Location-aware Load Prediction in Edge Data Centers

Chanh Nguyen Le Tan, Cristian Klein, and Erik Elmroth

Department of Computing Science, Umeå University, Sweden
Email: (chanh, cklein, elmroth)@cs.umu.se

***Abstract*—Mobile Edge Cloud (MEC) is a platform complementing traditional centralized clouds, consisting in moving computing and storage capacity closer to users – e.g., as Edge Data Centers (EDC) in base stations – in order to reduce application-level latency and network bandwidth. The bounded coverage radius of base station and limited capacity of each EDC intertwined with user mobility challenge the operator's ability to perform capacity adjustment and planning. To face this challenge, proactive resource provisioning can be performed. The resource usage in each EDC is estimated in advance, which is made available for the decision making to efficiently determine various management actions and ensure that EDCs persistently satisfies the Quality of Service (QoS), while maximizing resource utilization.**

In this paper, we propose location-aware load prediction. For each EDC, load is not only predicted using its own historical load time series – as done for centralized clouds – but also those of its neighbor EDCs. We employ Vector Autoregression Model (VAR) in which the correlation among adjacent EDCs load time series are exploited. We evaluate our approach using real world mobility traces to simulate load in each EDC and conduct various experiments to evaluate the proposed algorithm. Result shows that our proposed algorithm is able to achieve an average accuracy of up to 93% on EDCs with substantial average load, which slightly improves prediction by 4.3% compared to the state-of-the-art approach. Considering the expected scale of MEC, this translates to substantial cost savings e.g., servers can be shutdown without QoS violation.

***Index Terms*—Workload Prediction, Proactive Resource Management, Mobile Edge Cloud, VAR Model, User Mobility.**

I. Introduction

The rapid deployment of mobile technology and Internet of Things (IoTs) has lead to an increasing number of real-time interactive applications with clear geographic locality, such as industrial process control, augmented reality context recognition, coordinated guiding of self-driving cars at traffic hotspots, etc. In contrast to traditional cloud applications, such applications may have strong requirement on short response time and typically produce data that is only of local interest. These are both argument for having such applications deployed in a highly distributed fashion with capacity allocated close to end-users, following closely their mobility. To this end, Mobile Edge Clouds (MECs) are emerging, aimed at reducing latency and data transfers through geographic distribution of the cloud, so that computations and data processing is not only performed at centralized large-scale data centers but also at the edge of the network [1].

In MEC, Edge Data Centers (EDC) with heterogeneous scales and costs are distributed in the proximity of users, such as in radio base stations or WiFi access points, enabling MECs to provide services with higher bandwidth and lower latency than would be possible with centralized cloud. Furthermore, MECs also bring users other appealing benefits, such as the ability to run locally-targeted services on EDCs which are closely-coupled to the radio network, require guaranteed robust or low-latency communication, send a lot of data from end-user devices, require analysis of enormous amounts of data immediately after the data was captured. EDCs can also help minimize the central cloud's ingress bandwidth requirements, and bring flexibility in deploying and removing network functions in response to the demands of the users. In fact, many industrial and academic projects target MEC to harness these tremendous benefits. For example, Nokia and its partners [2] are developing the communication framework for vehicals to vehicals and vehicals to infrastructure aimed at reducing the traffic accidents caused by human error.

Nonetheless, the bounded coverage radius of radio base stations and limited capacity of EDC intertwined with the high degree of user mobility bring new challenges to managing the resources of EDCs. As users move across different geographical areas, they should ideally connect to the nearest EDC. This in turn causes large variations in resource demand at each EDC. An EDC's load can feature large variation, e.g., in some periods, there are a large wave of users concurrently using an application on a single EDC, whereas in other EDCs only a few users interact with the application. Over-provisioning resource to an application incurs unnecessary costs, whereas under-provisioning leads to poor Quality of Service (QoS) and user experience. The question is then how to efficiently adjust capacity in each EDC so as to guarantee the QoS of hosted applications, while at the same time maximize the resource utilization. Unused servers should be turned off, extra resource should be allocated/released to accommodate peak/idle periods accordingly. Since some of these resource management actions take time to enact, proactive capacity adjustments should be performed to decide in advance the right amount of capacity to allocate to each application in each EDC. This, in turn, requires predicting future capacity demand. EDCs located close to each other might present some correlation in load variation which may help to make more accurate predictions. Most existing approaches for resource management focus on centralized clouds and do not considered the impact of user mobility as well as information about locations of data centers, hence they might miss an opportunity for more accurate predictions in MEC.

