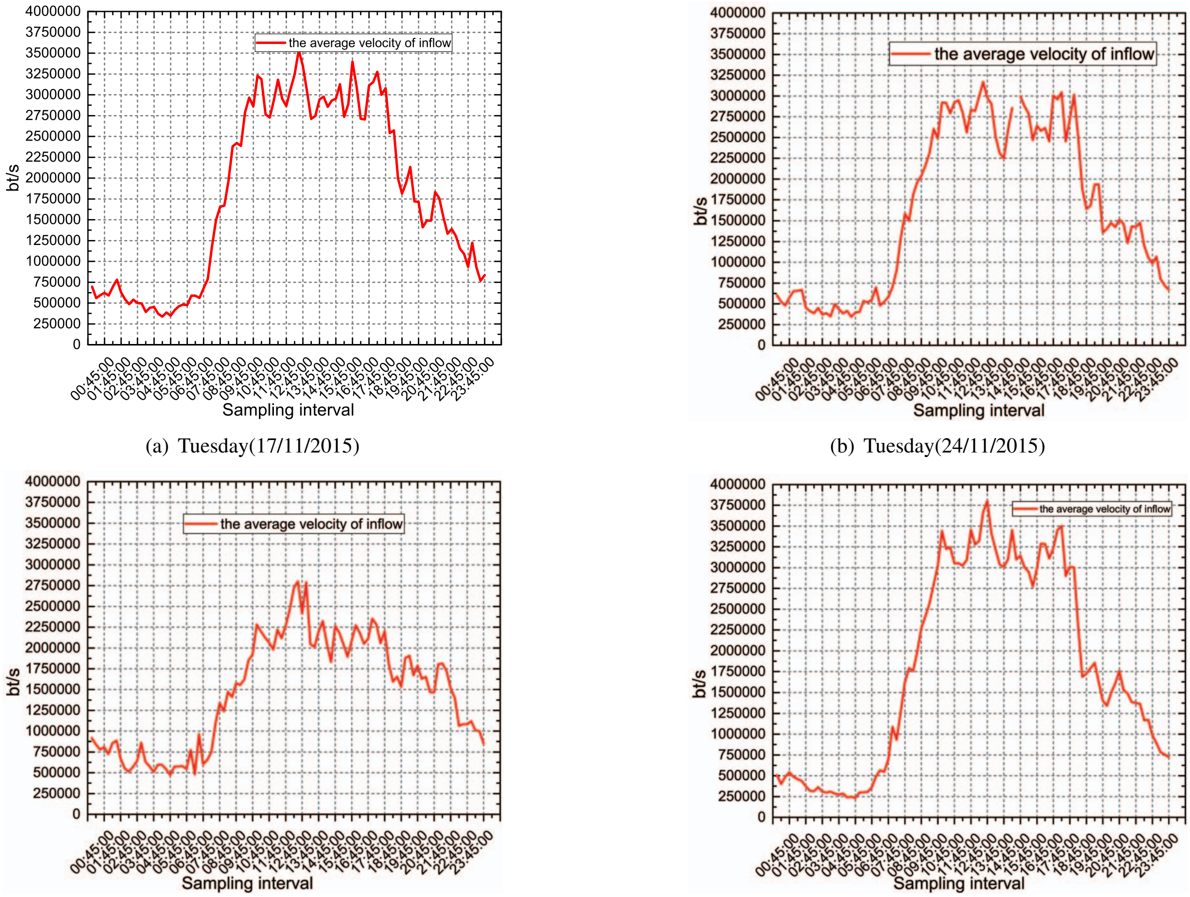
$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n} \mid x_i - \hat{x_i} \mid^2} \tag{8}$$

where x_i is the observed internet traffic flow (label), and $\hat{x_i}$ is the predicted internet traffic flow.

C. The Optimal Parameter Settings and Performance Evaluation

In this paper, we used the proposed model(SDAMP) to predict 15-min internet traffic flow. Before the experiment, we need to set model parameters of SDAMP, such as the number of hidden layers, and the number of hidden units in each hidden layer, and so on. In order to get the best parameters that are suitable for this scenario, for each parameter in Tab.I, experiments are carried out with other parameters unchanged.

