DAPS - A Web Based System for Predicting IoT Sensor Data

Biljana Risteska Stojkoska, *Member, IEEE*

***Abstract* — IoT systems are expected to generate voluminous raw sensor measurements that require high bandwidths to be transmitted to the clouds. IoT data prediction is one solution to this issue. Yet, many algorithms based on different models for times series prediction can be used for this purpose. IoT developers have to choose among many of them, since they perform differently for different sensor measurements. In this paper, we present DAta Prediction System (DAPS), a web-based online tool that helps IoT developers to choose the most suitable data prediction algorithms for their application.**

***Keywords* — IoT, prediction, sensor data, web-based system.**

I. INTRODUCTION

SENSORS have been widely used for decades in many applications, so there is a constant need for new algorithms for their analyses. The spread of Internet of Things (IoT) paradigm induced new challenges associated not only for data analyses, but also for data transmission over the wireless medium. There are some predictions that by 2020 IoT will consist of more than 50 billion of objects [1], in different application domains like smart home [2], ambient assisted living [3], health care [4], smart city [5], etc. Considering the volume and velocity of data produced in most IoT scenarios, the problem of data size reduction becomes crucial if one wants to optimize the costs for the potential solution and reduce the data latency. There have been many ways in the literature that aim to decrease the number of messages transmitted through the network. The very traditional approach is data compression, where historical sensor measurements are compressed and send periodically. Although this method is proven to be energy efficient, it falls to respond in real-time applications. Other approach is data prediction, which uses well known techniques from time series analysis. This method is based on Dual Prediction Scheme (DPS), where sensor measurements are sent to the cloud only if the predicted sensor value differs greatly from the actual one. Many research groups have developed algorithms for sensor data prediction, and evaluated them on different dataset. However, there is no a golden standard for choosing the best algorithm, since it is an application specific task.

This research aims to help developers of IoT solution to choose the best algorithm for data prediction regarding their application data. For this propose, we developed DAta Prediction System (DAPS), a web-based online tool that implements five different algorithms for data prediction. Users can upload historic sensor measurement from their application, and DAPS can analyze which data prediction algorithm fits best regarding two evaluation metrics: prediction accuracy and reduction in number of transmissions. To the best of our knowledge, DAPS is the first tool designed for the IoT solution developers to help them create energy efficient applications.

This paper is organized as follows. Data prediction algorithms are presented in Section II. Section III describes the development of the DAPS. Two case study are presented in Section IV. Finally, the conclusion is given in Section V.

II. DATA PREDICTION ALGORITHMS

In this section we are going to explain the basics of data prediction. Then, we briefly introduce the algorithms used in DAPS.

Data prediction is important in many different fields (studying wildlife, environment, households, analytics in multimillion companies, etc.) [6], as it helps to make better decisions about some repeatable event. Different models are used in the literature for data prediction [7]. These models can have many forms and represent different stochastic processes. There are three broad classes of models: Autoregressive (AR), Integrated (I) and Moving Average (MA) [8], which rely on previous data points. Combinations of these classes are possible, such are Autoregressive Moving Average (ARMA), Autoregressive Integrated Moving Average (ARIMA), etc.

After the next value is predicted, we compare it with the real value (y) and decide if the prediction is in range [y-Ɛ, y+Ɛ], where Ɛ is a constant that is user defined. If the prediction is within this range then we take it as correct, if not, then it is incorrect and the actual value from the time series is sent to the cloud, and also used later for the next prediction.

In our web-based system, we implemented there Moving Average (MA) algorithms of different order and two Least Mean Square (LMS) based algorithms.

A. Simple Moving Average

Simple Moving Average (SMA) is the unweighted mean of the previous n data. The calculated value is the

This project was financially supported by the Faculty of Computer Science and Engineering, Skopje, Macedonia.

Biljana Risteska Stojkoska is with the Faculty of Computer Science and Engineering, University Ss. Cyril and Methodius, Rugjer Boshkovikj 16, P.O. Box 393, 1000 Skopje, Macedonia (e-mail: biljana.stojkoska@finki.ukim.mk).

