Predicting Encounters in Opportunistic Networks using Gaussian Process

Cristian Chilipirea, Andreea-Cristina Petre, Ciprian Dobre
University POLITEHNICA of Bucharest
Bucharest, Romania
Emails: {cristian.chilipirea, andreea.petre}@cti.pub.ro, ciprian.dobre@cs.pub.ro

Abstract

In particular types of Delay-Tolerant Networks (DTN) such as Opportunistic Mobile Networks, node connectivity is transient, and connections are sparse and small in length. For this reason, traditional routing mechanisms are no longer suitable. Routing algorithms designed for such networks try to maximize the probability of successful message delivery. The most popular approach is to compute the probability of delivering a message using information such as node contacts and location knowledge, thus using past encounters to predict future ones. In this paper we investigate the predictability of human mobility and interactions patterns. We propose the use of supervised learning techniques together with Gaussian process modeling to predict future encounters based on historical patterns of individual nodes. We analyze their accuracy compared to previous prediction techniques, using real-world mobility data traces.

Keywords: opportunistic networking, smart ad-hoc networks, Gaussian process, prediction algorithm.

1. Introduction

Today smart phones and tablet PCs are becoming commodity hardware. They are seen everywhere, as more people realize that having more sensing and computing capabilities in every-day situations is attractive for many reasons. This, combined with the proliferation of wireless technologies (like Bluetooth or Wi-Fi), create the basis for a new paradigm of communication and interaction, ad-hoc networks, such as Opportunistic Mobile Networks (OMN) [5]. An OMN is composed of mobile devices that are carried by individuals and act as nodes in the network. Having no a priori structure, the nodes exchange data while in proximity without predetermined end-to-end paths. Thus, if an encountered node is not the destination of a message that must be delivered, the current node makes the decision of forwarding the message or not. The decision of choosing the next hop is taken considering if it will be able to get the message to the destination, either through a future direct transfer or through more nodes. Moreover each device has limited storing capacity, bandwidth, computing power, energy. Therefore, a different approach in the routing process must be adopted, by extending the classical store-and-forward paradigm to store-carry-forward (SCF) [4].

In the context of a network composed of mobile devices used by individuals, understanding behavioral patterns can prove to be a key factor in optimizing the routing process. Previous protocols, such as BUBBLE Rap [1, 16], use social data about the nodes to calculate the probability that the current neighbor is able to deliver the packet. Others try to predict future encounters of a device by indentifying a function that describes individual mobility patterns, e.g. the Poisson distribution [3].

Individuals tend to follow patterns in their movement. Such patterns can be observed in the encounters detected by their mobile devices, as we previously demonstrated in [3]. In this paper, we propose a generalized method of predicting future encounters based on an artificial intelligence technique, the Gaussian process probabilistic classifier [12]. Such a classifier will provide the basis for intelligent routing, predicting future behavior by learning from past encounters. Based on previous work [3] we have determined that the most suitable approach is to predict the hour of the day of the week in which the node has encounters and how many encounters with individual nodes are there.

Predicting object behavior can also be used in the case of complex workflow scheduling in Grid environments [22].

For our experiments we used a real-life mobile trace gathered at the Faculty of Automatic Controls and Computers, University POLITEHNICA of Bucharest, in the Spring of 2012 [15]. To validate our findings we also used the publically available trace collected at the St. Andrews Sassy [14].

The rest of the paper is structured as follows: In Section 2 we present related work; Section 3 presents the proposed algorithm, together with implementation details. Section 4 describes the experimental setup and presents results. Section 5 concludes and presents future work.

978-0-7695-4980-4/13 $26.00 © 2013 IEEE
DOI 10.1109/CSCS.2013.15

2. Related Work

The growth in popularity of OMN shown over the past years caused an increase in research activity in this area. The nodes in an opportunistic network are devices carried by individuals. A comprehensive review of opportunistic networking can be found in [7]. Functions such as security, message forwarding, data dissemination and mobility models were analyzed mostly in the context of the EU Haggle project. The authors introduce HCMM, a mobility model that fuses the spatial and social dimensions [7]. Additionally, several well-known opportunistic forwarding algorithms were analyzed, out of which BUBBLE Rap [1], PROPICMAN [8] and HIBOp [9] presented the most promising networking capabilities.