We also analyse the effect of parameter setting of our proposed model on model performance, and find some sensitive parameters that are shown as Fig.6 and Fig.7. It should be noted that, in order to draw a more clear picture containing MAE, MRE, and RMSE, the sample data is mapped to [0,1], MRE is multiplied by 100 in Fig.6 and Fig.7, and each experiment performed three times and performance indicators



(a) Tuesday(17/11/2015)

(b) Tuesday(24/11/2015)

(c) Saturday(12/12/2015)

(d) Monday(14/12/2015)

Figure 5: Typical daily internet traffic flow pattern.

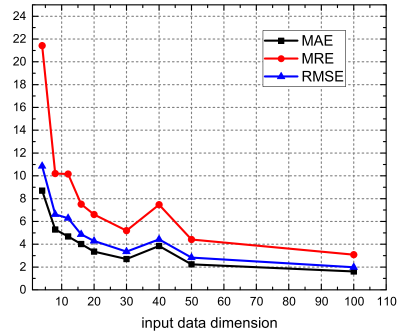
Table I: The Parameters Set of SDAPM for Internet Traffic Flow

Parameter Name	Symbol	Initial Value	Final Value
sample data partition ratio	spr	6:2:2	6:2:2
noise ratio of Gaussian	SNR	30db	30db
input data dimension	x_{id}	15	16
input layer dimension	x_{il}	10	8
label dimension	x_{la}	5	8
number of hidden layers	nl	4	4
number of hidden units	nnl_i	[25,25,25,25]	[8,8,8,8]
binary masking noise probability	p_i	[0.01,0.02,0.03,0.04]	[0.01,0.02,0.03,0.04]
pre-train epochs	pte	100	100
training epochs	te	50	50
pre-train learning rate	$ptlr$	0.0001	0.0001
train learning rate	tlr	0.001	0.001
the batch input size	bts	1	1

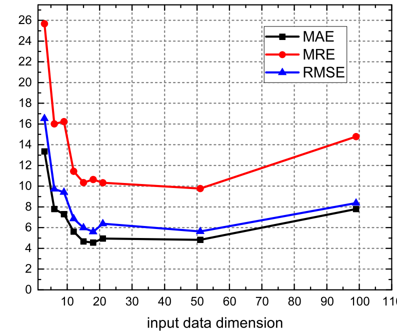
value are averaged.

1) input data dimension x_{id} and the ratio $x_{il} : x_{la}$ of the input layer dimension to label dimension vs. the performance of the SDAPM model: As described in Fig.6, we observe that there are fatal differences on the SDAPM model performance between the input layer dimension and the ratio $x_{il} : x_{la}$ of the input layer dimension to label dimension. In general, MAE, MRE and RMSE of SDAPM sharply decreases while input data dimension x_{id} increases, and as the ratio $x_{il} : x_{la}$ increases, the performance of SDAPM distinctly decreases. When the ratio $x_{il} : x_{la} = 1$ and $x_{id} > 15$, the average MAE, MRE and REME are lower than 2.97, 5.72%, and 5.06, respectively. At same time, the probability of over fitting gradually increases. According to these results, the input data dimension x_{id} is 16 and input layer dimension divided by label dimension $x_{il} : x_{la}$ is set $1 : 1$.

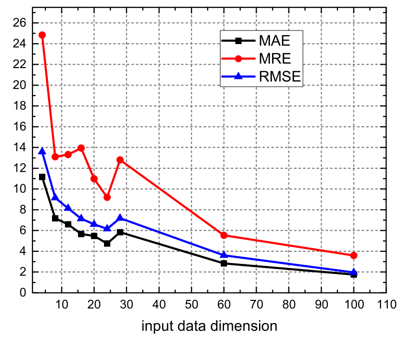
2) number of hidden layers nl and the ratio $x_{il} : nnl_i$ of the input layer dimension to number of hidden units vs. the performance of the SDAPM model: As it is shown as Fig.7, number of hidden layers nl and the ratio $x_{il} : nnl_i$ of the input layer dimension to number of hidden units have a significant impact on the performance of the SDAPM. In general, MAE, MRE and RMSE of SDAPM firstly decrease and then slowly



