Mobile Network Traffic Prediction Using MLP, MLPWD, and SVM

Ali Yadavar Nikravesh Samuel A. Ajila Chung-Horng Lung
Department of Systems and Computer Engineering, Carleton University
1125 Colonel By Drive, Ottawa K1S 5B6, Ontario Canada
{alinikravesh, ajila, chung}@sce.carleton.ca

Wayne Ding
LTE System, Business Unit Radio Ericsson
Ottawa, Ontario, Canada K2K 2V6
wayne.ding@ericsson.com

***Abstract*— Mobile networks are critical for today's social mobility and the Internet. More and more people are subscribing to mobile networks, which has led to substantial demands. The network operators need to find ways of meeting the huge demands. Since mobile network resources, such as spectrum, are expensive, there is a need for efficient management of network resources as well as finding a way to predict future use for network management and planning. Network planning is crucial for network operators to provide services that are both cost effective and have high degree of quality of service (QoS). The aim of this research is to apply data analysis techniques to support network operators to maximize the resource usage for network operators, that is, to prevent both under-provisioning and over-provisioning. Therefore, this paper investigates the prediction accuracy of machine learning techniques – Multi-Layer Perceptron (MLP), Multi-Layer Perceptron with Weight Decay (MLPWD), and Support Vector Machines (SVM) – using a dataset from a commercial trial mobile network. The experimental results show that SVM outperforms MLP and MLPWD in predicting the multidimensionality of the real-life network traffic data, while MLPWD has better accuracy in predicting the unidimensional data. Our experimental results can help network operators predict future demands and facilitate provisioning and placement of mobile network resources for effective resource management**

Keywords—Mobile Networks, Traffic Analysis, , Multi-Layer Perceptron, Multi-Layer Perceptron with Weight Decay, Support Vector Machine

I. Introduction

The term of big data is mainly used to describe enormous datasets [1]. Unlike traditional datasets, big data typically includes masses of unstructured data which need more real-time analysis. Moreover, big data analysis helps researchers and business analysts to gain in-depth understanding of the hidden aspects of business problems and enables them to devise better solutions for business issues [2].

The ubiquity of mobile phones and the increasing amount of data generated by mobile phone users give rise to enormous datasets which can be used to characterize and understand users' mobility, communication, and interaction patterns [3]. Hence, mobile network operators can utilize big data analysis techniques to extract information from the mobile network usage datasets. The extracted information helps mobile network operators to identify more effective solutions to manage mobile networks.

In recent years, the demand for mobile networks has dramatically increased [1] which necessitates mobile network operators to deal with 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. Under-provisioning results to saturation of network resources, which may lead to mobile users' dissatisfaction. On the other hand, over-provisioning is the result of excessive amount of network resources, which leads to the waste of valuable network resources, such as spectrum. To prevent under-provisioning and over-provisioning, mobile network operators can use a prediction system to provide insights into the future behaviors of the mobile network traffic and help network operators to accommodate future traffic demands more efficiently.

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 virtual machines (VM) provisioning in cloud resource management to the prediction of network resource usage by using a real life dataset from a commercial trial mobile network. According to [4], the most dominant prediction technique for cloud resource provisioning is time-series analysis. Since cloud and network resource provisioning areas are similar in essence, in this paper we investigate the accuracy of time-series prediction algorithms in mobile network traffic prediction systems.

According to [5], Support Vector Machines (SVM) and Artificial Neural Networks (ANN) are effective algorithms to predict future system characteristics. Therefore, in this paper we compare ANN and SVM algorithms to verify their accuracy in predicting future behavior of mobile network traffic. Moreover, there are different forms of ANN. This paper specifically evaluates the accuracy of Multi-Layer Perceptron (MLP) and Multi-Layer Perceptron with Weight Decay (MLPWD) algorithms from the ANN family. The objective of this paper is to compare the accuracy of MLP, MLPWD, and SVM algorithms in predicting mobile network traffic using data supplied by a commercial mobile network.