Previous results showed that social information is particular useful for determining the next hop to which to forward messages [7], since individuals tend to interact with each other in some contexts in concordance with social relationships. This is an overspecialization of human interaction because proximity, a key element in forwarding messages in opportunistic networks, is not determined by social relationships in all contexts (an example is a college environment, where students are organized into groups not based by affiliations and common interests). Therefore a more suitable approach is the use of past encounters for the prediction of future encounters. BUBBLE Rap algorithm [1] is an example of a socially-aware forwarding algorithm in delay tolerant networks. It forms communities based on past encounters, and it uses node centrality to forward a message to the most popular node in the current community (the community of the node that forwards the message) by means of a local rank, and then sends it in between communities using a global rank. When the message reaches the community a destination node is a part of, it has a greater chance of reaching that node. It is important to note that communities change in time, students take different classes, individuals periodically change their work place or work assignment. How a community is determined massively impacts BUBBLE Rap performance. Using social data, like Facebook friend status, alone is not always appropriate, take the case of students where one's affiliation does not necessarily determine once attendance to classes.

In [6], a study of predictable human interactions on forwarding in pocket switched networks (PSN) is made by Hui and Cowcroft. Here it is shown that human mobility is predictable on a daily bases and a distributed forwarding algorithm that uses node centrality is designed, using a dataset extracted from the MIT's Reality Mining traces [10].

Some approaches to forwarding in opportunistic networks compute the probability of delivering a message using information such as node contacts and location knowledge. PROPHET [8] is such an algorithm, utilizing past encounters to predict future ones. We find that the most successful algorithms, like [1] [8] [9] [2] [6], utilize nodes history in a way or another. And as such we consider that past encounters are probably a good estimate to determine the probability of a future encounter.

In case of SAP [2] a multitude of variables are used to determine future encounters, they include message freshness, hour of the day at which past encounters occurred, community, Facebook social data, popularity. All of them are combined to determine the probability that the current neighbor can deliver or can help in the delivering the message and the message is forwarded if the probability reaches a certain threshold. In SAP the prediction uses a Poisson distribution to simulate individual movement patterns [3]. In fact, as shown previously, the Poisson distribution fits well with the traces collected within the University POLITEHNICA of Bucharest [15]. The results presented in [15] are promising, and they prove that in most cases individuals act according to the probability they predicted. The authors split the time into hours of day and day of the week and they calculate the probability of encountering a number of nodes in each specific timeframe. In the paper they also show that patterns are clearly noticeable inside the traces, students tend to have connections in the weekdays during school hours and very few outside these intervals. However, using a Poisson distribution might prove to not best fit all scenarios. Determining the best function to fit a trace can be time consuming and is not applicable in a real life scenario or can be determined for a large number of nodes, it does not scale.

Our method extends what [3] proposes by the use of a Gaussian process to determine a best fit for individual nodes. As such it is a generalization of the work presented in [3]. Furthermore the proposed method can replace [3] in algorithms such as SAP [6].

3. Proposed Solution

To achieve the desired results we decided to use the Gaussian Process classifier described in the following subsection. We decided to use the Gaussian Process after we tested our data with a Neural Network.

The Neural Network was not particularly compatible because of its linearity. Node encounter patterns tend to have centers in the middle of the day and as such we could not use a linear classifier. We searched for a classifier that would fit for a pattern that has a bell shaped center, where most encounters are actually located.

3.1. Gaussian Process

Because of the mobility and the unknown states that a Delay Tolerant Network can have at any point in time, routing packets in OMNs can be a difficult task. If a pattern of the node movements in the network could be found, then we could predict future connections and construct a route for the packets that need to be delivered. In the rest of the section we describe a solution of this problem based on Gaussian Process, using a Pearson VII Universal Kernel (PUK). The formal description of PUK can be found at [17].