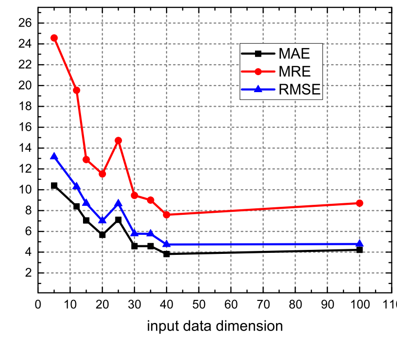
(a) $x_{il} : x_{la} = 1 : 1$



(b) $x_{il} : x_{la} = 2 : 1$

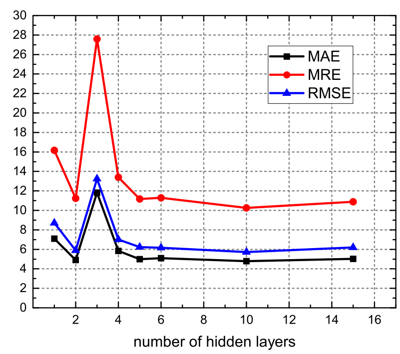


(c) $x_{il} : x_{la} = 3 : 1$

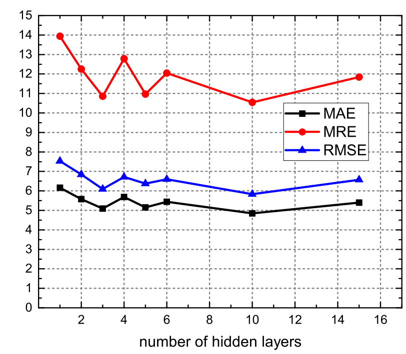


(d) $x_{il} : x_{la} = 4 : 1$

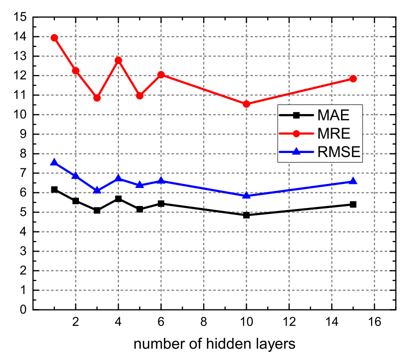
Figure 6: input data dimension x_{id} and the ratio $x_{il} : x_{la}$ of the input layer dimension to label dimension vs. the performance of the SDAPM model.



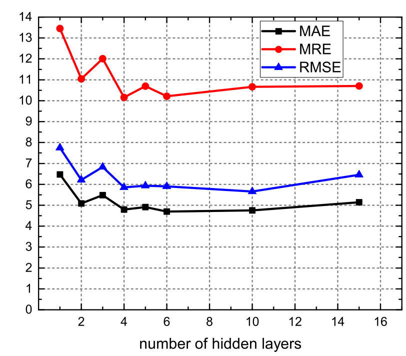
(a) $x_{il} : nnl_i = 2 : 1$



(b) $x_{il} : nnl_i = 1 : 1$



(c) $x_{il} : nnl_i = 2 : 3$



(d) $x_{il} : nnl_i = 2 : 5$

Figure 7: number of hidden layers nl and the ratio $x_{il} : nnl_i$ of the input layer dimension to number of hidden units vs. the performance of the SDAPM model.

Table II: The Performance of SDAPM vs. MLP for Internet Traffic Flow

	SDAPM	MLP
MAE	3.362119089	3.616250717
MRE	6.315260949	6.678017925
RMSE	4.225000029	4.438654154

rise when nl increases, and the ratio $x_{il} : nnl_i$ has a significant fluctuations on the performance of SDAPM. When the ratio $x_{il} : nnl_i = 1$ and hidden layers $nl \in [4, 6]$, the SDAPM has a a relatively stable performance, and the standard variance of MAE, MRE, and RMSE are lower than 0.265, 0.512 and 0.246, respectively. According to these results, number of hidden layers nl is set as 4 and number of hidden units nnl_i in each layer is set as the same as input layer dimension x_{il}.

In addition, binary masking noise probability can not be set too large to prevent the model remember the noise characteristics resulting in over fitting.

According to our experimental results, the data is divided into training set, validation set and test set according to the ratio of 6:2:2, and the dimension of input data $\boldsymbol{x}$ is set 16, and label dimension is set 8; For the hidden layers here, we set the number of hidden units nl as 4, the number of hidden units in each hidden layer as $[8, 8, 8, 8]$, and other parameters unchanged. The final results are shown in Tab.I.

