AN EXPERIMENTAL INVESTIGATION OF MOBILE NETWORK TRAFFIC PREDICTION ACCURACY

Ali Yadavar Nikravesh Samuel A. Ajila Chung-Horng Lung
Department of Systems and Computer Engineering
Carleton University
{alinikravesh, ajila, chlung}@sce.carleton.ca

Wayne Ding
LTE System
Business Unit Radio Ericsson
wayne.ding@ericsson.com

Abstract

The growth in the number of mobile subscriptions has led to a substantial increase in the mobile network bandwidth demand. The mobile network operators need to provide enough resources to meet the huge network demand and provide a satisfactory level of Quality-of-Service (QoS) to their users. However, in order to reduce the cost, the network operators need an efficient network plan that helps them provide cost effective services with a high degree of QoS. To devise such a network plan, the network operators should have an in-depth insight into the characteristics of the network traffic. This paper applies the time-series analysis technique to decomposing the traffic of a commercial trial mobile network into components and identifying the significant factors that drive the traffic of the network. The analysis results are further used to enhance the accuracy of predicting the mobile traffic. In addition, this paper investigates the accuracy of machine learning techniques – Multi-Layer Perceptron (MLP), Multi-Layer Perceptron with Weight Decay (MLPWD), and Support Vector Machines (SVM) – to predict the components of the commercial trial mobile network traffic. The experimental results show that using different prediction models for different network traffic components increases the overall prediction accuracy up to 17%. The experimental results can help the network operators predict the future resource demands more accurately and facilitate provisioning and placement of the mobile network resources for effective resource management.

Keywords: Mobile Network, Traffic Analysis, Prediction, Multi-Layer Perceptron, Multi-Layer Perceptron with Weight Decay, Support Vector Machine

1. INTRODUCTION

In recent years, mobile data traffic has increased rapidly [1]. The analysis reports show the mobile data traffic grows by 60 percent year-on-year [2]. The growth in the mobile data traffic is due to the rising number of smartphone subscriptions, as well as the increasing data consumption per subscriber [2]. The ubiquity of smartphones and the increasing amount of data generated by mobile phone users give rise to enormous datasets which can be used to characterize and understand user mobility, communication, and interaction patterns [3].

The increase in the mobile network traffic necessitates the mobile network operators to deal with a network resource management issue. Deciding the right amount of network resources is a nontrivial task and may lead to either under-provisioning or over-provisioning conditions. Under-provisioning condition occurs when the provisioned network resources are not adequate to serve the workload of the network, which may cause users' dissatisfaction. On the other hand, over-provisioning condition is the result of provisioning an excessive amount of network resources, which leads to the waste of valuable network resources, such as the spectrum. To prevent the under-provisioning and the over-provisioning conditions, mobile network operators need to gain insight into the factors that affect the network traffic. Knowing the characteristics of the network traffic helps the operators devise an effective resource provisioning plan and accommodate future traffic demands more efficiently.

This paper investigates the network resource provisioning issue. Network resource provisioning is similar to cloud resource provisioning. Intensive research has been conducted on cloud resource management and provisioning. The objective of this research is to apply the methodology and techniques used in predicting the cloud resources to the prediction of network resource usage by using a real life dataset from a commercial trial mobile network.

A comparison between three machine learning algorithms (i.e., Support Vector Machine, Multi-Layer Perceptron, and Multi-Layer Perceptron with Weight Decay) to predict the future traffic of a mobile network is presented in our previous paper [4]. According to our previous paper [4], dimensionality of the traffic data can affect the prediction accuracy of the regression models. Based on our results, Support Vector machine (SVM) outperforms Multi-Layer Perceptron with Weight Decay (MLPWD) and Multi-Layer Perceptron (MLP) in predicting the multidimensionality of the real-life traffic data, while MLPWD has better accuracy in predicting the unidimensional data. In addition, the experimental results in [4] indicate that using multidimensional traffic datasets significantly increases the prediction accuracy of the MLP, MLPWD, and SVM algorithms. However, since none of the future values of the network traffic attributes is known a priori, it is not feasible to use a multidimensional dataset to predict the future network traffic. Therefore, the goal of this paper is to improve accuracy of the regression models to predict the unidimensional mobile network traffic datasets.

This paper enhances our previous unidimensional prediction results by further analyzing individual factors that affect the mobile network traffic. This paper uses time-series analysis technique to decompose the mobile network traffic into *trend*, *seasonal* and *remainder* components. The trend component shows the long term direction of the traffic and indicates whether the network traffic is increasing or decreasing. The seasonal component represents the repetitive and predictable movement around the trend line of the network traffic. Knowing the trend and the seasonality of the traffic helps the network operators to accommodate the future traffic more efficiently.

Moreover, this paper investigates the accuracy of the MLP, MLPWD, and SVM algorithms to predict the future values of each of the network traffic components (i.e., trend, seasonal, and remainder). This facilitates to improve the overall prediction accuracy by using the most accurate prediction model for each of the traffic components. According to [5], SVM and Artificial Neural Networks (ANN) are effective algorithms to predict future system characteristics. Therefore, in this paper we compare SVM and two variations of ANN algorithm (i.e., MLP and MLPWD) to verify their accuracy in predicting the future behavior of mobile network traffic's components. The main difference between the MLP and MLPWD algorithms is the risk minimization principle that is used to create MLP and MLPWD regression models. Therefore, comparing MLP with MLPWD shows the impact of the risk minimization principle on the prediction accuracy of the artificial neural networks. Furthermore, comparing MLPWD and MLP with SVM can help evaluate the ANN against the vector machines and highlight the capability of each of the prediction models to predict the components of the mobile network traffic. The contributions of this paper are:

- Decomposing a real life mobile network traffic dataset into components and providing an insight into the factors that are likely to affect the future network traffic.
- Comparing the accuracy of the MLP, MLPWD, and SVM algorithms to predict future behavior of individual components of the mobile network traffic.
- Analyzing the impact of the sliding window size on the prediction accuracy of MLP, MLPWD, and SVM algorithms.

The remainder of the paper is organized as follows: Section 2 discusses the background and related work. This is followed by the presentation of experiments and analysis of results in section 3. Conclusions and possible directions for the future research are discussed in section 4.

2. BACKGROUND AND RELATED WORK

This section briefly introduces fundamental concepts that are used in the paper. Sub-section 2.1 describes the machine learning concept and the prediction algorithms used in the experiment. In sub-section 2.2 an overview of the mobile network resource provisioning approaches is presented.

2.1 MACHINE LEARNING

Machine learning is a study of algorithms which can learn complex relationships or patterns from empirical data and make accurate decisions [6]. Machine learning includes a broad range of techniques such as data pre-processing, feature selection, classification, regression, association rules, and visualization. In big data analytics, machine learning techniques can help extract insightful information from the enormous datasets and identify hidden relationships.

Vapnik indicates that machine learning corresponds to the problem of function approximation [7]. Based on this definition, the machine learning regression objective is to find the best available approximation to a given time-series. To choose the best approximation, the loss function between the actual values of the time-series and the response provided by the learning machine should be measured. The expected value of the loss, given by the risk function, is:

$$R(W) = \int L(y, f(x,w))dP(x,y) \qquad (1)$$

where $f(x,w)$ is the response provided by the machine learning algorithm, given x is the input and w is the parameter of the function, y is the actual value of the time-series, and $L(y, f(x,w))$ is the loss function. The problem in minimizing the functional risk is that the joint probability distribution $P(x,y) = P(y|x)P(x)$ is unknown and the only available information is contained in the training set. In other words, we only have the supervisor's response for the training set, and there is no access to the supervisor's response for the testing data set.