We view the prediction of individual behavior as a classification problem, where the positive data is given by the actual encounters. The negative data is in this case everything that is not positive. Such a vast domain makes the training of the classifier computationally difficult and, therefore, a balanced number of negative examples were selected randomly.

A detailed formal description of the Gaussian Process for probabilistic binary classification is presented in [18]. A Gaussian process is a collection of random variables, any finite set of which have a joint Gaussian distribution [18]. In other words it is a distribution over functions, definition that is consistent with a function space view.

Gaussian process classification is a non-parametric classification method. Such an approach has the advantage of not requiring large amounts of data in order to obtain the parameters of the model and the complexity of the model increases as more data points are added. This fits the case of OMNs, where connections are sparse. Parametric methods are sensitive to the choices of various parameters. It is sensible to avoid such assumptions by using a nonparametric model when the feature space is strongly non-linear and a wrong choice of parameters might lead to erroneous results.

A Gaussian process is completely specified by a mean function or a covariance function (or the kernel):

$$f(x) \sim GP(m(x), k(x, x'))$$

where $m(x)$ is the mean function and $k(x, x')$ the covariance function of a real process $f(x)$. In this paper the kernel was selected to be a Pearson VII Universal Kernel.

Let $x_i \in D$ be data points from a finite dataset *D*. The classifier is trained using supervised learning and, therefore, each data point has an associated class label $y_i \in \{C_+, C_-\}$, where C_+ represents the positive class and C_- represents the negative class. The classification problem lays in predicting the probability that a given test point x_* belongs to one of the mentioned classes.

The positive class membership probability $P(y = C_+ \mid x)$ has the following expression:

$$P(y = C_+ \mid x) = sig(f(x))$$

where *f* is a latent function $f : \mathbb{R} \to \mathbb{R}$, that is mapped into the interval [0, 1] by means of a sigmoid function $sig : \mathbb{R} \to [0,1]$ (thus *sig* is a sigmoid transformation function).

The Gaussian Process needs the matrix $X = [x_1, x_2, \dots, x_n]$ that has the size n x d data points representing the training points, and the vector $y = [y_1, y_2, \dots, y_n]^T$ of size n x 1 (class labels for the training set) and the latent function values $f = [f_1, f_2, \dots, f_n]^T$, where $f_i = f(x_i)$. The predictive class membership probability $p_* = P(y_* = C_+ | x_*, y, X, \theta)$ is obtained by calculating the following integral (usually by approximations), representing the marginal likelihood (evidence):

$$P(y_* | x_*, y, X, \theta) = \int P(y_* | f_*) P(f_* | x_*, y, X, \theta)\, df_*$$

In the previous formula θ stands for the hyper parameters (free variables) the covariance function depends on.

The choice of kernel or covariance function is important. The main purpose of a kernel function is to transform a non-linear input space into a space characterized by more dimensions such that the solution of the problem can be represented in a linear form. The particular selection of a kernel function is greatly dependent on the input data, of the underlying relationships that need to be modeled; therefore a non-fitting function can lead to a less then optimal result. There are various kernel functions, out of which some of the most used are linear kernel, polynomial kernel, sigmoid kernel or Radial Basis Function (RBF). In this paper we used a Pearson VII Universal Kernel (PUK), a flexible function that can change its peak shape from Gaussian to Lorentzian and more by adapting a small number of parameters [17]. This makes PUK suitable as a generic universal kernel and this characteristic determined its utilization in the current work.

Such a predictive model was used because we believe that humans exhibit patterns in their behavior, especially in environments such as academia or work places, as is the case for the traces described in the Experiment Setup section.

3.2. Implementation

To be able to predict future encounters we have decided to split the time into hours of the day and day of the week. We believe that these 2 variables make the most difference in ones behavior. Choosing a smaller time unit like a minute or half an hour is possible; we decided to use time intervals of an hour because of the random aspects in a person's behavior. A shorter time frame might have been too strongly affected by the small changes in schedule such as being late to a meeting. The hour of the day is small enough to give a needed resolution and big enough to not be affected by small changes in a person's pattern.

We split the available data in two, the training data for the Gaussian Process and the validation data. We have observed that different splits give different results, thus making changes in behavioral patterns apparent.