D. *Performance Evaluation*

In this section, we implement the SDAPM and Multilayer Perceptron model (MLP) in depth learning framework Theano. The experiment data is the internet traffic flow data of certain Network element port, which is collected in the last two months of the year 2015 by China United Network Communications. The data is divided into 3 parts in the proportion of 6:2:2, that is train set, validation set and test set, respectively. The train set and validation set are used to train the model, the test set is used to evaluate the effectiveness of the model. Tab.II presents the output of the proposed model and MLP for 15-min traffic flow prediction. In Tab.II, it is shown that the MAE, MRE and RMSE are better than that of MLP. Specifically, the MAE, MRE and RMSE of MLP is 7.08%, 5.74%, and 5% more than that of SDAMP. A visual display of the performance of the MRE derived with SDAMP and MLP is given in Fig.8. It displays for each method the cumulative distribution function (cdf) of the MRE, which describes the statistical results on 15-min traffic flow prediction. The method that uses SDAMP leads to improved traffic flow prediction performance when compared with the MLP. Specifically, when MRE is more than 0.5, the cdf of SDAMP is always better than that of MLP. Even when MRE is 0.8, the cdf of SDAMP is 1, while the value of MLP is only about 0.9. These results shows our proposed model effective for internet traffic flow prediction. Thus, the effectiveness of the SDAMP method for traffic flow prediction is promising and manifested.

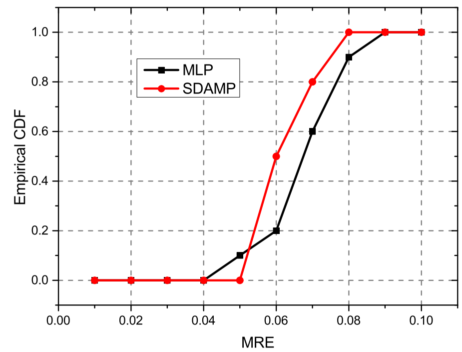


Figure 8: Empirical cdf of the MRE of SDAMP vs. MLP.

V. Conclusion

In this paper, we studied stacked denoising autoencoder in internet traffic flow prediction, and exploited the deep learning concepts and big data to improve the forecast accuracy. We collected the sample data, developed a SDAPM, which contains data preprocessing and SDAPM generation to predict internet traffic. We also investigated the parameter optimization in SDAPM, and then show our proposed model is effective.

Acknowledgment

This work is supported by the National Natural Science Foundation of China No.61572062, and No.61502019, the National High Technology Research and Development Program of China (863 Program) No. 2015AA016105, National Key Research and Development Program of China No.2016YFB1000503. The authors would like to thank great support.

References

[1] Hong, Chi-Yao, et al., *"Achieving high utilization with software-driven WAN,"* in ACM SIGCOMM Computer Communication Review. Vol. 43. No. 4. ACM, 2013.

[2] J. Zhang et al., *"Data-driven intelligent transportation systems: A survey,"* in IEEE Trans. Intell. Transp. Syst., vol. 12, no. 4, pp. 1624–1639, Dec. 2011.

[3] C. L. Philip Chen and C.Y. Zhang, *"Data-intensive applications, challenges, techniques and technologies: A survey on Big Data,"* in Inf. Sci., vol. 275, pp. 314–347, Aug. 2014.

[4] Y. Bengio, *"Learning deep architectures for AI,"* in Found. Trends Mach. Learn., vol. 2, no. 1, pp. 1–127, Jan. 2009.

[5] G. E. Hinton and R. R. Salakhutdinov, *"Reducing the dimensionality of data with neural networks"*,in Science, vol. 313, no. 5786, pp. 504–507, Jul. 2006.

[6] R. Collobert and J. Weston, *"A unified architecture for natural language processing: Deep neural networks with multitask learning,"* in Proc. 25th ICML, 2008, pp. 160–167.

[7] I. J. Goodfellow, Y. Bulatov, J. Ibarz, S. Arnoud, and V. Shet, *"Multi-digit number recognition from street view imagery using deep convolutional neural networks,"* in arXiv preprint arXiv:1312.6082, 2013.