Since in the regression problems the actual future values of the time-series are unknown (i.e., $P(y|x)$ is unknown), the loss function cannot be calculated. To solve the functional risk problem, Vapnik [7] proposes an induction principle of replacing the risk functional $R(w)$ by empirical risk functional:

$$E(w) = \frac{1}{l}\sum_{i=1}^{l} L\big(y_i, f(x_i,w)\big) \qquad (2)$$

where l is the size of the training dataset. The induction principle of empirical risk minimization (ERM) suggests that in the presence of specific conditions, the learning machine that minimizes functional risk over the training dataset (i.e., $E(w)$) is the learning machine that minimizes the risk function $R(w)$. Therefore, the function with the minimum empirical risk is the best approximation to the time-series.

Vapnik also proves that in the presence of specific conditions, ERM could lose its precision due to the over-fitting problem [8]. To prevent the over-fitting problem, structural risk minimization (SRM) principle is proposed to describe a general model of complexity control and to provide a trade-off between hypothesis space complexity (i.e., VC-dimension) and the quality of fitting the training data.

In our problem domain, mobile network traffic represents the time-series that is to be predicted. This paper aims to find the most accurate regression model that predicts mobile network traffic's future behavior. To this end, we compare three well-known machine learning regression models to investigate their precision in predicting network traffic. The machine learning regression models investigated in this paper are: MLP, MLPWD, and SVM. Each of the three algorithms is highlighted as follows:

2.1.1 Multi-Layer Perceptron (MLP)

MLP is a feed-forward ANN that maps input data to appropriate output. A MLP is a network of simple neurons called perceptron. Perceptron computes a single output from multiple real valued inputs by forming a linear combination to its input weights and putting the output through a nonlinear activation function. The mathematical representation of MLP output is:

$$y = \varphi\left(\sum_{i=1}^{n} w_i x_i + b\right) = \varphi(W^T X + b) \qquad (3)$$

where W denotes the vector of weights, X is the vector of inputs, b is the bias, and φ is the activation function.

MLP networks are typically used in supervised learning. Therefore, there are training and testing datasets that are used to train and evaluate the model, respectively. The training of MLP refers to adapting all the weights and biases to their optimal values to minimize the following equation:

$$E = \frac{1}{l}\sum_{i=1}^{l}(T_i - Y_i)^2 \qquad (4)$$

where T_i denotes the predicted value, Y_i is the actual value, and l is the training set size. Equation (4) is a simplified version of equation (2) and represents the ERM. In other words, MLP uses the ERM approach to create its regression model.

2.1.2 Multi-Layer Perceptron with Weight Decay (MLPWD)

The MLPWD uses the SRM approach to create prediction model. In addition to empirical risk, SRM describes a general model of capacity (or complexity) control and provides a trade-off between the complexity (i.e., VC-dimension) of the prediction model and the quality of fitting the training data. The general principle of SRM can be implemented in many different ways. According to [9], the first step to implement the SRM is to choose a class of functions with a hierarchy of nested subsets in the order of increasing complexity. The authors in [7] suggest three approaches to build a nested set of the functions implemented by neural networks:

- Create the nested set of the functions by the architecture of the neural network. This approach creates the nested set by increasing the number of the neurons in the hidden layer of the neural network.
- Create the nested set of the functions by the learning procedure. In this approach the architecture of the neural network (i.e., the number of the neurons and layers) is fixed. The nested set is created by changing the risk minimization equation (i.e., learn procedure)
- Create the nested set of the functions by preprocessing the input of the neural network. In this approach the architecture and the learning procedure of the neural network are fixed. The nested set is created by changing the representation of the input of the neural network.

The second proposed structure (i.e., given by the learning procedure) uses "weight decay" to create a hierarchy of nested functions. This structure considers a set of functions $S = \{f(x,w), w \in W\}$ that is implemented by a neural network with a fixed architecture. The parameters {w} are the weights of the neural network. A nested structure is introduced through $S_p = \{f(x,w), |(|w|)| \leq C_p\}$ and $C_1 < C_2 < \cdots < C_n$, where C_i is a constant value that defines the ceiling of the norm of neural network weights. For a convex loss function, the minimization of the empirical risk within the element S_p of the structure is achieved through the minimization of:

$$E(w,\gamma_p) = \frac{1}{l}\sum_1^l L(y_i, f(x_i,w)) + \gamma_p||w||^2 \quad (5)$$

The nested structure can be created by appropriately choosing Lagrange multipliers $\gamma_1 > \gamma_2 > \cdots > \gamma_n$.

Training MLP with weight decay means that during the training phase, each updated weight is multiplied by a factor slightly less than 1 to prevent the weights from growing too large. The risk minimization equation for MLPWD is:

$$E = \frac{1}{l}\sum_{i=1}^{l}(T_i - Y_i)^2 + \frac{\lambda}{2}\sum_{i=1}^{l} w_i{}^2 \qquad (6)$$

where l, T_i and Y_i are identical to that used in equation (4), w represents the weights in the neural network, and λ is the penalty coefficient of the sum of squares of weights.

The authors in [10] have shown that conventional weight decay technique can be considered as the simplified version of structural risk minimization in neural networks. Therefore, in this paper we use the MLPWD algorithm to investigate the accuracy of neural networks using SRM in predicting mobile network traffic.

2.1.3 Support Vector Machine (SVM)

SVM is a learning algorithm used for binary classification. The basic idea is to find a hyper-plane which perfectly separates the multidimensional data into two classes. Because input data are often not linearly separable, SVM introduces the notion of "kernel induced feature space" which casts the data into a higher dimensional space where the data are separable [11]. The key insight used in SVM is that the higher dimensional space does not need to be dealt with directly. In addition, similar to MLPWD, SVM uses SRM to create a regression model. Although SVM is originally being used for binary classification, it also has been extended to solve regression tasks and is termed Support Vector Regression (SVR). In this paper we use SVM and SVR interchangeably.

2.2 Network Resource Provisioning Approaches

Different researchers have performed network traffic analysis and prediction. The objective of the network traffic analysis is to get an insight into the types of network packets and the data flowing through a network. Network traffic analysis involves network data preprocessing, analysis (i.e., data mining), and evaluation.

Network traffic prediction (the scope of this paper) is useful for congestion control, admission control, and network bandwidth allocation [13]. Authors in [12] have categorized network traffic prediction techniques under three broad categories: linear time series model, nonlinear time series model, and hybrid model.

The linear time series models to predict network traffic data include Auto Regressive (AR), Moving Average (MA), and Autoregressive Moving Average (ARMA) techniques. Moving average generally generates poor results for time-series analysis [4]. Therefore, it is usually applied only to remove noises from the time-series. In addition, results of [14] show that the performance of auto-regression highly depends on the monitoring the interval length, the size of the history window, and the size of the adaptation window. ARMA is a combination of moving average and auto-regression algorithms and has been widely used for network traffic prediction [15][16][17].

Linear time series models are not accurate in environments with complex network traffic behaviors [12]. Therefore, researchers have used nonlinear time series models to forecast complex network traffic. ANN is the most popular non-linear model that is used in existing research works to predict network traffic data [18][19][20]. ANN has different variations, e.g., two popular variations MLP and MLPWD are introduced in Section 2.1.