We made two separate tests: one where we attempted to predict the particular hour that an encounter with any node is more likely to occur, and another one where we tried to predict the number of individual encounters a node will have with other nodes at a specific hour/day of the week. As previously mentioned, for these experiments we used the data collected at UPB. Because the volume of training data was too large WEKA [19], the tool we used as a Gaussian process, took a large amount of time to get through the training. Thus, we decided to use the median of the number of individual encounters at a specific hour/day of the week. The same was done for the same test, in this case we only added an hour/day combination as a positive result only if during the weeks in the tests it had more weeks with encounters at that time frame then weeks without.

A real-life implementation of this procedure should retrain the Gaussian Process periodically so that any changes in behavior, such as moving to a second semester or next year or changing jobs should be detected by it. We should mention that keeping all the logs of past encounter is not necessary, the last few weeks or months should suffice; persons tend to change their behavior and this time interval should be large enough to express this.

At some points a log clean-up could be required, forced by a move to a different country or city, changing of jobs, advancing from academia to the job market. Detecting automatically when such a clean-up should be executed is an interesting subject that would require a new view on artificial intelligence and data gathering. To summarize it is simpler to detect when the current location has been changed but a lot harder to detect if this new location is a new permanent residence or not.

4. Experimental Setup and Results

This section represents an experimental analysis of the proposed method of using Gaussian Process with Pearson VII for kernel to predict future encounters.

4.1. Experimental Setup

To evaluate the prediction we first performed the experiments on the UPB 2012 trace [15]. The trace includes 73 participants from the University POLITEHNICA of Bucharest, Romania. From the begging of March to May participants had their phones record Bluetooth and Wi-Fi encounters with the mobile devices of the other participants. Next we validated our conclusions on another trace, from University St. Andrews standrews-sassy [14]. In this trace the data was collected from 25 mobile participants, and it includes encounters monitored on a larger scale (the town of StAndrews, in UK). We also split the data into two parts: the first part was used to train the Gaussian Process and the second part was used for validation.

First, we considered only 2 variables: 1) the hour of the day, and 2) the day of the week. If an encounter was present at that hour we classified it as 1 otherwise as 0. To determine if an encounter is present we counted the number of weeks that had an encounter at that specific hour and day of the week; we only counted it as encountered if it was bigger than half of the number of weeks.

Because the St. Andrews trace had encounters at every hour of day and night we decided to add another variable, the number of encountered nodes at that hour/day of the week. We used as training data the median number of encountered nodes at that hour/day of the week over all the weeks in the training set. This was done to provide efficiency, by lowering the amount of training data. Similar results can be obtained by using all the logs independently.

During our experiments we used the settings from Table I for the Gaussian Process.

TABLE I. GAUSSIAN PROCESS SETTINGS

Variable	Value
Noise	1
Filter Type	Normalize training data
Kernel	PUK (Pearson VII)
Omega	0.5
Sigma	0.5

To validate our prediction we compared it to the real data in the test set. We also used Cross Correlation and Pearson's chi-squared test [20].

4.2. Experimental Results

The first experimental results, shown in Figure 1, present a comparison between the prediction and the real data from the test set. The experiment itself trains the Gaussian Process from the training set with hours/day of the week at which encounters appear.