978-1-5386-2859-1/17/$31.00 ©2017 IEEE

We propose a novel location-aware load prediction approach to forecast the load at each EDC. Our approach utilizes historical load information of the EDC itself and the EDCs in its vicinity.

The **key contributions** of this paper are:

- We propose an algorithm based on Vector AutoRegression (VAR) to predict the load in each EDC (see Section III).
- We evaluate the accuracy of our method based on several error metrics and compare it to the state of the art location-unaware load prediction. We simulate an MEC as a hexagonal grid distributed across a geographical area, in which each EDC provides service to users in its vicinity. For user mobility, we use real mobility traces of taxis in San Francisco (see Section IV).

The results show that the proposed approach improved prediction accuracy by 4.3% when compared to state-of-the-art approaches. This allows a more efficient utilization of resource, e.g., by turning servers off for longer periods of time. Considering the large scale of MEC, this translates to significant cost savings to MEC operators.

II. Problem Definition

We consider an MEC composed of EDCs which are distributed across a geographical area. Each EDC is located with a radio base station which provides cloud-based services to subscribers in its physical proximity. As users frequently change their location over time, resource usage in any EDC features temporal variation.

Let $y_{i|t}$ be the time series characterizing historical load in EDC i collected up to time t. The goal is to estimate the load in each EDC at time $t+1$, e.g., $y_{i|t+1}$. The look-ahead time is short, i.e., a single time interval, so as to support planning for capacity adjustments in the very near future. The user mobility likely leads to correlations in the load of each EDC in the close vicinity, which is a latent information that can be used to improve prediction accuracy. In the following section, we propose employing the VAR model to build a load prediction model that utilizes such knowledge.

III. Location-aware Load Prediction Algorithm

Let $Y_t = (y_{1|t}, ..., y_{(n+1)|t})'$ be a $((n+1)\times 1)$ dimensional time series vector formed by combining every single time series containing the load observations at each time interval of EDC_i and its n neighbors. A VAR based process order p featured Y_t is represented by the following equation [3]:

$$Y_t = \phi_0 + \sum_{i=1}^{p} \phi_i Y_{t-i} + \epsilon_t \quad (1)$$

where ϕ_0 is a $n+1$ dimensional constant vector, and $\phi_i, i > 0$ are $((n+1)\times(n+1))$ coefficient matrices that have to be estimated in which $\phi_p \neq 0$, and ϵ_t is an $((n+1))\times 1)$ white noise vector process with time invariant covariance matrix Σ and mean zero.

Typically, to build an adequately fitted VAR model which is able to predict the load for EDCs, we need to go through a sequence of three stages (Figure 1): 1) Specifying and estimating parameters; 2) Model checking; 3) Prediction.

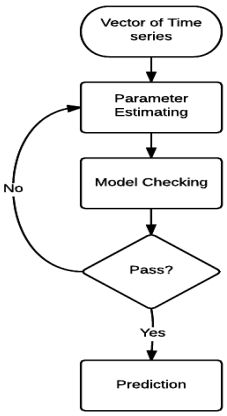


Fig. 1: VAR analysis three stages.

1) Specifying and estimating parameters: To estimate the coefficient matrices ϕ_i, several approaches such as Least Square or Maximum Likelihood methods can be employed separately to every individual equation decomposed from eq. (1).