978-1-5386-3073-0/17/$31.00 ©2017 IEEE

prediction for the following one in the time series [9].

$$SMA_n = \frac{1}{n} \sum_{t=k-n+1}^{k} P_t \quad (1)$$

The prediction is obtained using (1), where n is the number of previous measurements included in the average, k is the relative position of the measurement currently being considered within the total number of measurements, and P_t is the last value we have had until the moment of prediction.

There are different approaches of using the SMA technique depending on the number of previous data used in the prediction. Moving Average 1 (MA1) takes the previous value as the next prediction. It is the simplest predicting technique possible and very inaccurate, although for some data, like temperature measurements, it can be adequate. The Moving Average 2 (MA2) calculates the mean of the previous two values in the series, and performs slightly better than the MA1.

Additionally, there are other types of Moving Average algorithms, like Weighted Moving Average (WMA). This is an average that adds multiplying factors to give different weights to the precious data [10].

B. Least Mean Square Algorithms

Least Mean Square (LMS) algorithms are a class of adaptive filters used to mimic a desired filter. LMS gives the least mean square of the error signal, which is the difference between the desired and the actual signal [11].

There is an unknown system marked as h(n) and an adaptive filter $\hat{h}$ (n) which tries to adapt to the system and be as close as possible to it (Fig. 1). The input to the system x(n) is the data from the time series. The variables y(n) and ŷ(n) are the output of the system and they are compared to give the error e(n). ŷ(n) is calculated as a dot product with the filter weights (2).

$$\hat{y}(n)=W(n)*X(n) \quad (2)$$

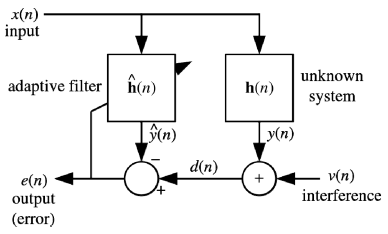

Fig. 1. Least Mean Square scheme

The main goal of the LMS is to adapt the filter weights, so the filter gives prediction as close as possible to the actual value (3). If the Mean Square Error (MSE) gradient is positive, it implies that the error would keep increasing positively. This means we need to reduce the weights, and vice versa.

$$W(n+1)=W(n)+2\mu e(n)X(n) \quad (3)$$

In (3), X(n) is the input signal vector of adaptive filter at n-th time, W(n) is the estimate value of weights vector, e(n) is the error signal, while μ is the step factor, which is used to control the stability and the convergence rate of the algorithm. The mean-square error, as a function of filter weights, is a quadratic function which means it has only one extremum, that minimizes the mean-square error, which is the optimal weight. The LMS thus, approaches towards optimal weights by ascending/descending down the MSE vs. filter weight curve.

A variation of LMS, known in literature as Least Mean Square with Variable Step Size (LMS-VSS) has reported better result than LMS [12][13]. The only difference to the LMS is that here the step factor μ is changing.

C. Evaluation metrics

Root Mean Square Error (RMSE) is a frequently used measurement for the accuracy. The RMSE is a method to measure the difference between the predicted and the actual values in a time series (6), where y_i and $\hat{y}_i$ are the true and the predicted measurement respectively.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n} (\hat{y}_i - y_i)^2}{n}} \quad (6)$$

In context of data reduction for IoT based solution, number of messages sent through the network can be more useful metric. Therefore, we define Percentage of sent messages, as a fraction of all measurements, considering that each message contains one measurement.

III. Development of DAPS

In this Section, we will explain the process of developing DAPS, the web part of the system, the implementation of the algorithms and the graph drawing part.

A. Design and implementation of DAPS

In order to visualize the importance of data prediction and show the results of its algorithms, we created a web-based system that shows the calculations in a more comprehensible way for the user. Since it is a system on the web, the client-server architecture is the most common and appropriate. It is designed in a way that the user sends all the necessary data to the server. The server analyses and processes the request, and later visualizes the results.