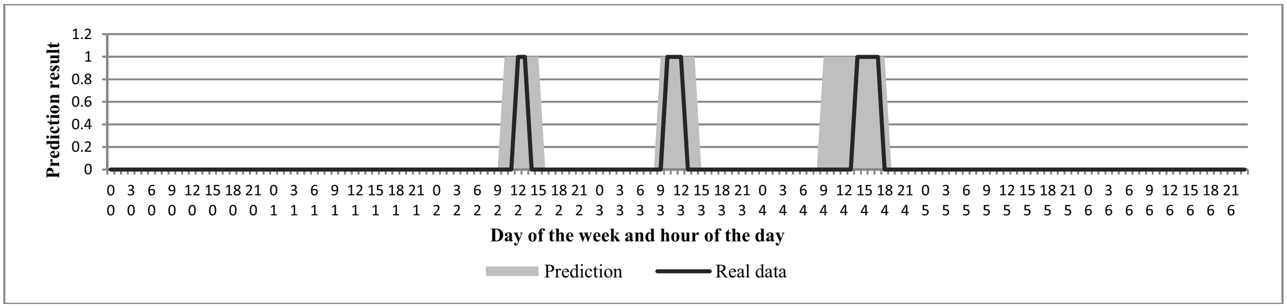

Fig. 1 UPB 2012 - Encounter Comparison

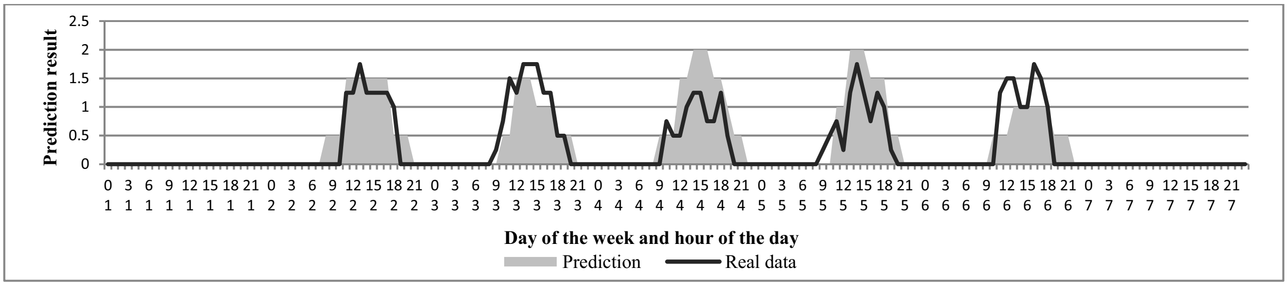

Fig. 2 UPB2012 - Number of Encountered Nodes Comparison

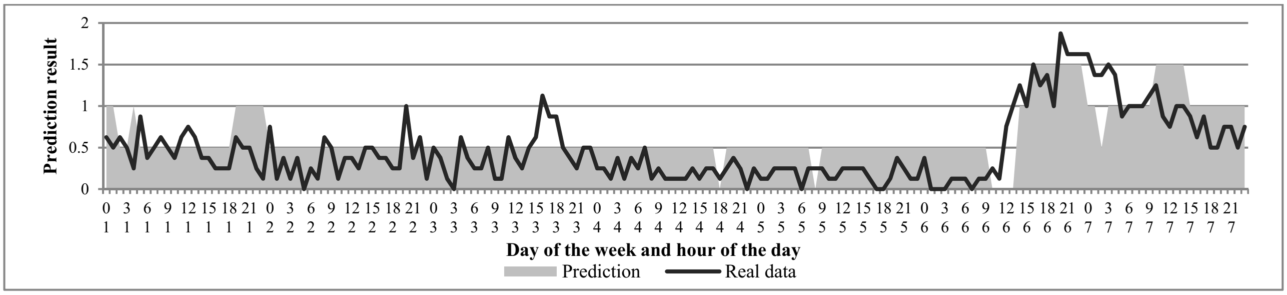

Fig. 3 St. Andrews-SASSY - Number of Encountered Nodes Comparison

It is visible how the prediction matches closely the actual encounters in the test set. The graphic in Figure 1 presents the results for a randomly-chosen node in the set, but the others show similar results.

The next experiment was run on both the UPB2012 trace and the standrews-sassy trace. The results for one node can be observed in Figure 2 and 3. The results were obtained using a standard binary classification. The training data itself uses 3 variables: the hour of the day, the day of the week and the number of encountered nodes. The last one is obtained as a mean over the number of encountered nodes at the specific hour/day of week inside the weeks in the training set. The classification itself is in fact binary. What the graph represent is a mean over the number of encounters it classifies in the "1" class. This is also why fractional numbers appear in the graph.

Different views of the same data can be seen in Figures 4, 5 and 6. In this figures different nodes were chosen than the ones in the previous figures (but, again, randomly). The figures themselves represent only the predicted number of nodes over all the hours of the day and all days of the week. This is the mean over all the number of nodes predicted for every specific hour/day of the week combination.