The main difference between MLP and MLPWD algorithms is the risk minimization approach that is being used to create MLP and MLPWD regression models. Therefore, comparing MLP with MLPWD shows the impact of the risk minimization approach on the prediction accuracy of neural networks. See Section II.A for more details on the risk minimization approach and the MLP and MLPWD algorithms. Furthermore, comparing MLPWD and MLP with SVM can help to evaluate neural networks against vector machines and

978-1-5090-2622-7/16 $31.00 © 2016 IEEE
DOI 10.1109/BigDataCongress.2016.63

highlights the capability of each to predict mobile network traffic usage accurately. The contributions of this work are:

- Comparing the accuracy of MLP, MLPWD, and SVM algorithms to predict future behavior of mobile network traffic.
- Analyzing the impact of the sliding window size on the prediction accuracy of MLP, MLPWD, and SVM algorithms.
- Comparing the accuracy of MLP, MLPWD, and SVM algorithms in predicting multidimensional and unidimensional mobile network traffic datasets.

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

II. Background and Related Work

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

A. Machine Learning

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

Vapnik indicated that machine learning corresponds to the problem of function approximation [7]. Based on this definition, the machine learning regression goal 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 functional 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 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:

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

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 proved 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. We have used the Java implementation of these algorithms in WEKA tool to carry out the experiment. Each of the three algorithms is highlighted as follows:

1) Multi-Layer Perceptron (MLP)

MLP is a feed-forward Artificial Neural Network 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:

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

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 problems. 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. Eqa. (4) is a simplified version of Eqa. (2) and represents the ERM. In other words, MLP uses the ERM approach to create its regression model.

2) Multi-Layer Perceptron with Weight Decay (MLPWD)

Unlike MLP, MLPWD algorithm uses 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 hierarchy of nested subsets in order of increasing complexity. The authors of [7] suggest three examples of structures build for the set of functions implemented by neural networks.

- Structure given by the architecture of the neural network.
- Structure given by the learning procedure
- Structure given by preprocessing

The second proposed structure (i.e., given by the learning procedure) uses "weight decay" to create 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. 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 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:

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

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 weights from growing too large. The risk minimization equation for Multi-Layer Perceptron with Weight Decay (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 Eqa. (4), w represents weights in the neural network, and λ is the penalty coefficient of the sum of squares of weights.

The authors of [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 MLPWD algorithm to study the accuracy of neural networks using structural risk minimization in predicting mobile network traffic.

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 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.

B. Network resource provisioning approaches

Different researchers have investigated network traffic analysis and prediction. The ultimate goal of network traffic analysis is to get insight into the type of network packets and the data flowing through a network. Network traffic analysis involves network data preprocessing, analysis (i.e., data mining), and evaluation. The achieved insight through the network traffic analysis is primarily used for security management purposes [12].

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

Linear time series technique 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 noise from the time-series. In addition, results of [14] show, 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 is 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 traffics. ANN is the most popular non-linear models that is used in the existing research works to predict network traffic data [18][19][20]. ANN has different variations. MLP and MLPWD (i.e., two popular variations of ANN) are introduced in Section II.A.

Hybrid model techniques are 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 concluded that ANN outperforms other models.

To the best our knowledge no research work has been published that looks at the prediction of a commercial network traffic 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 many mobile network resources is mostly static, hence the network cannot dynamically adapt to traffic change well. Often, the anticipated worst case scenario is considered in the lack of dynamic adaptation, which mostly results in over-provisioning and, hence, waste of resources.

III. Experiments and Results

The goal of this experiment is to compare the accuracy of MLP, MLPWD, and SVM prediction algorithms for network resource provisioning purposes. This section presents the experimental steps and the results of this research.

A. *Data preparation and cleaning*

The experiment was carried out 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 one specific cell 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 our experiment. This resulted in a new dataset with 175 rows (i.e., the selected network cell had 175 network traffic samples). Moreover, removing duplicated rows reduced the dataset size to 168 rows.

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

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

We used *CorrelationAttributeEval* for the attribute evaluator and *Ranker* for the search method. TABLE I represents *CorrelationAttributeEval* configuration parameters and their values.