Different parts of the system are implemented with different technologies. Spring Boot, which is Java based framework, was used for making this system web-based. The backend was made with Spring Boot. Views were made with HTML and interactions were made with JavaScript language. Data transfer between the controllers in Spring Boot and the HTML is managed with server-side Java template Thymeleaf.

Fig. 2 shows the flow of the actions in the system, step by step.

B. Functionalities of DAPS

The first part is uploading a file in CSV format with the one-dimensional time series data. Next, the user should choose Ɛ, which sets the range in which a prediction is considered correct. This is set by the user along with the interval in that range. The user also selects the algorithms to be executed on the data. One or more can be chosen and run.

After the user clicks the "Upload" button, the system reads the data, runs the chosen algorithms and writes the results in separate files that are created in the user's "Downloads" folder (Fig. 3). Finally, the c3.js library reads the results and draws two graphs with certain parameters. The first one shows the percent of sent messages with all of the selected algorithms. The second presents the Root Mean Square Error (RMSE) for each of them.

As mentioned before, this system runs prediction algorithms and then presents the results on graphs. Here we are going to explain what those "results" represent. There are 2 graphs, each of them representing one measurement. Those are percentage of sent messages and Root Mean Square Error (RMSE).

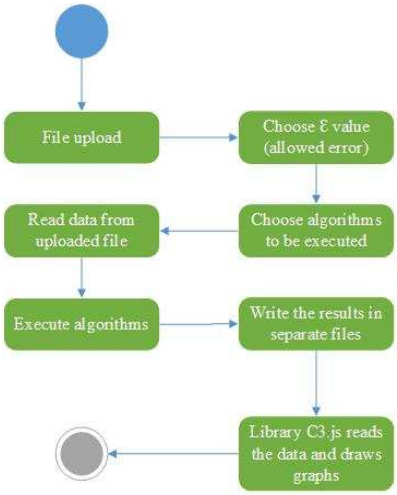

Fig. 2. Activity diagram for the DAPS

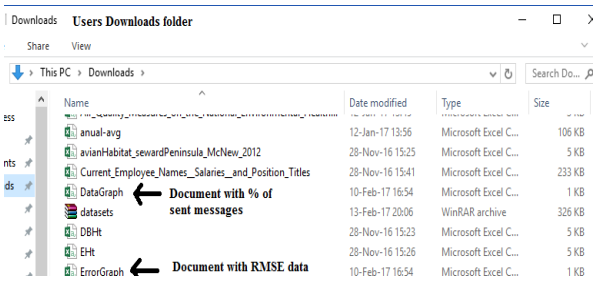

Fig. 3. User's Downloads folder and the two files created in it.

The percent of data reduction is self-descriptive. It shows the relation of the number of sent messages to the number of total possible messages in a sensor system. It means that each of the samples in the time series is considered as one message and if there are no prediction algorithms every single one of them must be transmitted in the system. But when an algorithm is run on the data we get some predictions and those predictions are compared to the actual data. If the prediction is good enough (the prediction is in [y-Ɛ, y+Ɛ] interval) then we consider that the message is not sent since the prediction is correct.

IV. Case Study

In this section, we will show the actual output of the system and see how it performs on different datasets. We consider two different datasets in order to compare the results. The first one is a dataset that has measurements for the air quality in the United States [14], and the second has data that shows the percentage of readmissions in hospitals in the US [15]. Readmission is when a patient comes back in the same hospital after initially being released.

In the percentage of sent messages graphs that are produced, the X axis represents the allowed error that the user sets. The Y axis is the percent of sent messages for each of the values for the allowed error. In the RMSE graphs, the X axis is also the allowed error and the Y axis represents the RMSE values for the corresponding allowed errors.