Hybrid model techniques are a combination of linear and nonlinear models [21][22]. Authors of [13] have compared ARMA (i.e., linear), ANN (i.e., nonlinear), and FARIMA (i.e., hybrid) models and conclude that ANN outperformed other models.

To the best our knowledge, no research work has been published that looks at the prediction of a commercial network using a real life dataset. However, there are research works analyzing and trying to understand network data. The work by Tang et al. [23] analyzes South China city network data and develops a Traffic Analysis System (TAS). The work by Esteves et al. [24] examines twelve weeks' trace of a city building block local area wireless network. Further research work by Esteves et al. in [25] analyzes the performance of k-means and fuzzy k-means algorithms in the context of a Wikipedia dataset. In addition, provisioning of mobile network resources is mostly static, hence the network cannot dynamically adapt to traffic changes well. Often, the anticipated worst case scenario is considered in the lack of dynamic adaptation, which mostly results in over-provisioning and, hence, a waste of resources.

3. Experiments and Results

This section presents the experiments that decompose real-life mobile network traffic into its components and to analyze the behavior of each of the traffic components. In addition, the experiment presented in Section 3.6 compares the accuracy of the MLP, MLPWD, and SVM algorithms for predicting individual components of the network traffic.

3.1 Data Preparation and Cleaning

Experiments have been carried out by using a real-life dataset from a commercial trial mobile network. The initial network traffic dataset was composed of 1,012,959 rows and 27 columns (features), each row representing aggregated traffic of a cell (or a base station) in the network. Data were collected every hour between January 25, 2015 and January 31, 2015 from 5840 unique wireless network cells. To prepare the data for the experiment, we reduced the number of rows by selecting data of one of the network cells. The cell with the maximum number of data points was chosen to be investigated in this research. This resulted in a new dataset with 175 rows (i.e., the selected network cell has 175 network traffic data). Moreover, removing duplicated rows reduced the dataset size to 168 rows.

Parameter name	Value	Description
generateRanking	True	Whether or not to generate ranking
numToSelect	-1	Specifies the number of attributes to retain. The default value (-1) indicates that all attributes are to be retained
startSet	Null	Specifies a set of attributes to ignore.
threshold	-1.79	Set threshold by which attributes can be discarded. The default value (-1.79) results in no attributes being discarded.

Table 1. CorrelationAttributeEval Configuration Parameters

We also reduced the data dimension (i.e., number of columns) by using the WEKA attribute selection tool. The attribute selection process in WEKA is separated in two parts [26]:

- Attribute evaluator: a method by which attribute subsets are assessed
- Search method: a method by which the space of possible subsets is searched.

In this research, the CorrelationAttributeEval algorithm and the Ranker algorithm are used as the attribute evaluator and the search method, respectively. *Table 1* represents the CorrelationAttributeEval configuration parameters and their values.

Table 2 shows the attribute names, description, and correlation of the data used for the analysis.

Please note that the real attribute names are replaced by codes for the information disclosure reason.

According to *Table 2*, the last three attributes X25, X26, and X27 are not correlated to the rest of the attributes, i.e., the correlation value is 0 for all those attributes; hence, they are not useful for the machine learning model construction purpose. Therefore, to create machine learning models, the last three attributes (i.e., X25, X26, and X27) are discarded from the dataset. Removing the aforementioned attributes, results in a new dataset with 24 dimensions or attributes.

Machine learning algorithms predict the future value of a time-series dataset by discovering the relations between the features of the historical data and using the discovered relations for prediction. After the initial data preparation step, the dataset has 168 network traffic data points, and each data point has 24 dimensions (i.e., attributes). To perform prediction, at least one of the attributes should be selected as the target class (i.e., the attribute that is being predicted) and the rest of the attributes should be used to predict the target class. However, since none of the future values of the attributes is known a priori, it is not feasible to use them to predict the future value of the target class. Therefore, in this research one of the attributes is selected as the target class and the rest of the attributes are removed from the dataset.

The result is a new dataset with 168 data points and each data point has only one attribute (a unidimensional dataset). Since all of the 24 features in the initial dataset follow the same periodic pattern, we selected the X12 feature as the target class. Feature X12 is the number of the active users who are connected to the network cell and represents the workload of the cell. The reasons for selecting X12 as the target class are:

- Feature X12 represents the workload of the cell during a period and is a crucial parameter for the network operators.
- Most of the features in the dataset have a strong correlation with feature X12.

The experimental results in our previous paper [4] indicate the regression models are not very accurate in predicting the unidimensional network traffic datasets. The goal of this research is to increase the accuracy of the regression models to predict the unidimensional network traffic.

Attribute name	Description	Correlation
X1	PDCP signaling radio bearers volume at downlink (DL)	0.0678
X2	Total UEs scheduling time at uplink (UL)	0.0667
X3	Sum of radio resource control connections	0.0667
X4	Max radio resource control connection	0.0667
X5	PDCP signaling radio bearers volume at UL	0.0666
X6	PDCP latency at DL	0.066
X7	The aggregated scheduling time per cell at UL	0.0658
X8	Total UEs scheduling time at DL	0.0657
X9	PDCP data radio bearers volume at DL	0.0656
X10	The aggregated scheduling time per cell at DL	0.0655
X11	Total no. of packets for latency measurement at DL	0.0655
X12	Total no. of active user equipment at DL	0.0653
X13	PDCP packets received at UL	0.0653
X14	PDCP packets received at DL	0.0652
X15	Aggregated transport time over UEs at DL	0.0648
X16	Active user equipment	0.0641
X17	Min PDCP data radio bearers bit rate at UL	0.0639
X18	PDCP DRB vol. at DL	0.0637
X19	Aggregated transport time over UEs at UL	0.0629
X20	Max PDCP data radio bearers bit rate at UL	0.0613
X21	Max PDCP data radio bearers bit rate at DL	0.0601
X22	Min PDCP data radio bearers bit rate at DL	0.0553
X23	PDCP data radio bearers volume at UL	0.0537
X24	Aggregated transport time over UEs at UL	0.042
X25	Number of objects in measurement	0
X26	Sample of radio resource control connections	0
X27	Network cell ID	0

Table 2. Correlation Attributes Ranking

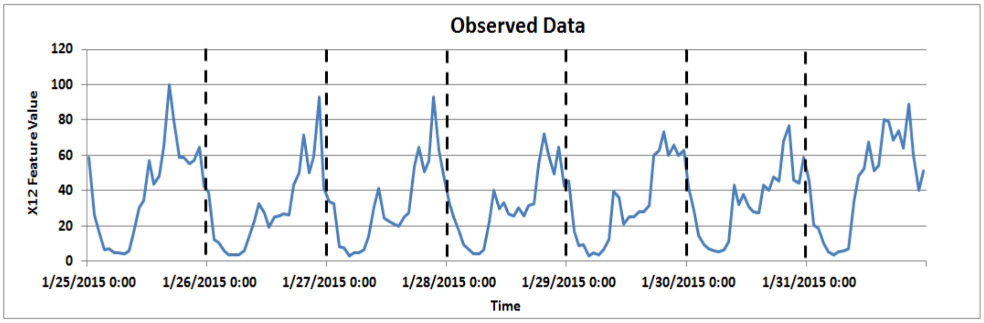

Figure 1. The network traffic time-series

3.2 TIME-SERIES DECOMPOSITION

Figure 1 shows the time-series which is composed of the values of feature X12 between January 25th, 2015 and January 31st, 2015. This section depicts the decomposition of the time-series into the components that affect its behavior. Any given time-series y_t typically consists of the following four components [27]:

- A *seasonal* component: a seasonal pattern exists when a series is influenced by seasonal factors (e.g., the quarter of the year or day of the week). Seasonality is always of a fixed and known period.
- A *trend* component: is the long term pattern of a time series, which can be positive or negative depending on whether the time series exhibits an increasing long term pattern or a decreasing long term pattern.
- A *cyclical* component: exists when the time-series exhibits rises and falls that are not of fixed period. The duration of these fluctuations is usually of at least 2 years. The time-series that is analyzed in this paper represents a period of seven days and does not include the cyclical component. Therefore, in this paper the cyclical component of the time-series is neglected.
- A *remainder* component: includes anything else after removing the trend, seasonal, and cyclical components from the time-series.