TABLE I. CorrelationAttributeEval Configuration Parameters

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 II shows the attribute names, description, and correlation of the data used for the analysis. Please note that attributes' real names were replaced by codes due to the information disclosure reasons.

TABLE II. Correlation Attributes Ranking

Attribute Name	**Description**	**Correlation**
X1	PDCP signaling radio bearers (SRB) volume at downlink	0.0678
X2	Total UEs scheduling time at uplink	0.0667
X3	Sum of radio resource control connections	0.0667
X4	Maximum radio resource control connection	0.0667
X5	PDCP signaling radio bearers (SRB) volume at uplink	0.0666
X6	PDCP latency time at downlink	0.066
X7	The aggregated scheduling time per cell at uplink	0.0658
X8	Total UEs scheduling time at downlink	0.0657
X9	PDCP data radio bearers (DRB) volume at downlink	0.0656
X10	The aggregated scheduling time per cell at downlink	0.0655
X11	Total number of packets for latency measurement at downlink	0.0655
X12	Total number of active user equipment at downlink	0.0653
X13	PDCP packets received at uplink	0.0653
X14	PDCP packets received at downlink	0.0652
X15	Aggregated transport time over UEs at downlink	0.0648
X16	Active user equipment at uplink	0.0641
X17	Minimum PDCP data radio bearers (DRB) bit rate at uplink	0.0639
X18	PDCP DRB volume at downlink for the last transmission time interval	0.0637
X19	Aggregated transport time over UEs at uplink	0.0629
X20	Maximum PDCP data radio bearers (DRB) bit rate at uplink	0.0613
X21	Maximum PDCP data radio bearers (DRB) bit rate at downlink	0.0601
X22	Minimum PDCP data radio bearers (DRB) bit rate at downlink	0.0553
X23	PDCP data radio bearers (DRB) volume at uplink	0.0537
X24	Aggregated transport time over UEs at uplink	0.042
X25	Number of objects in measurement	0
X26	Sample of radio resource control connections	0
X27	Network cell ID	0

According to TABLE II, the last three attributes *X25*, *X26*, and *X27* are not correlated to the rest of the attributes and are not useful for machine learning model construction purpose. Therefore, to create machine learning models, the last three attributes (i.e., *X25, X26,* and *X27*) should be discarded from the dataset. Removing the aforementioned attributes, results in a new dataset with 24 dimensions or attributes.

B. *Training and Testing MLP, MLPWD, and SVM*

After the "data preparation and cleaning" step, our dataset has 168 network traffic data points, and each data point has 24 dimensions (i.e., attributes). Network traffic data 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 [27] we proved that the optimum training duration for neural networks and SVM algorithms is 60% of the experiment duration. Therefore, we considered 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.

Another important factor in training and testing time-series prediction algorithms is the number of data features. Machine learning algorithms predict future value of a time-series by discovering relations between features of the historical data and using the discovered relations to predict future values. After the data preparation step, our dataset has 24 attributes. To perform prediction, 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 realistic to use them to predict future value of the target class. Therefore, we have investigated the prediction accuracy of MLP, MLPWD, and SVM in two experiments:

- **Experiment I:** comparing the accuracy of MLP, MLPWD, and SVM, assuming that the future values of all the attributes except the target class are known.
- **Experiment II:** comparing the accuracy of MLP, MLPWD, and SVM, assuming that the future values of all the attributes are unknown.

In the first experiment, training and testing datasets include 24 attributes (or features) and in the second experiment training and testing datasets include only one attribute. In the second experiment, sliding window technique was 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 future value of that feature. For example, to predict value of b_{k+1}, sliding window technique uses [b_1, b_2, …, 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.

To prevent the over-fitting problem a cross-validation technique was used in the training phase. Readers are encouraged to see [28] for more details about the cross-validation technique. Table III presents configuration of MLP and MLPWD algorithms in our experiments. Configuration of SVM is shown in TABLE IV.