In order to determine the lag order p, one can employ one of three common information criteria: Akaike (AIC), Schwarz-Bayesian (BIC) and Hannan-Quinn (HQ) which specified by:

$$AIC(p) = ln \mid \tilde{\Sigma}(p) \mid + \frac{2}{T} pk^2$$
$$BIC(p) = ln \mid \tilde{\Sigma}(p) \mid + \frac{lnT}{T} pk^2$$
$$HQ(p) = ln \mid \tilde{\Sigma}(p) \mid + \frac{2lnlnT}{T} pk^2$$

where $\tilde{\Sigma}(p) = T^{-1} \sum_{t=1}^{T} \hat{\epsilon}_t \hat{\epsilon}_t{}'$ is the residual covariance matrix estimator from a VAR(p), T is number observations in time series input and k is number dimensions of VAR. Each criteria may select different set of lag orders p which can be employed to build starting models for a multivariate time series.

Basically, a positive integer number p_{max} is chosen as the maximum possible lag order that can be selected for the model, for example $p_{max} = 8$. Depending on the sample size, one can choose an efficient information criterion to avoid over-fitting the model. Once an information criterion is selected, an order $p \in [0, p_{max}]$ can be chosen that minimizes the criterion.

2) Model checking: The predefined p_{max} is critical and impacts the selected order p, hence influencing the fitness of the model. To check that the fitted model with chosen lag order p is an adequate one, we need to evaluate whether its residuals behave like a white noise series and all fitted parameters are statistically significant. We hence calculate cross-correlations of the residuals by applying the standard multivariate Portmanteau statistics approach [4]. The detailed description of this technique can be found at [3].

3) Prediction: If the fitted model shows to be satisfactory to the model checking conditions, it can be used for predicting the future load in each EDC. The prediction is computed for $h = 1$ look-ahead step.

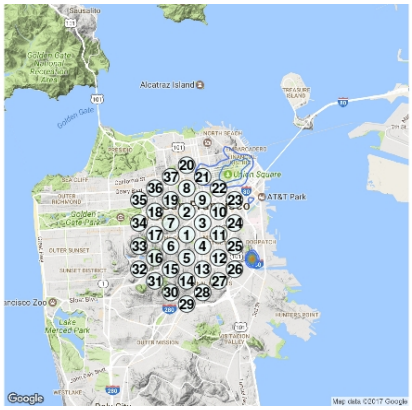


Fig. 2: MEC used for our evaluation: EDCs are co-located with base stations, hence organized as a hexagonal grid.

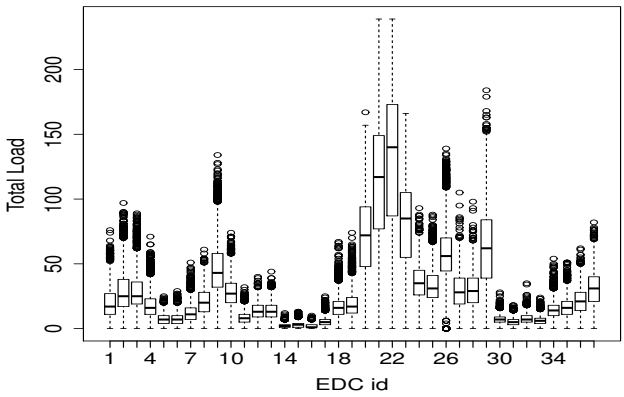


Fig. 3: Workload in 37 EDCs according to our simulation.

In order to realize all stages of the model, we used the package `vars` [5] which contains different functions and methods for conducting VAR analysis implemented in the R platform.

IV. Experiments and Results

In this section, we first describe how we simulated an MEC infrastructure using real mobility traces to generate load at each EDC. Then, we conduct extensive studies to evaluate our proposed algorithm.

A. Experiment setup

We use the real mobility traces of taxis in San Francisco, USA to simulate the load in the EDCs. The dataset contains information about GPS coordinates of 536 taxis collected over 25 days from May 17th, 2008 at a fine time granularity [6], [7]. We filtered abnormal records from the raw GPS traces. Specifically, we set the maximum velocity V_{max} = 115 km/h according to urban traffic regulation and ignore entries with velocity greater than V_{max}.