There are two approaches to mathematically model a time-series: additive time-series model (c.f. equation (7)) and multiplicative time-series model (c.f. equation (8)).

$$y_t = S_t + T_t + E_t \qquad (7)$$

$$y_t = S_t \times T_t \times E_t \qquad (8)$$

where t is the time, S_t is the seasonal component, T_t is the trend component, and E_t is the remainder component.

The additive model is most appropriate if the magnitude of the seasonal fluctuations or the variation around the trend line does not vary with the level of the time series. When the variation in the seasonal pattern or the variation around the trend appears to be proportional to the level of the time-series, then a multiplicative model is more appropriate. Because the magnitude of the seasonal fluctuations the time-series of *Figure 1* does not change, the additive model is used in this paper to model the time-series.

There are different methods for the additive decomposition of the time-series, such as classical decomposition, X-12-ARIMA decomposition, and Seasonal and Trend decomposition by using Loess (STL) [27]. The classical decomposition does not provide the trend values for the first few and the last few observations. In addition, the classical decomposition is not robust when there are occasional unusual small observations in the time-series. Furthermore, the X-12-ARIMA method only decomposes the quarterly and monthly time-series [27], and because the time-series of *Figure 1* represents a weekly pattern, the X-12-ARIMA method cannot be used in this research. The STL method is used in this paper to decompose the time-series. *Figure 2*, *Figure 3* and *Figure 4*, respectively, show the trend, seasonal, and remainder components of the mobile network traffic.

The grey bars on the right of *Figure 2*, *Figure 3* and *Figure 4* show the relative scales of the figures. The grey bars in all of the figures represent the same length but because the figures are on different scales, the bars vary in size. Therefore, the larger is the gray bar the smaller is the scale of the diagram. For instance the gray bar in *Figure 2* (i.e., the seasonal component) is larger than the gray bar in *Figure 3* (i.e., the trend component). This indicates that the seasonal component has smaller scale compared to the trend component. In other words, if the figures are shrunk until their bars became the same size, then all of the figures would be on the same scale.

The seasonal component shows a periodic pattern which is repeated twice a day. The seasonal component can be easily predicted with a machine learning prediction model.

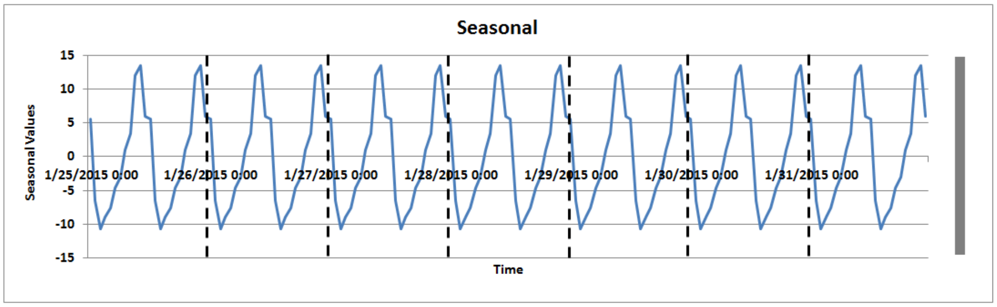

Figure 2. The seasonal component of the traffic

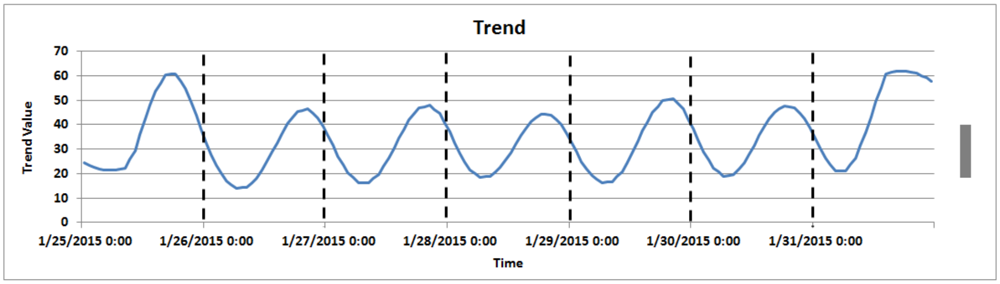

Figure 3. The trend component of the traffic

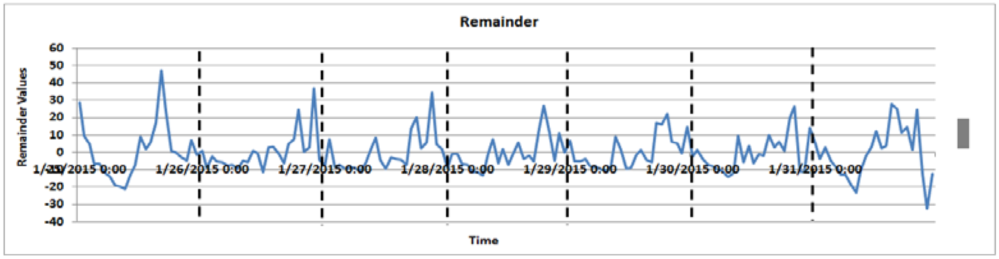

Figure 4. The remainder component of the traffic

However, since the variations in the seasonal component are small (i.e., the gray bar in *Figure 2* is large), increasing the prediction accuracy of this component cannot significantly increase the overall traffic prediction accuracy.

Similar to the seasonal component, the trend component of the network traffic follows a periodic repeatable pattern (c.f. *Figure 3*). According to the trend component, the traffic of the network cell has an increasing trend between 4:00 AM and 6:00 PM and decreases afterwards. In addition, the peak traffic increases during the weekends (i.e., the peaks of the trend curve are on Sunday 25th and Saturday 31st). Therefore, the network operators need to consider more resources during the peak hours and over the weekends.

Unlike the seasonal and the trend components, the remainder component does not follow a periodic pattern and includes unpredictable fluctuations (c.f. *Figure 4*). In addition, the variation of the remainder component is larger compared to the seasonal and trend components. Therefore, increasing the accuracy of the remainder component's predictions can significantly increase the overall traffic prediction accuracy. To increase the overall accuracy of the traffic predictions, in the following sections the accuracy of three machine learning algorithms (i.e., MLP, MLPWD, and SVM) to predict the future values of the components of the network traffic are evaluated.

3.3 TRAINING AND TESTING OF MP, MLPWD, AND SVM

The network traffic data used represent hourly performance condition of a network cell. Since the dataset includes 168 data points, the experiment duration is 168 hours. In our previous work [28] we demonstrated that the optimum training duration for the ANN and the SVM algorithms is 60% of the experiment duration. Therefore, we consider the first 100 data samples (i.e., 60%) of the mobile network traffic dataset as the training set and the rest 68 data samples as the testing set.