TABLE III. MLP AND MLPWD CONFIGURATION PARAMETERS

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	(attributes + classes) / 2	(attributes + classes) / 2
Decay	False	True

TABLE IV. SVM CONFIGURATION PARAMETERS

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

C. Evaluation Metrics

The accuracy of the experimental results can be evaluated based on different metrics such as Mean Absolute Percentage Error (MAPE), Root Mean Square Error (RMSE), PRED (25) and R2 Prediction Accuracy [29]. Among these metrics, PRED(25) only considers percentage of observations whose prediction accuracy falls within 25% of the actual value. In addition, R2 Prediction Accuracy is a measure of goodness-of-fit, which its value falls within the range [0, 1] and is commonly applied to linear regression models [29]. Due to the limitations of PRED (25) and R2 Prediction Accuracy, we used MAPE and RMSE metrics in this work. Formal definitions of these metrics are:

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\frac{|YP_i - Y_i|}{Y_i} \tag{7}$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(YP_i - Y_i)^2}{n}} \tag{8}$$

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

MAPE usually expresses the accuracy as a percentage and is a popular metric in statistics, especially in trend estimation. RMSE represents the sample standard deviation of the differences between predicted values and observed values. A smaller MAPE and RMSE value indicate a more effective prediction scheme.

D. Hardware Configuration

Hardware configuration influences the performance (i.e., the time required to create regression model) of the prediction algorithms. Therefore, to eliminate impact of the hardware configuration on the prediction results, the same hardware was used to create MLP, MLPWD, and SVM regression models. TABLE V shows the hardware configuration used in our experiments.

TABLE V. HARDWARE CONFIGURATION

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

E. Experiment I

In this experiment training and testing datasets include 24 features. To perform the experiment, one of the features should be selected as the target class. Since all the features follow the same periodic pattern, we selected the *X12* feature as the target class. Feature *X12* is the number of active users 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 crucial parameter for network operators.

- Most of the features in the dataset have a strong correlation with feature *X12*.

Figure 1 illustrates MLP, MLPWD, and SVM prediction results against the actual values of X*12* feature.

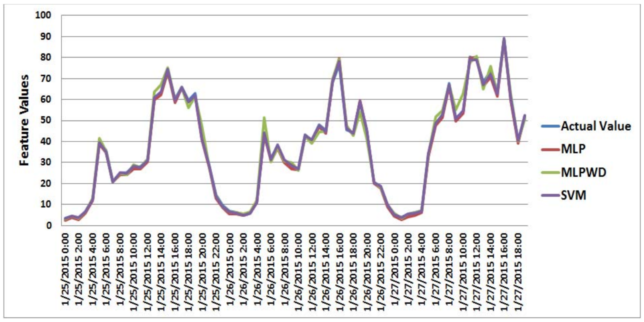

Fig. 1. Prediction results (Experiment I)

As shown in Figure 1, there is no statistical difference between SVM, MLP, and MLPWD prediction results. Figure 2 and Figure 3 illustrate MAPE and RMSE values for MLP, MLPWD, and SVM in the training phase.

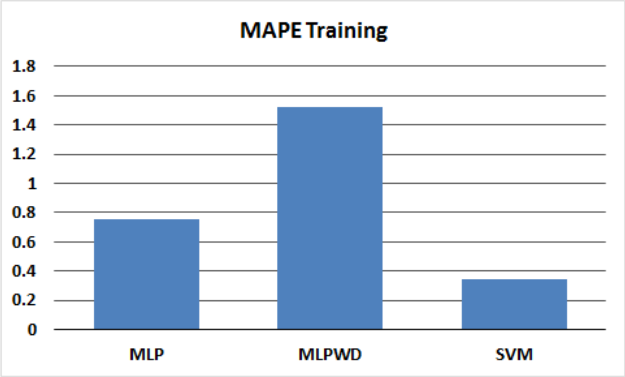

Fig. 2. MAPE values of MLP, MLPWD, and SVM in the training phase (Experiment I)