We assume a cellular infrastructure of 37 cells organized as a hexagonal grid deployed in the area of San Francisco (Figure 2). Each cell contains an EDC which provides service to users located closest to it. The distance between the center point of two adjacent cells is 1 km. At any time slot t, a user connects to the EDC located in the nearest cell and consume one unit of computing capacity (e.g., 1 CPU core). As observed within the trace, users are spatially distributed unevenly, most of them being highly concentrate in some areas at certain times of the day. Hence on average, EDCs located in these highly concentrated locations observe a higher load, while in the opposite sense, EDCs located further from such hot spots observe little load and even appear to have zero load during some time slots t. For instance, as shown in Figure 3, the average load in EDCs #14, #15, #16, #17 is rarely less than 10, while in EDCs #20, #21, #22 this amount is significantly higher. Such load disparity causes great challenge for any prediction model. As we detail below, the performance of our proposed algorithm is significantly divergent when applied over EDCs in the hexagonal grid.

In order to generate the historical load time series for each EDC, we aggregate the load in each cell for 15-minute intervals. Figure 4 shows the resulting time series for three EDCs (EDCs #1, #20 and #37) We use the first 15 days of the time series, dated from 17 May 2008 to 31 May 2008 for training purposes. The remaining data of the days from 1 June 2008 to 10 June 2008 are used for evaluation purposes. The load prediction algorithm is used online to predict the load in each EDC for every 15 minutes ahead. After each run, the training time series is updated by integrating the new observed data from the current interval, and by removing the oldest data.

B. Evaluation metric

We use two error metrics to evaluate the performance of our proposed method. The first is the Symetric Mean Absolute Percentage Error (SMAPE) which is computed as the follows:

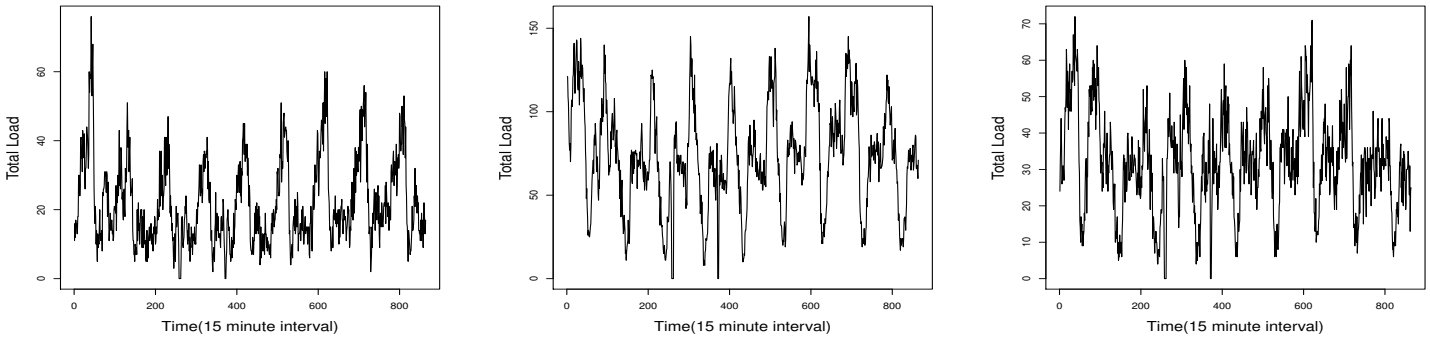
$$SMAPE = \frac{1}{n}\sum_{t=1}^{n}\frac{\mid A_t - F_t \mid}{\mid A_t \mid + \mid F_t \mid}$$

where A_t, F_t are respectively the actual and predicted value at time slot t, and n is the number of observations in the dataset. In essence, SMAPE expresses prediction error as a percentage value, where lower values indicate less residual variance and a better prediction model.
The second measure is the Mean Absolute Error (MAE), which is computed as follows:

$$MAE = \frac{1}{n}\sum_{t=1}^{n}\mid A_t - F_t \mid$$

C. Result and Discussion

1) Comparison with location-unaware prediction: The key idea of our algorithm is that to predict the load for a specific EDC, the algorithm not only utilizes the single historical time series of that EDC, but also the historical data of its neighbors. The knowledge about cross-correlation among these time series might improve the prediction accuracy. To verify that, we first implemented a state-of-the-art baseline method which is often used to characterize and predict data center workloads in literature [8], [9]. We choose Auto Regressive Integrated Moving Average (ARIMA) [10] as the technique to develop



(a) EDC #1 (b) EDC #20 (c) EDC #37

Fig. 4: Load time series on three EDCs from 18 May 2008 to 27 May 2008.