[8] B. Huval, A. Coates, and A. Ng, *"Deep learning for class-generic object detection,"*in arXiv preprint arXiv:1312.6885, 2013.

[9] H. C. Shin, M. R. Orton, D. J. Collins, S. J. Doran, and M. O. Leach, *"Stacked autoencoders for unsupervised feature learning and multiple organ detection in a pilot study using 4D patient data,"* in IEEE Trans. Pattern Anal. Mach. Intell., vol. 35, no. 8, pp. 1930–1943, Aug. 2013.

[10] Haykin S., *"Neural Networks: A Comprehensive Foundation,"* in 2nd edn. Prentice Hall PTR, Upper Saddle River,1998.

[11] Bonald T., *"The Erlang model with non-poisson call arrivals,"* in ACM SIGMETRICS Performance Evaluation Review, 2006, 34(1): 276–286.

[12] Leland WE, Taqqu MS, Willinger W, Wilson DV, *"On the self-similar nature of Ethernet, traffic,"*in ACM SIGCOMM Computer Communication Review, 1993, 23(4): 183–193.

[13] M. S. Ahmed and A. R. Cook, *"Analysis of freeway traffic time-series data by using Box-Jenkins techniques,"*in Transp. Res. Rec., no. 722, pp. 1–9, 1979.

[14] Granger, Clive WJ, and Roselyne Joyeux, *"An introduction to long-memory time series models and fractional differencing,"* in Journal of time series analysis 1.1 (1980): 15–29.

[15] Periyanayagi, S., and V. Sumathy, *"S-ARMA model for network traffic prediction in wireless sensor networks,"* in J. Theor. Appl. Inf. Technol 60 (2014): 524–530.

[16] Y. Kamarianakis and P. Prastacos, *"Forecasting traffic flow conditions in an urban networkComparison of multivariate and univariate approaches,"* in Transp. Res. Rec., no. 1857, pp. 74-84, 2003.

[17] B. M. Williams and L. A. Hoel, *"Modeling and forecasting vehicular traffic flow as a seasonal ARIMA process: Theoretical basis and empirical results,"* in J. Transp. Eng., vol. 129, no. 6, pp. 664–672, Nov./Dec. 2003.

[18] M. vanderVoort, M. Dougherty, and S. Watson, *"Combining Kohonen maps with ARIMA time series models to forecast traffic flow,"* in Transp.Res. C, Emerging Technol., vol. 4, no. 5, pp. 307–318, Oct. 1996.

[19] B. Ghosh, B. Basu, and M. OMahony, *"Multivariate short-term traffic flow forecasting using time-series analysis,"* in IEEE Trans. Intell. Transp. Syst., vol. 10, no. 2, pp. 246–254, Jun. 2009.

[20] G. A. Davis and N. L. Nihan, *"Nonparametric regression and short-term freeway traffic forecasting,"* in J. Transp. Eng., vol. 117, no. 2, pp. 178–188, Mar./Apr. 1991.

[21] H. Chang, Y. Lee, B. Yoon, and S. Baek, *"Dynamic near-term traffic flow prediction: System oriented approach based on past experiences,"* in IET Intell. Transport Syst., vol. 6, no. 3, pp. 292–305, Sep. 2012.

[22] N. E. El Faouzi, *"Nonparametric traffic flow prediction using kernel estimator,"* in Iin Proc. 13th ISTTT, 1996, pp. 41–54.

[23] H. Y. Sun, H. X. Liu, H. Xiao, R. R. He, and B. Ran, *"Use of local linear regression model for short-term traffic forecasting,"* in Transp. Res. Rec., no. 1836, pp. 143–150, 2003.

[24] S. Sun, C. Zhang, and Y. Guoqiang, *"A Bayesian network approach to traffic flow forecasting,"*in IEEE Intell. Transp. Syst. Mag., vol. 7, no. 1, pp. 124–132, Mar. 2006.

[25] Y. S. Jeong, Y. J. Byon, M. M. Castro-Neto, and S. M. Easa, *"Supervised weighting-online learning algorithm for short-term traffic flow prediction,"* in IEEE Trans. Intell. Transp. Syst., vol. 14, no. 4, pp. 1700–1707, Dec. 2013.