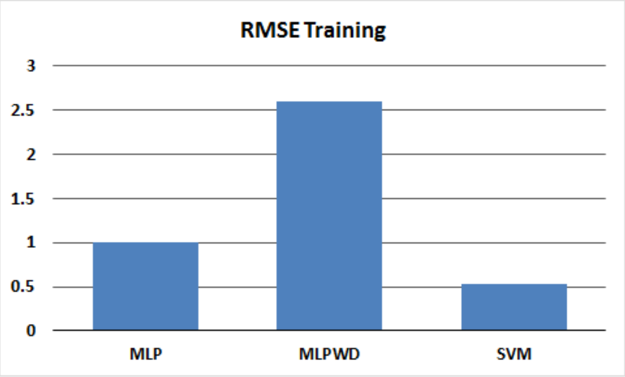

Fig. 3. RMSE values of MLP, MLPWD, and SVM in the training phase (Experiment I)

Figure 2 and Figure 3 show that in the training phase, MLP has less functional risk compared to MLPWD in the regression model construction process. According to the training results, to construct the regression model, MLPWD algorithm neglects some complexities of the data to control the VC-dimension (see Section II.A.2). Figure 4 and Figure 5 represent MAPE and RMSE values for MLP, MLPWD, and SVM in the testing phase.

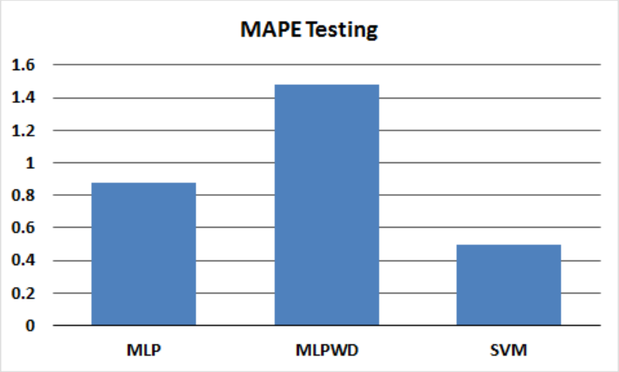

Fig. 4. MAPE values of MLP, MLPWD, and SVM in the testing phase (Experiment I)

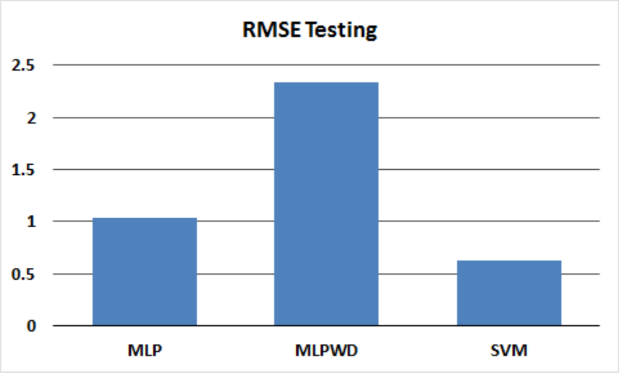

Fig. 5. RMSE values of MLP, MLPWD, and SVM in the testing phase (Experiment I)

According to Figure 4 and Figure 5, SVM outperforms MLP and MLPWD in predicting the *X12* feature when future values of the rest of the attributes are known. In other words, our results show that for multidimensional datasets (i.e., datasets that include more than one feature) SVM has better accuracy compared to MLP and MLPWD.

Since MLP is more accurate than MLPWD, it can be concluded that for multidimensional datasets ERM outperforms SRM. This is because in the environments with many features, training and testing datasets are complex (i.e., greater VC-dimension value) and according to [7] in environments with greater VC-dimension values, ERM performs better than SRM. However, in this experiment SVM (based on SRM, see Section II.A) shows a better accuracy compared to MLP and MLPWD, which indicates that support vectors can identify relations between features in multidimensional environments significantly better than the neural networks.

F. Experiment II

In this experiment, the future values of the features are unknown. Therefore, training and testing datasets include only one feature which is the target class. The same target class that was investigated in experiment I (i.e., *X12*) was selected as the target class in this experiment as well.