Because the datasets have only one feature, the sliding window technique is used to train and test the machine learning prediction algorithms. The sliding window technique uses the last k samples of a given feature to predict the future value of that feature. For example, to predict value of b_{k+1}, the sliding window technique uses $[b_1, b_2, \dots, b_k]$ values. Similarly, to predict b_{k+2}, the sliding window technique updates the historical window by adding the actual value of b_{k+1} and removing the oldest value from the window (i.e., the new sliding window is $[b_2, b_3, \dots, b_{k+1}]$).

Parameter name	MLP value	MLPWD value
Learning Rate (ρ)	0.3	0.3
Momentum	0.2	0.2
Validation Threshold	20	20
Hidden Layers	1	1
Hidden Neurons	$\frac{(\text{attributes} + \text{classes})}{2}$	$\frac{(\text{attributes} + \text{classes})}{2}$
Decay	False	True

Table 3. MLP and MLPWD configuration parameters

To reduce the effect of the over-fitting problem, cross-validation technique is used in the training phase. In this experiment 10 runs of 10-fold cross-validation technique is used to minimize the over-fitting effect. Readers are encouraged to see [29] for more details about the cross-validation technique. *Table 3* presents configuration of MLP and MLPWD algorithms in this experiment. The configuration of SVM is shown in *Table 4*.

Parameter name	Value
C (complexity parameter)	1.0
Kernel	RBF Kernel
regOptimizer	RegSMOImproved

Table 4. SVM Configuration Parameters

3.4 EVALUATION METRICS

The accuracy of the experimental results can be evaluated based on the different metrics such as Mean Absolute Error (MAE), Root Mean Square Error (RMSE), PRED (25) and R2 Prediction Accuracy [30]. Among these metrics, PRED(25) only considers the percentage of the observations whose prediction accuracy falls within 25% of the actual value. On the other hand, R2 Prediction Accuracy is a measure of the goodness-of-fit, whose value falls within the range [0, 1] and is commonly applied to the linear regression models [30]. Due to the limitations of PRED (25) and R2 Prediction Accuracy, the MAE and the RMSE metrics are used in this paper to measure the accuracy of the prediction algorithms. The formal definitions of these metrics are [30]:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|YP_i - Y_i| \tag{9}$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(YP_i - Y_i)^2}{n}} \tag{10}$$

where YP_i is the predicted output and Y_i is the actual output for the ith observation, and n is the number of observations for which the prediction is made.

The MAE metric is a popular metric in statistics, especially in the prediction accuracy evaluation. The RMSE represents the sample standard deviation of the differences between the predicted values and the observed values. A smaller MAE and RMSE value indicates a more accurate prediction scheme.

The MAE metric is a linear score which assumes all of the individual errors are weighted equally. Moreover, the RMSE is most useful when large errors are particularly undesirable [31]. Since the large errors can significantly reduce the network QoS, the network operators prefer to use a regression model which generates a greater number of small errors rather than a regression model that generates a fewer number of the large errors. As a result, in the mobile network resource provisioning domain, the RMSE factor is more important than the MAE factor. However, considering both metrics provides a comprehensive analysis of the accuracy of the prediction models. The greater the difference between RMSE and MAE is, the greater the variance in the individual errors is in the sample.

3.5 HARDWARE CONFIGURATION

Hardware configuration influences the performance (i.e., the time required to create a regression model) of the prediction algorithms. Therefore, to eliminate the impact of the hardware configuration on the prediction results, the same hardware is used to create MLP, MLPWD, and SVM regression models. Table 5 shows the hardware configuration that is used in the experiment.

Hardware	Capacity
Memory	8 Gigabytes
Processor	Intel Core i5
Storage	2 Terabytes HDD

Table 5. Hardware configuration

3.6 EXPERIMENT AND RESULTS

The main objective of the experiment is to compare accuracy of the MLP, MLPWD, and SVM algorithms in predicting the future values of the traffic components (i.e., trend, seasonal, and remainder components). In Section 3.2 the STL method was used to decompose the network traffic into three components. The STL method decomposes the network traffic dataset into three datasets, each representing one of the traffic components. As a result, in this experiment there are three datasets (each dataset includes historical values of one of the components) and the objective is to find the most accurate prediction model to forecast each of the datasets.

Since there is only one feature in the datasets, the sliding window technique is used to train and test the prediction models. Choosing the right size for the sliding window is not a trivial task. Usually smaller window sizes do not reflect the correlation between the data points thoroughly, while using greater window size increases the chance of overfitting. Therefore, this experiment also uses different window sizes to measure the effect of the sliding window size on the prediction accuracy of the MLP, MLPWD, and SVM algorithms.

Table 6 shows the prediction accuracy of the MLP, MLPWD, and SVM algorithms to forecast the future value of the seasonal component of the network traffic. The metric values (i.e., MAE and RMSE values) are plotted in *Figure 5* and *Figure 6*, as well.

The results show that by increasing the window size, the accuracy of the regression models increases. In addition, when the window size is greater than 8 time slots the MLP algorithm can predict the exact future value of the seasonal component (c.f. *Table 6*). Because MLP uses empirical risk minimization principle, it can increase the complexity of the regression model until the regression model becomes fully tailored to the training dataset. Since the seasonal component's behavior in the testing and the training datasets is identical, fitting the model to the training dataset increases the model's accuracy to predict the testing dataset, as well. Therefore, to predict the future seasonal value of the network traffic, it is better to use the MLP regression model.

The resulting metric values for the algorithms to predict the trend component are shown in *Table 7*, *Figure 7* and *Figure 8*. In the training and the testing phases, the MLP and the SVM algorithms have similar accuracy results which increase by increasing the window size. However, increasing the window size that is greater than 4 time slots neither improves nor decreases the prediction accuracies of MLP and SVM algorithms. On the other hand, increasing the window size to greater than 4 time slots has a negative impact on the MLPWD's accuracy.

In the training phase, the SVM algorithm has slightly better accuracy compared with MLP to forecast the trend component. However, MLP has smaller RMSE value compared to SVM. This indicates that the predictions of the MLP algorithm have fewer large errors compared with the SVM algorithm. Therefore, similar to the seasonal component, it is better to use the MLP algorithm to predict the trend values of the network traffic.

Unlike the seasonal and the trend components, the remainder component does not follow a periodic pattern. The complex nature of the remainder component necessitates the prediction models to increase their VC-dimension to predict the time-series. Increasing the VC-dimension may cause the overfitting problem. The prediction accuracy of the

MLP, MLPWD, and SVM algorithms are shown in *Table 8*, *Figure 9* and *Figure 10*.