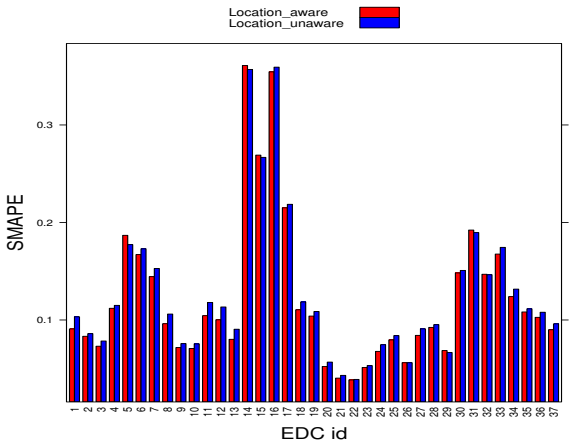


Fig. 5: The prediction accuracy of location-unaware method, and location-aware method.

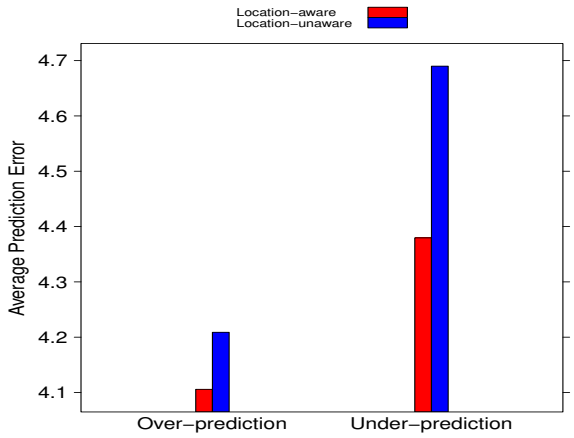


Fig. 6: Over-, under-prediction error of the two models.

the baseline method and borrow the Hyndman-Khandakar algorithm [11] to fit such a model within the R platform. An ARIMA-based prediction technique only considers the historical load of the predicted EDC as input time series, hence we call it location-unaware method to contrast it to our proposed technique. The location-aware and location-unaware load prediction method are invoked to predict the load for 37 EDCs separately with the initial input data setting as presented above. Figure 5 shows the SMAPE difference between the two models. Compared to the baseline method, our proposed technique outperforms in most EDCs, the prediction accuracy being improved by slightly over 4.3%.

We continue to compare the efficacy of the two models by measuring how likely it is for them to over-predict and under-predict. An over-prediction occurs if the prediction value is larger than the observed value, whereas an under-prediction occurs if the actual value is greater than the prediction value. We use MAE metric to evaluate the error in both cases. Figure 6 shows that the location-aware method has better accuracy with both over- and under-predictions.

2) Prediction Accuracy of Location-aware Load Prediction: In this experiment, we analyze the influence of the input data on the performance of our prediction method. Figure 7 represents the SMAPE result over 37 EDCs and their corresponding average load. It can be observed that excluding EDCs #14, #15, #16, #17 where our proposed algorithm returns poor result (SMAPE values on these nodes greater than 0.2), the numeric result on other EDCs are acceptable (SMAPE values less than 0.2). This is due to the fact that the input data in EDCs #14, #15, #16, #17 are extremely sparse with the average load being less than 10, while other EDCs feature more dense input time series. Overall, our proposed method leads to an average prediction accuracy ranging from 64% to 96%.