Since there was only one feature in the dataset, sliding window technique was 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 correlation between data points thoroughly, while using greater window size increases the chance of overfitting. Therefore, in addition to the main goal of the experiment, in this experiment we used different window sizes to measure the effect of sliding window size on the prediction accuracy of MLP, MLPWD, and SVM algorithms. Figure 6 shows prediction results against the actual *X12* values using MLP, MLPWD, and SVM.

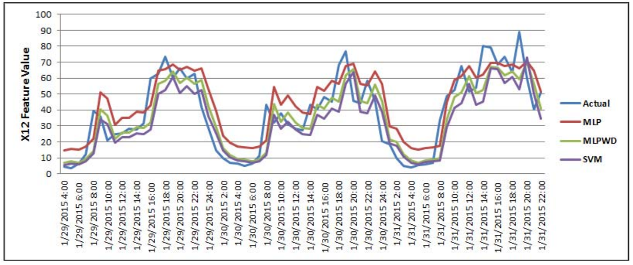

Fig. 6. Prediction results for sliding window size = 2 (Experiment II)

Figure 6 demonstrates that MLPWD and SVM are more accurate than MLP when the sliding window with size of 2 timeslots is used. TABLE VI and TABLE VII illustrate the experimental results.

TABLE VI. MAPE AND RMSE VALUES (EXPERIMENT II – TRAINING)

Window Size	MAPE			RMSE		
	MLP	MLPWD	SVM	MLP	MLPWD	SVM
2	12.02	9.2639	10.17	14.037	12.3679	13.859
3	10.03	9.4115	10.21	12.434	12.4089	13.843
4	10.07	9.345	10.13	13.085	12.7016	13.683
5	11.98	9.3671	10.2	15.618	12.5739	13.686
6	12.66	9.4642	9.892	16.545	12.4906	13.177
7	15.07	9.1575	9.722	20.804	12.0542	13.138
8	11.86	9.2812	9.516	15.188	12.1036	12.826
9	19.97	9.0478	9.672	23.039	11.9403	13.276
10	16.87	8.9453	9.631	23.393	11.92	13.317

The results in TABLE VI, TABLE VII, Figure 7, and Figure 8 show that MLPWD outperforms MLP and SVM in almost all scenarios when the sliding window technique is used. In addition, increasing the window size does not have much impact on the accuracy of MLPWD and SVM. The reason is that SVM and MLPWD use SRM, thus increasing the window size does not increase their VC-dimension and their accuracy is not affected. However, since MLP uses ERM, increasing the window size increases its VC-dimension which has negative effect on its accuracy. Furthermore, MLPWD has slightly better accuracy compared to SVM, which indicates that neural networks outperform vector machines in capturing relations when sliding window technique is being used.

TABLE VII. MAPE AND RMSE VALUES (EXPERIMENT II – TESTING)

Window Size	MAPE			RMSE		
	MLP	MLPWD	SVM	MLP	MLPWD	SVM
2	11.712	9.6893	9.8702	15.725	13.0964	14.086
3	10.965	9.8791	10.257	14.451	13.3479	14.399
4	10.168	9.582	10.479	13.763	13.1692	14.63
5	10.257	9.5024	10.402	13.916	13.1467	14.539
6	11.118	9.4307	9.6997	15.942	12.9783	13.526
7	10.926	9.3973	9.3097	14.821	13.0807	13.264
8	12.11	9.7438	9.6887	16.856	13.5266	13.566
9	11.664	9.7524	9.8191	15.661	13.4867	13.720
10	13.838	9.8596	9.6293	19.51	13.8235	13.7582

Figure 7 and Figure 8 compare the MAPE and RMSE values of MLP, MLPWD, and SVM in the testing phase.