Phase	Window Size	MAE			RMSE		
		MLP	MLPWD	SVM	MLP	MLPWD	SVM
Training	2	3.93	3.51	3.79	5.37	5.07	5.39
	3	3.67	3.18	3.01	5.04	4.61	4.58
	4	0.68	2.45	3.42	0.84	3.04	4.96
	5	0.16	2.28	2.80	0.23	2.92	4.08
	6	0.005	2.21	2.54	0.025	2.76	3.89
	7	0.005	2.34	2.63	0.009	2.70	3.96
	8	0.002	2.22	2.81	0.003	2.55	4.42
	9	0.001	1.17	1.20	0.003	1.49	2.06
	10	0	1.02	1.11	0	1.38	2.06
Testing	2	5.92	3.56	3.55	7.24	4.843	4.981
	3	3.86	3.03	2.91	4.87	4.328	4.392
	4	0.41	2.08	2.98	0.567	2.633	4.572
	5	0.15	2.23	2.55	0.2	2.908	3.760
	6	0.03	2.71	2.51	0.031	3.222	3.904
	7	0.002	2.20	2.28	0.003	2.578	3.449
	8	0.002	2.02	2.05	0.002	2.307	3.577
	9	0	0.93	0.83	0	1.236	1.626
	10	0	1.02	0.77	0	1.321	1.506

Table 6. MAE and RMSE values (seasonal component)

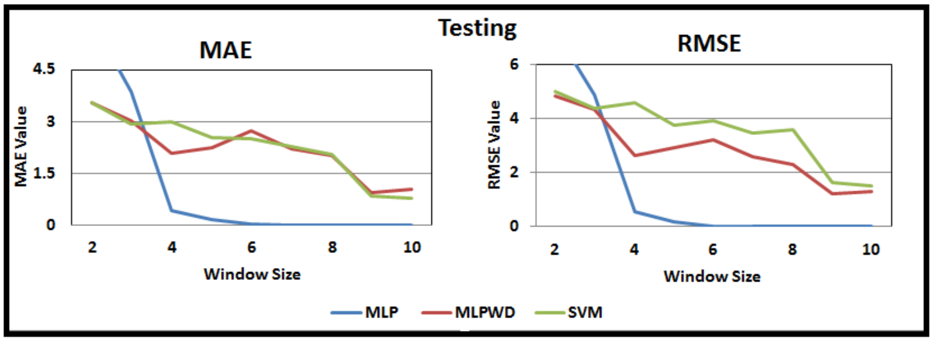

Figure 5. MAE and RMSE values in the training phase (seasonal component)

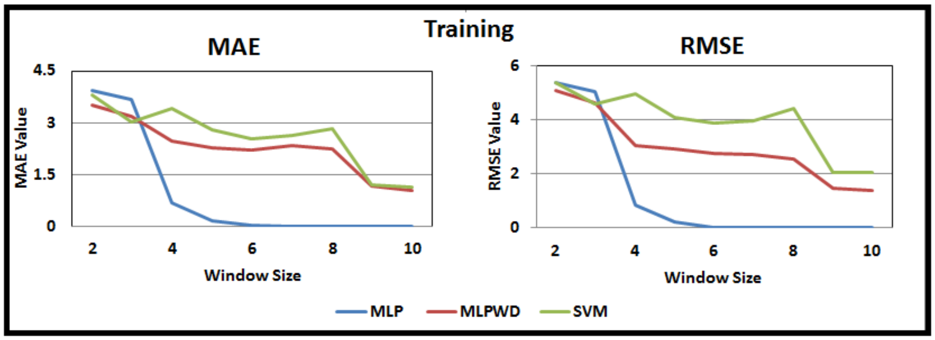

Figure 6. MAE and RMSE values in the testing phase (seasonal component)

Phase	Window Size	MAE			RMSE		
		MLP	MLPWD	SVM	MLP	MLPWD	SVM
Training	2	3.079	2.879	2.867	3.680	3.399	3.333
	3	1.036	3.15	0.863	1.291	3.686	1.044
	4	0.758	1.325	0.474	0.998	1.750	0.665
	5	0.709	0.918	0.575	0.930	1.222	0.801
	6	0.499	1.018	0.628	0.721	1.353	0.896
	7	0.887	1.219	0.615	1.116	1.624	0.874
	8	0.693	1.317	0.591	0.950	1.764	0.863
	9	0.608	1.476	0.624	0.840	2.037	0.908
	10	0.64	1.533	0.844	0.838	2.058	1.211
Testing	2	2.914	3.152	2.517	3.369	3.658	3.074
	3	1.002	3.279	1.067	1.246	3.784	1.253
	4	0.972	1.216	0.830	1.294	1.620	1.176
	5	0.881	1.295	0.993	1.148	1.845	1.443
	6	0.838	1.463	1.066	1.084	2.136	1.564
	7	0.640	1.991	1.062	0.939	2.830	1.540
	8	0.783	2.077	1.065	1.051	3.042	1.543
	9	0.653	2.211	1.100	0.867	3.290	1.621
	10	0.692	2.566	0.596	1.043	3.821	0.868

Table 7. MAE and RMSE values (trend component)

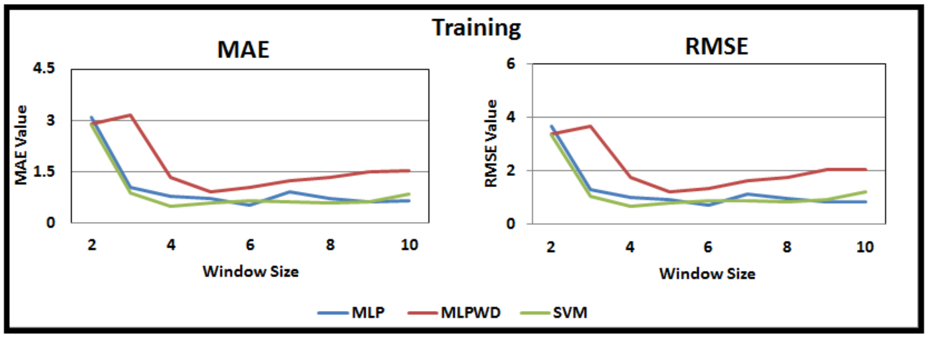

Figure 7. MAE and RMSE values in the training phase (trend component)

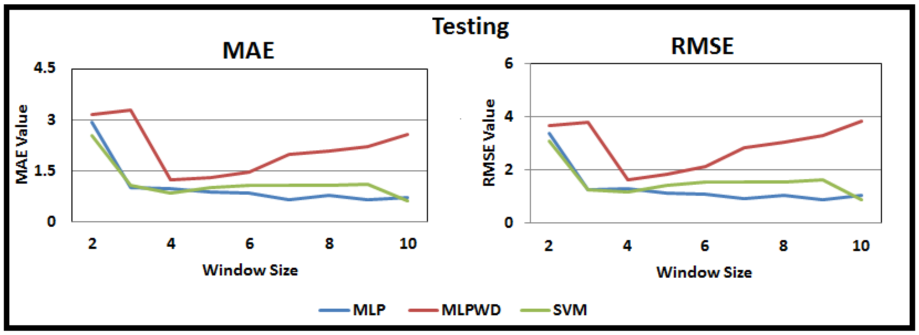

Figure 8. MAE and RMSE values in the testing phase (trend component)

Phase	Window Size		MAE			RMSE	
		MLP	MLPWD	SVM	MLP	MLPWD	SVM
Training	2	9.650	7.644	7.221	12.69	10.69	10.44
	3	9.553	7.802	7.434	12.24	10.827	10.89
	4	8.448	8.004	7.668	11.54	11.070	11.04
	5	10.10	8.053	7.957	12.59	11.178	11.11
	6	9.347	7.755	7.798	13.14	10.664	10.90
	7	9.529	7.865	7.493	12.30	10.900	10.62
	8	10.92	7.607	7.476	14.85	10.550	10.74
	9	10.59	7.552	7.709	16.38	10.178	10.75
	10	9.724	7.330	7.521	12.68	10.124	10.71
Testing	2	9.694	9.396	8.038	11.68	11.851	10.92
	3	8.270	8.64	8.242	10.74	11.167	11.20
	4	8.989	9.385	8.360	12.20	11.937	11.32
	5	9.506	8.637	8.403	13.26	11.063	11.16
	6	10.92	8.908	8.348	14.62	11.391	11.08
	7	11.59	8.601	8.052	15.53	10.938	10.90
	8	10.31	8.193	8.099	13.32	10.105	10.61
	9	10.90	8.318	8.014	15.42	10.36	10.84
	10	11.71	8.102	7.767	15.12	10.186	10.67