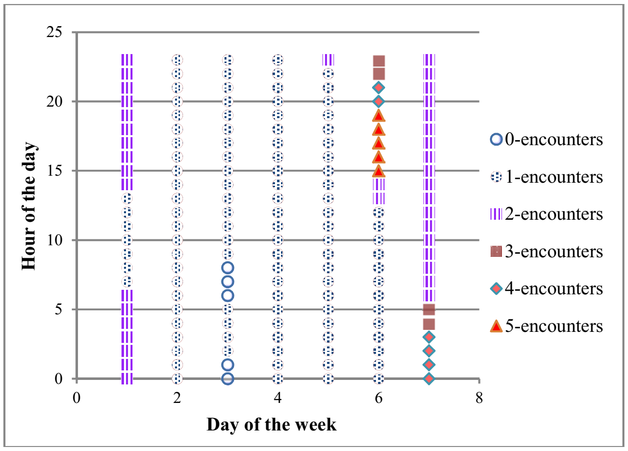

Fig. 4 St. Andrews SASSY - Number of Encountered Nodes Prediction

Unlike the UPB 2012 trace in which most encounters happen during weekdays at specific hours we can see that the encounters in the St Andrews trace happen during all hours of a day and even in the weekends.

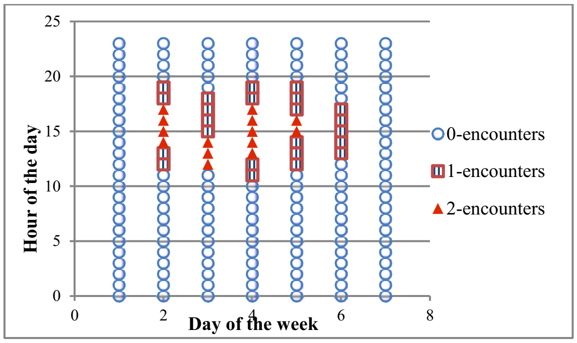

Fig. 5 UPB2012 - Node 37 - 50% Training Data - Number of Encountered Nodes Prediction

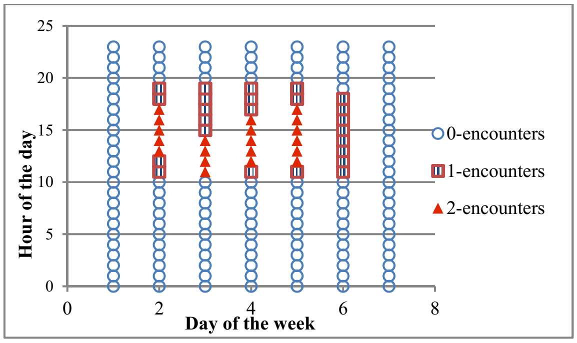

Fig. 6 UPB2012 - Node 37 - 80% Training Data - Number of Encountered Nodes Prediction

Figures 5 and 6 present the prediction of the number of encountered nodes for the same node (node with ID 37). The first figure represents the prediction after only 50% of the training data has been used while the other uses 80% of the data for the prediction. We can notice that even though the graphics retain most of the same structure there are subtle changes between them. These differences appear because of changes in the nodes behavior. In our case, for a node representing a student, the change is probably an indication that s/he managed to regulate his/her schedule and perhaps stops being late to some classes. The previously mentioned figures show that just training the Gaussian Process at some point in time is not enough for a real life application and that periodic or event triggered training sessions should be conducted. This confirms our hypothesis that persons change their behavior in time and as such they change their encounter patterns.

Previous methods like the use of Poisson [3] or a new prediction algorithm inspired from the decomposition of a complex wave into simpler waves with fixed frequencies (similar to Fourier decomposition) [21]. The first method is not suited for dynamic environments; as such our solution is superior in this matter. The second one can be used by the resource management systems, which can be dynamic, in order to improve the scheduling decisions, also for complex workflows [22], assuring the load balancing and optimizing the resource utilization.