Our method can also return the prediction intervals that limit the confidence interval for predictions. Figure 8 presents the predicted and observed value in our simulation and the corresponding prediction interval for the last week of the workload. Such prediction intervals can be cleverly applied in different strategies e.g., when one puts higher priority on increasing resource utilization rather than Quality of Service by selecting the numerical value of lower 80% and 95%; or willing to decrease utilization in favor of guaranteeing Quality of Service by choosing the higher 80% and 95%.

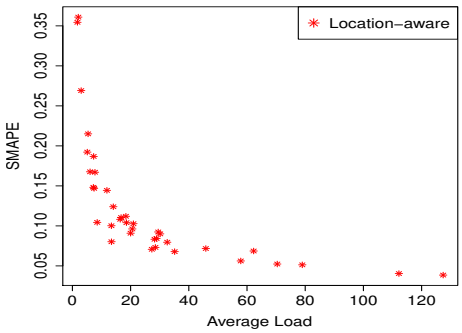


Fig. 7: The influence of input data on the prediction accuracy.

TABLE I: Number neighbors of each EDC

EDC	Number of Neighbors
#1, #2, #3, #4, #5, #6, #7, #8, #9, #10, #11, #12, #13, #14, #15, #16, #17, #18, #19	6
#21, #22, #24, #25, #27, #28, #30, #31, #33, #34, #36, #37	4
#20, #23, #26, #29, #32, #35	3

3) Impact of number of neighbors: Depending on the EDC's position in our simulated MEC, an EDC may have 6, 4 or 3 neighbors (see Table I). The previous experiment shows that our proposed algorithm outperforms the baseline, which indicates that historical load information of an EDC's neighbors is an important information for predicting the EDC's load. Let us evaluate the impact of the number of neighbors of an EDC on the prediction accuracy. Recall that the original taxi trace is unevenly distributed, causing some EDC to have very sparse load which is difficult to predict. Therefore, in this experiment we only consider EDCs for which both models, the baseline and our location-aware one, return reasonably accurate predictions. In order to have more comprehensible view in the difference we use SMAPE metric to evaluate the prediction accuracy of two methods. Figure 9 presents the SMAPE improvement between the proposed algorithm and baseline running over EDCs with different number of neighbors. For EDCs with 3 neighbors, the improvement is not significant (2.2%). However, the betterment is considerable in EDCs with 4 and 6 neighbors, with average prediction accuracy improving by roughly 5.5% abd 7.4%, respectively.

4) Practicability of Real-time Load Prediction: Since load in each EDC may change frequently, it is importance to make sure that the prediction is performed fast enough to allow a resource manager using the predictions amble time to react to the changes. We evaluated the running time of our proposed algorithm measured from initial stage of parameter estimation to the final stage that return the predicted value when conducting the experiments presented above. We run the proposed algorithm to predict load at 37 EDCs for one look-ahead step (15 minute interval), conducting on a PC with Intel i7-4790 CPU and 32GB RAM using a single thread. Recall that the training set includes 15 days of data points, for each 15 minute interval, e.g., 1400 data points in total for each run. As summarized in Table II the aggregated running time of the algorithm over 37 EDCs is 6.75 seconds in which the parameter estimating requires longest time (4.82 seconds). We can conclude that the proposed algorithm can be employed for online predictions.

TABLE II: Running time of the location-aware load prediction over 37 EDCs (in second)