Table 8. MAE and RMSE values (remainder component)

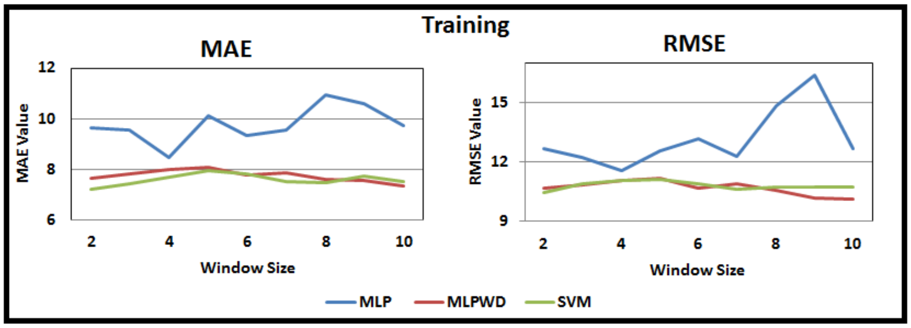

Figure 9. MAE and RMSE values in the training phase (remainder component)

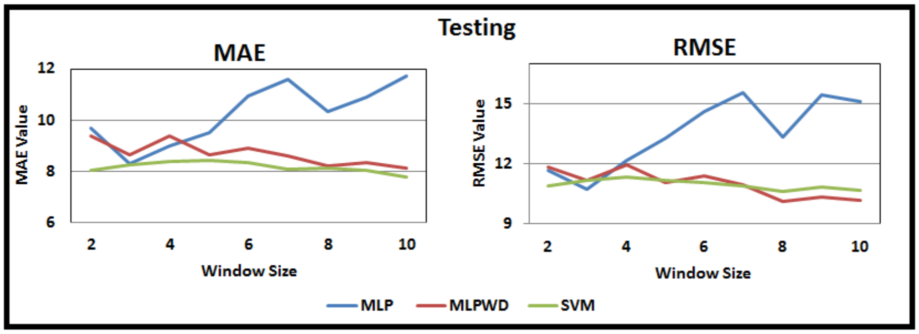

Figure 10. MAE and RMSE values in the testing phase (remainder component)

The accuracy of the MLP algorithm decreases by increasing the window size. This is because MLP uses empirical risk minimization principle and becomes overfitted to the training dataset. Because

MLPWD and SVM algorithms use structural risk minimization, their accuracies doesn't change much by increasing the window size.

According to the testing results (c.f. *Figure 10*), SVM and MLPWD algorithms have similar accuracy results. However, SVM has slightly better accuracy. Therefore, it is better to use the SVM algorithm to predict the remainder component.

The experimental results suggest using MLP algorithm to predict the trend and the seasonal components and using SVM algorithm to forecast the remainder component. In addition, increasing the window size helps to predict the seasonal and the trend components more accurately, but has no substantial impact on predicting the remainder component.

3.7 EVALUATION

The experimental results in our previous work [28] showed that the MLPWD algorithm is more accurate than the MLP and the SVM algorithm to predict the unidimensional mobile traffic. Section 3.6 suggests using MLP and SVM to use an ensemble of the prediction algorithms to predict the different components of the traffic and combine the prediction results to improve the accuracy. In the ensemble approach, the STL algorithm is used to divide the network traffic into its components. Then the MLP algorithm is used to predict the seasonal and the trend components of the traffic and SVM algorithm is used to forecast the remainder component. Finally, the prediction results of the components are aggregated to create the traffic predictions.

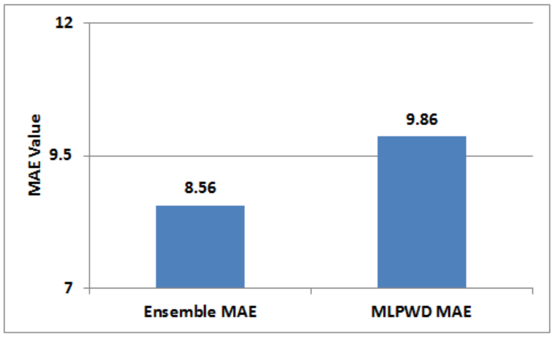

Figure 11. MLPWD and ensemble approach MAE values

This section compares the accuracy of the MLPWD algorithm with the accuracy of the ensemble approach. *Figure 11* and *Figure 12* show the comparison between the MAE and the RMSE values of the ensemble and the MLPWD algorithms to forecast the future traffic of the mobile network.

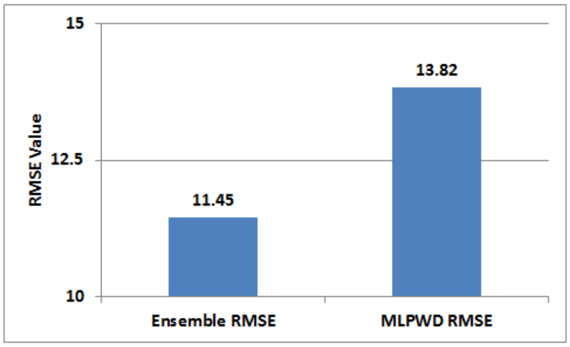

Figure 12. MLPWD and ensemble approach RMSE values

The result shows that the ensemble approach improves the accuracy of forecasting the unidimensional datasets. According to *Figure 11* and *Figure 12*, using the ensemble approach improves the MAE and the RMSE metrics by 13.16% and 17.1%, respectively.

4. CONCLUSIONS AND FUTURE WORK

The goal of this paper is to improve the prediction accuracy of the existing machine learning algorithms to forecast the unidimensional mobile network traffic datasets. To this end, this paper investigates the components of the mobile network traffic (i.e., the seasonal, trend and remainder components) and analyzes impact of each of the components on the future behavior of the network traffic. In addition, this paper compares the accuracy of the MLP, MLPWD, and SVM algorithms in predicting the future behavior of the mobile network traffic components. According to our experimental results, SVM outperforms MLPWD and MLP in predicting the remainder component of the network traffic, while MLP has better accuracy in predicting the seasonal and the trend components of the network traffic. The experimental results show using an ensemble of MLP and SVM algorithms improves the prediction accuracy of the regression models up to 17%.

This paper uses a commercial trial mobile network to carry out the experiments. The dataset includes the traffic of the network cells for one week. Although the dataset shows the seasonality and the trend of the traffic, it is not large enough to show the monthly or the yearly behavior of the traffic. Using a larger dataset in the experiments can help to identify the behavior of the traffic more accurately and prove the network operators with more information about the long term predictions of the traffic. In addition, it is important to find the dominant feature in the set of the features which can then be used to predict the future network traffic.

5. ACKNOWLEDGEMENT

The authors of this paper would like to thank Ericsson, Canada for providing various contributions and technical support. In addition, this research is partly sponsored by an NSERC (Canada Natural Sciences and Engineering Research Council) Engage grant and an NSERC Discovery grant.

6. REFERENCES

[1] Chen, M., Mao, S., and Liu, Y. (2014), Big data: A survey, Mobile Networks Application Journal, vol. 19, no. 2, pp. 171–209.

[2] Ericsson Mobile Traffic Report (2016), [Online], Available: https://www.ericsson.com/mobility-report/mobile-traffic.