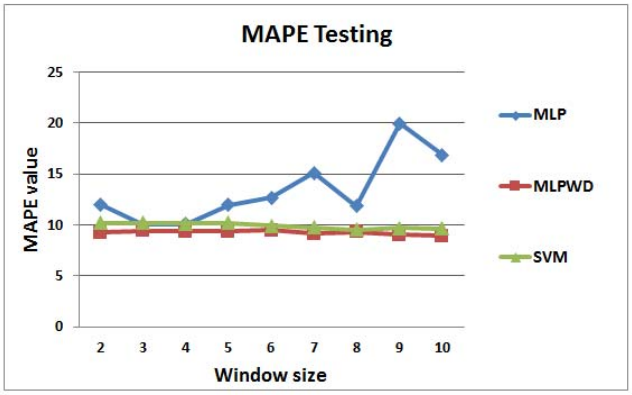

Fig. 7. MAPE values of MLP, MLPWD, and SVM in the testing phase (Experiment II)

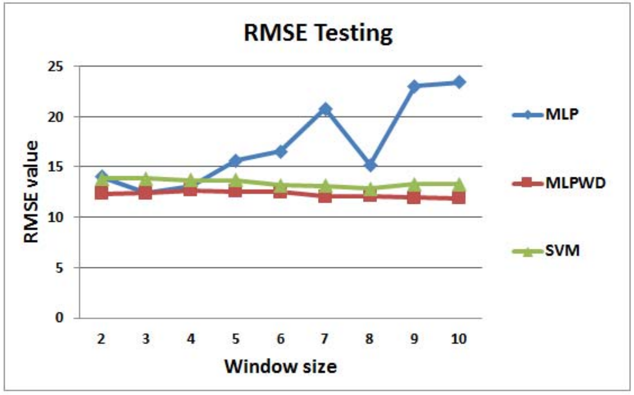

Fig. 8. RMSE values of MLP, MLPWD, and SVM in the testing phase (Experiment II)

This experiment investigates the accuracy of MLPWD, MLP, and SVM regression models to forecast future values of *X12* feature. However, because all of the features in the dataset (i.e., features *X1* to *X24*) follow similar periodic pattern, and since the *X12* feature is correlated to the rest of the features in the dataset (c.f. TABLE II), the experimental results can be extended to the rest of the features. Therefore, it can be concluded that MLPWD outperforms MLP and SVM algorithms to predict unidimensional mobile network traffic datasets.

IV. CONCLUSION AND FUTURE WORK

In this paper we have investigated the accuracy of MLP, MLPWD, and SVM algorithms in predicting future behavior of mobile network traffic. According to our experimental results, dimensionality of the traffic data can affect the prediction accuracy of the regression models. Based on our results, SVM outperforms MLPWD and MLP in predicting the multidimensionality of the real-life traffic data, while MLPWD has better accuracy in predicting the unidimensional data. In addition, the MAPE and RMSE values, as depicted in Figure 4 and Figure 5, as well as TABLE VII, indicate that using multidimensional traffic datasets significantly increases the prediction accuracy of MLP, MLPWD, and SVM algorithms. Therefore, it can be concluded that to increase the precision of prediction results, multidimensional network traffic data as well as SVM algorithm should be used. However, if dependent/independent variables do not exist (i.e., the traffic dataset is unidimensional), MLPWD together with the sliding window technique are the best combination to predict future network behavior.

Since it is nontrivial to carry out prediction in a multidimensional data, especially when the size is large, one of the future works in this research is to investigate the data by using a parallel algorithm that will handle effectively the multidimensionality of the dataset to facilitate real time data analytics. In addition, it is important to find the dominant feature in the set of features which can then be used to predict future network traffic.

Acknowledgment

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 Engage grant and an NSERC Discovery grant.

REFERENCES

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

[2] H. Chen, R. H. L. Chiang, and V. C. Storey, "Business Intelligence and Analytics: From Big Data To Big Impact," *Mis Q.*, vol. 36, no. 4, pp. 1165–1188, 2012.

[3] J. Laurila, D. Perez, I. Aad, J. Blom, O. Bornet, T. Do, O. Dousse, J. Eberle, and M. Miettinen, "The mobile data challenge: Big data for mobile computing research," in *Proceedings of the Workshop on the Nokia Mobile Data Challenge,* 2012, pp. 1–8.