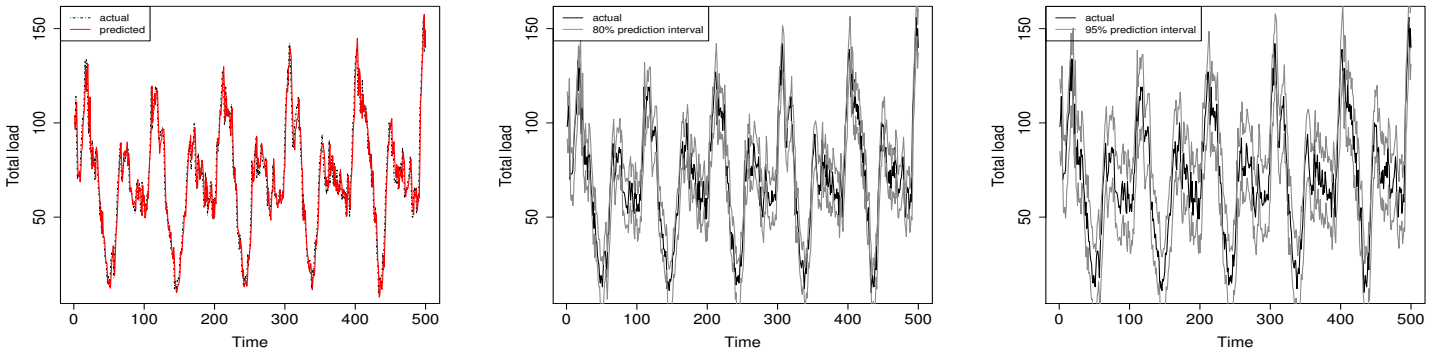
Training set generating	Parameter estimating	Model fitting and checking	Prediction	Total
0.03	4.82	1.74	0.16	6.75

V. Related Work

Since the emergence of traditional centralized clouds, workload prediction and characterization has extensively been studied, both by industry and academia, to guide resource management decisions and optimize infrastructure usage, in particular for auto-scaling. The works in [8], [9], [12], [13] applied various univariate time series models for forecasting server and application workloads. The general idea of such approaches is that they first examine the potential load patterns, including daily, weekly and even monthly variation. Then models such as Autoregressive Moving Average (ARMA) or ARIMA are adopted to predict the load over a period ranging from half an hour to several hours. Caron et al. [14] propose short-term resource demand prediction using pattern matching, i.e., a pattern similar to the current load window is searched for in historical data and trends in the found match are used for predicting the next load window. In [15], [16], artificial neural networks and linear regression are used in concert to develop a hybrid prediction model for application workloads.

In general, the state-of-the-art techniques mentioned only consider the knowledge of a single server or application workloads to anticipate future resource demand. They can efficiently work in the context of centralized Clouds, however, in the context of MECs, the user mobility and the small radius of coverage of EDCs make predictions more challenging. Compared to existing works, our proposed method employs the Vector AutoRegression (VAR) model, that is capable of tackling these challenges by tacking into account the dynamic correlation patterns among the workloads of EDCs in close vicinity.

The VAR model has been used for real-time forecasting in other fields, particularly for predicting macro-economic metrics, stocks performance and road traffic patterns. For example, Sims and Zha [17] introduce a highly accurate prediction approach for inflation, GDP growth and unemployment by exerting the reduced form and structural VAR model trained



(a) Actual and predicted values (b) 80% prediction interval and actual values (c) 95% prediction interval and actual values

Fig. 8: Results of the Location-aware load prediction for 5 days of EDC #20.

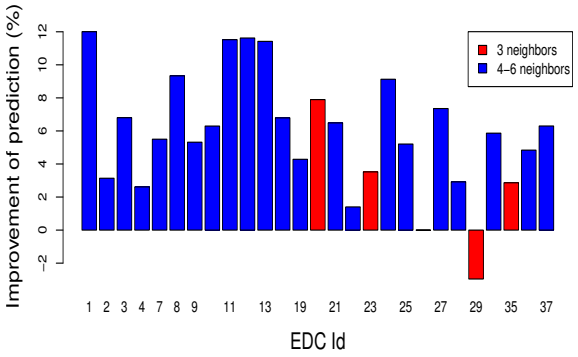


Fig. 9: The improvement of predictions between the two models depending on the number of neighbors of each EDC

with prior information. Chandra et al. [18] applied the VAR model to develop a short term traffic prediction on freeways that takes into account the effects of upstream and downstream location information. To our knowledge, our work is the first to employ the VAR model for load prediction in MEC.

VI. Conclusion and Future Work

In order to circumvent the challenge of capacity adjustment in EDCs, we proposed a novel location-aware load prediction approach which deals with user mobility by correlating load fluctuations of EDCs in physical proximity. The experimental results show that our proposed method is capable of achieving high accuracy prediction for EDCs with significant load. Our method is performing particularly well for EDCs with many neighbors. Overall, our method outperforms state-of-the-art location-unaware prediction methods by up to 4.3%, which, considering the scale of MECs, can lead to more efficient resource allocation and substantial cost savings.