[3] Laurila, J. K., Gatica-Perez, D., Aad, I., Blom, J., Bornet, O., Do, T., Dousse, O., Eberle, J., and Miettinen, M. (2012), The mobile data challenge: Big data for mobile computing research, Proceedings of the Workshop on the Nokia Mobile Data Challenge, in Conjunction with the 10th International Conference on Pervasive Computing, pp. 1–8.

[4] Nikravesh, A., Ajila, S.A., and Lung, C-H., (2016), Using MLP, MLPWD, and SVM to Forecast Mobile Network Traffic, Proceedings of IEEE International Congress on Big Data.

[5] Bankole, A.A., and Ajila, S.A., (2013), Cloud Client Prediction Models for Cloud Resource Provisioning in a Multitier Web Application Environment, IEEE 7th International Symposium. Service System Engineering, pp. 156–161.

[6] Wang, S., and Summers, R. M., (2012), Machine learning and radiology, Medical Image Analysis Journal, vol. 16, no. 5, pp. 933–951.

[7] Vapnik, V., (1992), Principles of risk minimization for learning theory, Advanced Neural Information Processing Systems Journal, pp. 831–838.

[8] Vapnik, V., and Chervonenkis, A. Y., (2013), Necessary and sufficient conditions for the uniform convergence of means to their expectations, pp. 7–13.

[9] Sewell, A., (2008), VC-Dimension, Department of Computer Science University College London.

[10] Yeh, I., Tseng, P.-Y., Huang, K.-C., and Kuo, Y.-H., (2012), Minimum Risk Neural Networks and Weight Decay Technique, Emerging Intelligence Computing Journal, pp. 10–16.

[11] Smola, J., and Schölkopf, B., (2004),A tutorial on support vector regression, Statistics Computing Journal, vol. 14, pp. 199–222.

[12] Joshi, M., and Hadi, T. H., (2015), A Review of Network Traffic Analysis and Prediction Techniques, Cornell University Library Archive, p. 23.

[13] Feng, H., and Shu, Y., (2005) Study on network traffic prediction techniques, Proceedings of International Conference on Wireless Communications, Networking and Mobile Computing, pp. 995–998.

[14] Ghanbari, H., Simmons, B., Litoiu, M., and Iszlai, G., (2011), Exploring Alternative Approaches to Implement an Elasticity Policy, IEEE 4th International Conference on Cloud Computing, pp. 716–723.

[15] Hoong, N.K., Hoong, P.K., Tan, I., Muthuvelu, N., and Seng, L.C., (2011), Impact of Utilizing Forecasted Network Traffic for Data Transfers, Proceedings of 13th International Conference on Advanced Communications, pp. 1199–1204.

[16] KuanHoong, P., Tan, I., and YikKeong, C., (2012), Gnutella Network Traffic Measurements, International Journal of Computing Networks Communication, vol. 4, no. 4.

[17] Yu, Y., Song, M., Ren, Z., and Song, J., (2011), Network Traffic Analysis and Prediction Based on APM, Proceedings of 6th International Conference on Pervasive Computing and Applications, pp. 275–280.

[18] Park, D.-C., and Woo, D.-M., (2009), Prediction of Network Traffic Using Dynamic Bilinear Recurrent Neural Network, Proceedings of 5th International Conference on Natural Computation, pp. 419–423.

[19] Junsong, E., Jiukun, W., Maohua, Z., and Junjie, W., (2009), Prediction of internet traffic based on Elman neural network, Chinese Control Decision Conference, pp. 1248–1252.

[20] Zhao, H., (2009), Multi-scale analysis and prediction of network traffic, Proceedings of the IEEE International Conference on Performance, Computing and Communications, pp. 388–393.

[21] Burney, S.M., and Raza, A., (2007), “Montecarlo simulation and prediction of Internet load using conditional mean and conditional variance model, Proceeding of the 9th Islamic Countries Conference on Statistical Sciences.

[22] Zhou, B., He, D., and Sun, Z., (2006), Traffic predictability based on ARIMA/GARCH model, Proceedings of 2nd Conference on Next Generation Internet Design and Engineering, pp. 200–207.

[23] Tang, D., and Baker, M., (2000), Analysis of a local-area wireless network, Proceedings of 6th Annual International Conference on Mobile Computing Network, pp. 1–10.

[24] Rong, C., and Esteves, R.M., (2011),, Using Mahout for clustering Wikipedia’s latest articles: A comparison between k-means and fuzzy c-means in the cloud, Proceedings of 3rd IEEE International Conference on Cloud Computing Technology Science, pp. 565–569.

[25] Esteves, R.M., Pais, R., and Rong, C., (2011), K-means clustering in the cloud - A Mahout test, IEEE Workshop of International Conference on Advanced Information Networking and Applications.

[26] Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I., (2009), The WEKA data mining software, ACM SIGKDD Explorer Newsletter, vol. 11, no. 1, p. 10.

[27] Hyndman, R., and Athanasopoulos, G., (2013), Forecasting: principles and practice, 1st edition, OTexts publisher.

[28] Nikravesh, A., Ajila, S.A., and Lung, C.-H., (2014), Measuring Prediction Sensitivity of a Cloud Auto-scaling System, Proceedings of 38th IEEE Annual International Computers, Software and Applications Conference Workshop, pp. 690–695.

[29] Hastie, T., Tibshirani, R., and Freidman, J., (2009), The Elements of Statistical Learning, 2nd edition, Springer publisher.

[30] Witten, I., Frank, E., and Hall, M., (2011), Data Mining: Practical Machine Learning Tools and Techniques, 3rd edition, Morgan Kaufmann Publisher.

[31] Chai, T., and Draxler, R., (2014), Root mean square error (RMSE) or mean absolute error (MAE) – Arguments against avoiding RMSE in the literature, Journal of Geoscience Model Deveopment, vol. 7, no. 1, pp. 1247 – 1250.

Authors

Ali Yadavar Nikravesh received the B.S. and the M.S. degrees in Software Engineering from Shahid Beheshti University, Iran and the Ph.D. degree in Computer Engineering from Carleton University, Canada. In September 2016 he joined Microsoft Corporation where he is now a Software Engineer. His research interests include: Cloud computing, machine learning algorithms, and time-series prediction.

Samuel A. Ajila is currently an associate professor of engineering at the Department of Systems and Computer Engineering, Carleton University, Ottawa, Ontario, Canada. He received B.Sc. (Hons.) degree in Computer Science from University of Ibadan, Ibadan, Nigeria and Ph.D. degree in Computer Engineering specializing in Software Engineering and Knowledge-based Systems from LORIA, Université Henri Poincaré – Nancy I, Nancy, France. His research interests are in the fields of Software Engineering, Cloud Computing, Big Data Analytics and Technology Management.

Chung-Horng Lung received the B.S. degree in Computer Science and Engineering from Chung-Yuan Christian University, Taiwan and the M.S. and Ph.D. degrees in Computer Science and Engineering from Arizona State University. He was with Nortel Networks from 1995 to 2001. In September 2001, he joined the Department of Systems and Computer Engineering, Carleton University, Ottawa, Canada, where he is now a Professor. His research interests include: Software Engineering, Cloud Computing, and Communication Networks.

Wayne Ding is a senior specialist in Ericsson, Canada. He is involved in 4G and 5G wireless system R&D. His work focuses on mobile big data and wireless network KPIs. He received the BS and MS degrees Dalhousie University, Canada.