A. Air quality measurements

In the first case, DAPS is used to predict air quality measurements, in particular annual averages of the presence of PM 2.5 particles in micrograms per cubic meter [14]. The data contains measurements from approximately 4,000 monitoring stations around the US, mainly in urban areas. Regarding the frequency of the sampling and taking measurements, it is different for every station. The data here is annual from all stations. The output of the system is shown in Fig. 4.

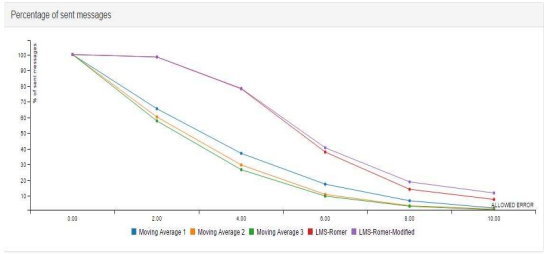

Fig. 4. Percentage of sent messages for air quality measurements

The graph shows that Moving Average 3 is the best, with the lowest number of sent messages, but the difference compared to other Moving Average algorithms is small.

Fig. 5 shows RMSE of the same dataset. As can be seen from the graph, the LMS has the smallest error, but by increasing the allowed values for the Ɛ, the results of all algorithms are getting closer.

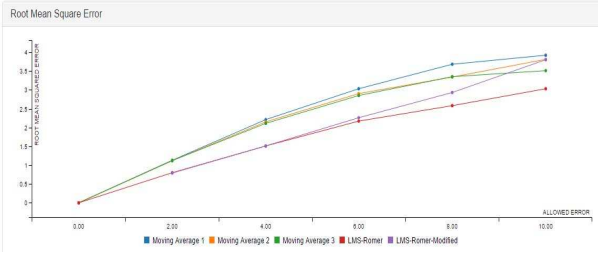

Fig. 5. Root Mean Square Error (RMSE) for air quality measurements

B. *Readmissions in hospital*

Dataset from [15] refers to the percentage of people who had been received back again after their first discharge from hospital for treatment. A readmission is when a patient comes back to the hospital in a time span of 30 days after being released from there. Readmission rates have been increasingly used as an outcome measure in health services research and as a quality benchmark for health systems. This is annual data from the US Department of Health.

Fig. 6 and Fig. 7 represent the results of applying [15] to DAPS. Fig. 6 shows that, for small Ɛ values, least sent messages are with Moving Average 1, but for bigger Ɛ values, Moving Average 2 and Moving Average 3 are better. Fig. 7, which is similar with the previous case, confirms that LMS algorithms have the smallest error.

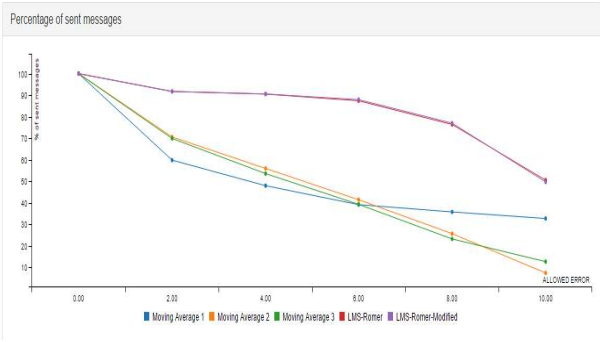

Fig. 6. Percentage of sent messages for people received back

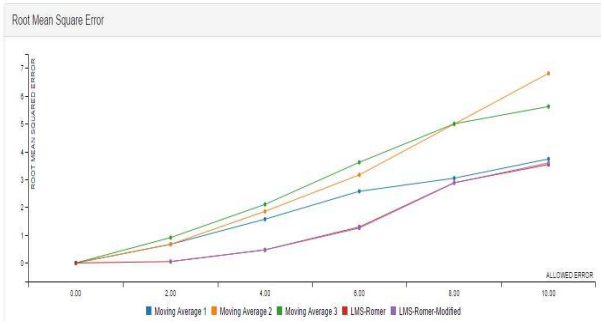

Fig. 7. Root Mean Square Error(RMSE) for people received back

Most of the analyzed datasets, as the two previous cases, show the following:

- Moving Average algorithms send less messages, but they have a greater RMSE.
- LMS algorithms send more messages, but they have lower RMSE.