As future work, we plan to test our approach on a more diverse dataset obtained from mobile network operators. Furthermore, our current implementation relies on centralized collection and processing of load time series, i.e., all monitoring data from all EDCs are collected and forecast is performed on a single server. We will work on devising a distributed prediction algorithm, that runs on each EDC or on cluster of EDCs in close vicinity. Finally, we will test whether load prediction in MECs can be improved using machine learning techniques such as deep learning.

Acknowledgement

This work was partially supported by the Wallenberg Autonomous Systems and Software Program (WASP).

References

[1] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the internet of things," in *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*. ACM, 2012, pp. 13–16.

[2] Nokia, "Multi-access Edge Computing(MEC)," https://networks.nokia.com/solutions/multi-access-edge-computing, 2016.

[3] H. Lütkepohl, *Vector autoregressive models*. Springer, 2011.

[4] J. R. Hosking, "The multivariate portmanteau statistic," *Journal of the American Statistical Association*, vol. 75, no. 371, pp. 602–608, 1980.

[5] B. Pfaff *et al.*, "Var, svar and svec models: Implementation within r package vars," *Journal of Statistical Software*, vol. 27, no. 4, pp. 1–32, 2008.

[6] M. Piorkowski, N. Sarafijanovic-Djukic, and M. Grossglauser, "CRAWDAD dataset epfl/mobility (v. 2009-02-24)," Downloaded from http://crawdad.org/epfl/mobility/20090224, Feb. 2009.

[7] ——, "A parsimonious model of mobile partitioned networks with clustering," in *2009 First International Communication Systems and Networks and Workshops*, Jan 2009, pp. 1–10.

[8] V. G. Tran, V. Debusschere, and S. Bacha, "Hourly server workload forecasting up to 168 hours ahead using seasonal arima model," in *2012 IEEE International Conference on Industrial Technology*, March 2012, pp. 1127–1131.

[9] W. Fang, Z. Lu, J. Wu, and Z. Cao, "Rpps: A novel resource prediction and provisioning scheme in cloud data center," in *2012 IEEE Ninth International Conference on Services Computing*, June 2012, pp. 609–616.

[10] G. E. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.

[11] R. J. Hyndman, Y. Khandakar *et al.*, "Automatic time series for forecasting: the forecast package for r," Tech. Rep., 2007.

[12] H. Zhang, G. Jiang, K. Yoshihira, H. Chen, and A. Saxena, "Intelligent workload factoring for a hybrid cloud computing model," in *2009 Congress on Services - I*, July 2009, pp. 701–708.

[13] N. Roy, A. Dubey, and A. Gokhale, "Efficient autoscaling in the cloud using predictive models for workload forecasting," in *2011 IEEE 4th International Conference on Cloud Computing*, July 2011, pp. 500–507.

[14] E. Caron, F. Desprez, and A. Muresan, "Forecasting for grid and cloud computing on-demand resources based on pattern matching," in *2010 IEEE Second International Conference on Cloud Computing Technology and Science*, Nov 2010, pp. 456–463.
[15] M. Sladescu, A. Fekete, K. Lee, and A. Liu, "Event aware workload prediction: A study using auction events," in *International Conference on Web Information Systems Engineering*. Springer, 2012, pp. 368–381.
[16] S. Islam, J. Keung, K. Lee, and A. Liu, "Empirical prediction models for adaptive resource provisioning in the cloud," *Future Generation Computer Systems*, vol. 28, no. 1, pp. 155–162, 2012.
[17] C. A. Sims and T. Zha, "Bayesian methods for dynamic multivariate models," *International Economic Review*, pp. 949–968, 1998.
[18] S. R. Chandra and H. Al-Deek, "Predictions of freeway traffic speeds and volumes using vector autoregressive models," *Journal of Intelligent Transportation Systems*, vol. 13, no. 2, pp. 53–72, 2009.