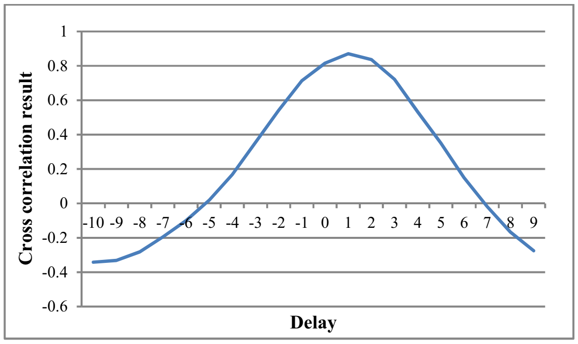

Fig. 7 Cross Correlation

Figures 7 and 8 present the validation results for the UPB 2012 trace. Figure 7 has a Cross Correlation between the predicted number of nodes to be encountered and the actual encounters. The closer the result gets to 1 the more correlated the data sets are. Cross Correlation is calculated with different delays. One can observe that the center is not at 0, as such the data is actually better correlated if it is delayed with 1 unit. Next, in Figure 8 we used a Pearson chi-squared test to validate our prediction.

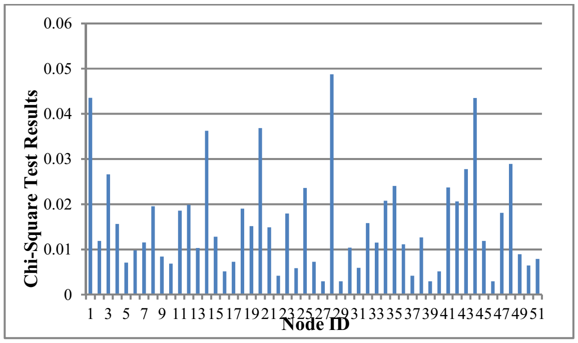

Fig. 8 Chi-Squared Test

In case of the chi-squared test the closer the result is to 0 the closer the prediction is to the actual results. The chi-squared test was run on all the nodes and as such we can observe differences between the accuracy of the prediction between nodes. This is expected as some nodes tend to be more predictable in their movement than other.

5. Conclusion

In this paper we presented a novel method for predicting a person's behavior in support for communication in an opportunistic network. Our method uses data gathered by mobile devices to determine the number of future encounters a person is likely to have with other persons and the hours of the day, days of the week at which this encounters are more likely to happen. Extensive validations of the proposed prediction method were executed using real-world data traces, UPB 2012 and standrews-sassy. We validated our findings using chi-squared test and cross correlation tests. We believe this method of obtaining predictions is useful in the case of an Opportunistic Mobile Network as any a priori knowledge about the movement of the individuals involved in the network can help in improving the routing decisions made by the mobile devices. As future work we plan to use this method to improve the SAP [2] algorithm, previously proposed in UPB, and to test our approach on more traces to further prove its validity.

Acknowledgments

This work was supported by project "ERRIC - Empowering Romanian Research on Intelligent Information Technologies/FP7-REGPOT-2010-1", ID: 264207. The work has been co-funded by the Sectoral Operational Programme Human Resources Development 2007-2013 of the Romanian Ministry of Labour, Family and Social Protection through the Financial Agreement POSDRU/89/1.5/S/62557.

6. References

[1] P. Hui, J. Crowcroft, E. Yoneki. "*Bubble rap: social-based forwarding in delay tolerant networks*". In 9th ACM int. symp. on Mobile ad hoc netw. and comp. (MobiHoc'08), pp 241–250, New York, USA, 2008.

[2] R. I. Ciobanu, "*Routing in Opportunistic Networks Using Contact Predictions and Social Connections*", Master Thesis, defended 2012, UPB, Romania.

[3] R. I. Ciobanu, C. Dobre, "*Predicting encounters in opportunistic networks*", In 1st ACM workshop on High performance mobile opportunistic systems (HP-MOSys '12). ACM, New York, NY, USA, pp. 9-14, 2012.