[26] M. C. Tan, S. C. Wong, J. M. Xu, Z. R. Guan, and Z. Peng, *"An aggregation approach to short-term traffic flow prediction,"* in EEE Trans. Intell. Transp. Syst., vol. 10, no. 1, pp. 60–69, Mar. 2009.

[27] S. A. Zargari, S. Z. Siabil, A. H. Alavi, and A. H. Gandomi, *"A computational intelligence-based approach for short-term traffic flow prediction,"* in Expert Syst., vol. 29, no. 2, pp. 124–142, May 2012.

[28] I. Okutani and Y. J. Stephanedes, *"Dynamic prediction of traffic volume through Kalman filtering theory,"* in Trans. Res. B, Methodol., vol. 18, no. 1, pp. 1?1, Feb. 1984.

[29] R. Tahmasbi and S. M. Hashemi, *"Modeling and forecasting the urban volume using stochastic differential equations,"* in IEEE Trans. Intell. Transp. Syst., vol. 15, no. 1, pp. 250–259, Feb. 2014.

[30] G. Comert and A. Bezuglov, *"An online change-point-based model for traffic parameter prediction,"* in IEEE Trans. Intell. Transp. Syst., vol. 14, no. 3, pp. 1360–1369, Sep. 2013.

[31] L. Li, W. H. Lin, and H. Liu, *"Type-2 fuzzy logic approach for short-term traffic forecasting,"* in Proc. Inst. Elect. Eng.-Intell. Transp. Syst., vol. 153, no. 1, pp. 33–40, Mar. 2006.

[32] S. Shiliang and X. Xin, *"Variational inference for infinite mixtures of Gaussian processes with applications to traffic flow prediction,"* in IEEE Trans. Intell. Transp. Syst., vol. 12, no. 2, pp. 466?75, Jun. 2011.

[33] G. Duncan and J. K. Littlejohn, *"High performance microscopic simulation for traffic forecasting,"* in Proc. IEE Colloq. Strategic Control InterUrban Road Netw. (Dig. No 1997/055), 1997, pp. 4/1–4/3.

[34] M. Ben-Akiva, E. Cascetta, and H. Gunn, *"An on-line dynamic traffic prediction model for an inter-urban motorway network,"* in Urban Traffic Networks, N. Gartner and G. Improta, Eds. Berlin, Germany: SpringerVerlag, 1995, pp. 83–122.

[35] Yisheng Lv, Yanjie Duan, Wenwen Kang, Zhengxi Li, and Fei-Yue Wang, *"ATraffic Flow Prediction With Big Data: A Deep Learning Approach,"* in IEEE Trans. Intell. Transp. Syst., vol. 16, no. 2, APR. 2015, pp. 865–873.

[36] Wenhao Huang, Guojie Song, Haikun Hong, and Kunqing Xie *"Deep Architecture for Traffic Flow Prediction: Deep Belief Networks With Multitask Learning,"* in IEEE Trans. Intell. Transp. Syst., vol. 15, no. 5, OCT. 2014, pp. 2191–2201.

[37] Tiago Prado Oliveira, Jamil Salem Barbar, Alexsandro Santos Soares, *"Multilayer Perceptron and Stacked Autoencoder for Internet Traffic Prediction,"* in Proc. 11th IFIP International Conferences on Networking, 2014, pp. 61–71.

[38] CHEN M, WEINBERGER K, SHA F, et al, *"Marginalized denoising auto-encoders for nonlinear representations,"* in Proc. the 31th International Conference on Machine Learning, New York: ACM, 2014, pp.1476–1484.

[39] G. E. Hinton, S. Osindero and Y.-W. Teh, *"A fast learning algorithm for deep belief nets,"* in Neural Comput., vol. 18, no. 7, pp. 1527–1554, Jul. 2006.

[40] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, *"Greedy layerwise training of deep networks,"* in Proc. Adv. NIPS, 2007, pp. 153–160.

[41] RUMELHART D E, HINTON G E, WILLIAMS R J., *"Learning representations by back-propagating errors,"* in Nature, 1986,323(9), pp.533–536.

[42] VINCENT P, LAROCHELLE H, LAJOIE I, et al., *"Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion,"* in Journal of Machine Learning Research, 2010,11(6), pp. 3371–3408.