[4] T. Lorido-Botran, J. Miguel-Alonso, and J. A. Lozano, "A Review of Auto-scaling Techniques for Elastic Applications in Cloud Environments," *J. Grid Comput.*, vol. 12, no. 4, pp. 559–592, 2014.

[5] A. a. Bankole and S. a. Ajila, "Cloud Client Prediction Models for Cloud Resource Provisioning in a Multitier Web Application Environment," *2013 IEEE Seventh Int. Symp. Serv. Syst. Eng.*, pp. 156–161, Mar. 2013.

[6] S. Wang and R. M. Summers, "Machine learning and radiology," *Med. Image Anal.*, vol. 16, no. 5, pp. 933–951, 2012.

[7] V. Vapnik, "Principles of risk minimization for learning theory," *Adv. Neural Inf. Process. Syst.*, pp. 831–838, 1992.

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

[9] M. Sewell, "VC-Dimension," *Dep. Comuter Sci. Univ. Coll. London*, 2008.

[10] I. Yeh, P.-Y. Tseng, K.-C. Huang, and Y.-H. Kuo, "Minimum Risk Neural Networks and Weight Decay Technique," *Emerg. Intell. Comput. …*, pp. 10–16, 2012.

[11] a J. Smola and B. Schölkopf, "A tutorial on support vector regression," *Stat. Comput.*, vol. 14, pp. 199–222, 2004.

[12] M. Joshi and T. H. Hadi, "A Review of Network Traffic Analysis and Prediction Techniques," p. 23, 2015.

[13] H. Feng and Y. Shu, "Study on network traffic prediction techniques," in *Proceedings. 2005 International Conference on Wireless Communications, Networking and Mobile Computing, 2005.*, 2005, vol. 2, no. 3, pp. 995–998.

[14] H. Ghanbari, B. Simmons, M. Litoiu, and G. Iszlai, "Exploring Alternative Approaches to Implement an Elasticity Policy," in *2011 IEEE 4th International Conference on Cloud Computing*, 2011

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

[16] P. KuanHoong, I. Tan, and C. YikKeong, "G Nutella N Etwork T Raffic M Easurements and," *Int. J. Comput. Networks Commun.*, vol. 4, no. 4, 2012.

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

[18] D.-C. Park and D.-M. Woo, "Prediction of Network Traffic Using Dynamic Bilinear Recurrent Neural Network," in *2009 Fifth International Conference on Natural Computation*, 2009, pp. 419–423.

[19] W. Junsong, W. Jiukun, Z. Maohua, and W. Junjie, "Prediction of internet traffic based on Elman neural network," *2009 Chinese Control Decis. Conf. CCDC 2009*, no. 2, pp. 1248–1252, 2009.

[20] H. Zhao, "Multiscale Analysis And Prediction Of Network Traffic," in *Proceedings of the IEEE International Conference on Performance, Computing and Communications*, 2009, pp. 388–393.

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

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

[23] D. Tang and M. Baker, "Analysis of a local-area wireless network," *Proc. 6th Annu. Int. Conf. Mob. Comput. Netw. - MobiCom '00*, vol. 200, no. i, pp. 1–10, 2000.

[24] C. Rong and R. M. Esteves, "Using Mahout for clustering Wikipedia's latest articles: A comparison between k-means and fuzzy c-means in the cloud," *Proc. - 2011 3rd IEEE Int. CloudCom*, pp. 565–569, 2011.

[25] R. M. Esteves, R. Pais, and C. Rong, *K-means clustering in the cloud - A Mahout test*. 2011.

[26] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA data mining software," *ACM SIGKDD Explor. Newsl.*, vol. 11, no. 1, p. 10, 2009.

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

[28] T. Hastie, R. Tibshirani, and J. Freidman, *The Elements of Statistical Learning*, no. 2. 2009.

[29] I. H. Witten, E. Frank, and M. a. Hall, *Data Mining: Practical Machine Learning Tools and Techniques, Third Edition*, vol. 54, no. 2. 2011.