[4] S. Jain, K. Fall, R. Patra. "*Routing in a delay tolerant network*". In 2004 conf. on Applications, technologies, architectures, and protocols for computer comm., SIGCOMM '04, pp. 145–158, New York, NY, USA, 2004.

[5] L. Pelusi, A. Passarella, M. Conti. "Opportunistic networking: data forwarding in disconnected mobile ad hoc networks". *IEEE Comm. Magazine*, 44(11):134–141, 2006.

[6] Pan Hui and Jon Crowcroft. "*Predictability of human mobility and its impact on forwarding*". In 2008 3rd Int. Conf. on Comm. and Netw. in China, pp. 543–547, 2008**.**

[7] M. Conti, S. Giordano, M. May, A. Passarella. "*From opportunistic networks to opportunistic computing".* Comm. Mag., 48:126–139, 2010.

[8] H. A. Nguyen, S. Giordano, A. Puiatti. "*Probabilistic Routing Protocol for Intermittently Connected Mobile Ad hoc Network*". In 2007 IEEE Int. Symp. on a World of Wireless Mobile and Multimedia Networks, pp. 1–6, 2007.

[9] C. Boldrini, M. Conti, J. Jacopini, and A. Passarella. "*HiBOp: a History Based Routing Protocol for Opportunistic Networks*". In World of Wireless, Mobile and Multimedia Netw. (WoWMoM), pp. 1–12, 2007.

[10] Nathan Eagle and Alex Pentland, "*Reality mining: sensing complex social systems*," Personal and Ubiquitous Computing, vol. V10, no. 4, pp. 255–268, May 2006

[11] A. Lindgren, A. Doria, O. Schelen. "*Probabilistic routing in intermittently connected networks*". SIGMOBILE Mob. Comput. Commun. Rev., 7(3):19–20, July 2003.

[12] David J. C. Mackay "*Gaussian Processes – A Replacement for Supervised Neural Networks?*", Tutorial lecture notes for NIPS 1997, Denver, CO, USA, 1997.

[13] Community Resource for Archiving Wireless Data At Dartmouth (CRAWDAD) official website, last accessed February 8, 2013, from http://crawdad.cs.dartmouth.edu/

[14] L. Pelusi, A. Passarella, and M. Conti, "*Beyond MANETs: dissertation on Opportunistic Networking*", IITCNR Tech. Rep., May 2006.

[15] R. I. Ciobanu, C. Dobre, "*Social Aspects to Support Opportunistic Networks in an Academic Environment*", In Proc. of ADHOC-NOW 2012, Springer-Verlag Berlin Heidelberg 2012, LNCS 7363, pp. 69–82, 2012.

[16] A. Asandei, C. Dobre, P. Johnson, "*An analysis of techniques for opportunistic networking*," IEEE Int. Conf. on Intelligent Computer Comm. and Processing (ICCP 2012), Cluj-Napoca, Romania, pp. 341-348, 2012.

[17] B. Ustun, W.J. Melssen, LMC Buydens. "*Facilitating the application of Support Vector Regression by using a universal Pearson VII function based kernel*". Chemometrics and Intelligent Laboratory Systems, 81(1):29-40, 2006.

[18] C. E. Rasmussen & C. K. I. Williams, "*Gaussian Processes for Machine Learning*", MIT Press, 2006.

[19] M. Hall, E. Frank, G. Holmes, et al, "*The WEKA Data Mining Software: An Update*", IGKDD Explor. Newsl., 11 (1), pp. 10-18, November 2009.

[20] R.L. Plackett, "*Karl Pearson and the Chi-Squared Test*", Int. Statistical Review, 51(1983), pg. 59-72.

[21] M. Istin, A. Vișan, F. Pop, V. Cristea, "*Decomposition Based Algorithm for State Prediction in Large Scale Distributed Systems*," Parallel and Distributed Computing (ISPDC), Ninth Int. Symp. on , vol., no., pp.17,24, 7-9 2010

[22] B. Simion, C. Leordeanu, F. Pop, V. cristea, "*A Hybrid Algorithm for Scheduling Workflow Applications in Grid Environments (ICPDP)*", Lecture Notes in Computer Science Volume 4804, 2007, pp 1331-1348