Of course, there are exceptions in this rule, confirmed in fewer processed datasets, which means that finally the results depend on the dataset itself.

V. CONCLUSION

In this paper, we present DAta Prediction System (DAPS), a web-based online tool that helps future developers of wireless sensor networks (WSN) and Internet of Things (IoT) solutions to choose the most suitable data prediction algorithms for their application. DAPS performs data prediction for one-dimensional sensor readings, using five different algorithms, and compares their performances regarding two different evaluation metrics. Additionally, the visualization engine from DAPS visualizes the results obtained from the data prediction, by means of MSE and percentage of data reduction.

REFERENCES

[1] Evans, Dave. "The Internet of Things How the Next Evolution of the Internet is Changing Everything (April 2011)." White Paper by Cisco Internet Business Solutions Group (IBSG) (2012).

[2] Stojkoska, Biljana L. Risteska, and Kire V. Trivodaliev. "A review of Internet of Things for smart home: Challenges and solutions." *Journal of Cleaner Production* 140 (2017): 1454-1464.

[3] Risteska Stojkoska, Biljana, Kire Trivodaliev, and Danco Davcev. "Internet of Things Framework for Home Care Systems." Wireless Communications and Mobile Computing 2017 (2017).

[4] Xu, Boyi, Li Da Xu, Hongming Cai, Cheng Xie, Jingyuan Hu, and Fenglin Bu. "Ubiquitous data accessing method in IoT-based information system for emergency medical services." IEEE Transactions on Industrial Informatics 10, no. 2 (2014): 1578-1586.

[5] Sanchez, Luis, Luis Muñoz, Jose Antonio Galache, Pablo Sotres, Juan R. Santana, Veronica Gutierrez, Rajiv Ramdhany et al. "SmartSantander: IoT experimentation over a smart city testbed." Computer Networks 61 (2014): 217-238.

[6] Zissis, Dimitrios, Elias K. Xidias, and Dimitrios Lekkas. "Real-time vessel behavior prediction." Evolving Systems 7, no. 1 (2016): 29-40.

[7] Sheskin, David J. Handbook of parametric and nonparametric statistical procedures. crc Press, 2003.

[8] Gershenfeld, Neil A. The nature of mathematical modeling. Cambridge university press, 1999.

[9] Risteska Stojkoska, Biljana, Andrijana Popovska Avramova, and Periklis Chatzimisios. "Application of wireless sensor networks for indoor temperature regulation." *International Journal of Distributed Sensor Networks* 10, no. 5 (2014): 502419.

[10] John Devcic, "Weighted Moving Averages", [Online] Available: http://www.investopedia.com/articles/technical/060401.asp

[11] Santini Silvia and Kay Romer. "An adaptive strategy for quality-based data reduction in wireless sensor networks." In Proceedings of the 3rd international conference on networked sensing systems (INSS 2006), pp. 29-36. 2006.

[12] Stojkoska Biljana, Dimitar Solev, and Danco Davcev. "Data prediction in WSN using variable step size LMS algorithm." In Proceedings of the 5th International Conference on Sensor Technologies and Applications. 2011.

[13] Stojkoska, Biljana Risteska, Dimitar Solev, and Danco Davcev. "Variable step size LMS Algorithm for Data Prediction in wireless sensor networks." Sensors & Transducers 14, no. 2 (2012): 111.

[14] "Air Quality Measures on the National Environmental Health Tracking Network", [Online], Available: https://catalog.data.gov/dataset/air-quality-measures-on-the-national-environmental-health-tracking-network

[15] "Readmissions and Deaths - Hospital", [Online], Available: https://catalog.data.gov/dataset/readmissions-and-deaths-hospital