

Multi-Periodic Contact Patterns in Predicting Future Contacts Over Mobile Networks

Sima Das and Sajal K. Das
Department of Computer Science
Missouri University of Science and Technology
Rolla, Missouri 65409.
{sdp4b, sdas}@mst.edu

Abstract—With advances in Internet technology and prominence of mobile and smart devices in our lives, opportunistic and pervasive networks are now ubiquitous in solving many existing service limitations. The challenge lies in the underlying time-varying graph of the network due to mobility and intermittent connectivity. This introduces technical limitations in successful realization of services and applications e.g., efficient routing, maximal coverage with minimal latency, data offloading, and effective dissemination over mobile networks. Efficient solution to these inter-related problems lies in the novel prediction strategies for most accurate future contacts (i.e., links or interactions). In contrast to the existing strategies that consider either network structure or regular pattern and periodic nature of contacts, we propose novel use of seasonal autoregressive integrated moving average model and recurrent neural network model that are capable of capturing multi-periodic, dependent contact patterns. We predict the number of contacts relative to a node and over all nodes in any future interval over a given user and a pair of users. Finally, we validate our models with three distinct empirical data set, and compare with doubly recurrent and homogeneous Poisson process model to demonstrate the superiority of our prediction models.

I. INTRODUCTION

Mobile networks include ad-hoc wireless networks, mobile-social networks, vehicular networks, as well as opportunistic networks at large that usually fall under the domain of pervasive networks [1], [2], [3], [4]. The dynamic (intermittent) nature of contacts (also called links or interactions) over pervasive networks is caused due to mobility and disruptions due to environment of the interacting element and studied over temporal domain. The problem of predicting future interactions is synonymous with determining neighbors of a node over successive intervals with implication in maintaining connectivity, routing over the network, and information dissemination, to mention a few. Over the mobile networks or mobility induced intermittently connected dynamic networks, connectivity among network elements is essential for successful realization of a wide range of applications [5], [6], [7], [8], [9]. In this context, the future connectivity among the mobile or participating devices sheds light on the feasibility of these applications, their efficient planning and successful execution.

Example pervasive networks, such as mobile social networks

[10], [11], [12], [13], opportunistic networks [14], vehicular networks [15] or mobile sensor networks [16] have the ad-hoc wireless network as its core, formed over the underlying devices that interact using the store-carry-forward paradigm [15], [17]. The proliferation of mobile devices, smart technology and wide scale use of the Internet paves the way for the realization of smart city based services such as smart health care, smart management, mobile virtual cloud, etc. The success of these services relies on the successful collaboration among the underlying devices. Because of the mobility factor associated with the constituting elements, disconnection and variable delays are an integral part of such networks. In this context, connectivity (or contact or interaction) between nodes is viewed as routing data closer to the destination.

The devices underlying the ad-hoc wireless networks are being carried with humans, hence associated with human mobility. Thus, not surprisingly, these participating devices exhibit a periodic pattern of connectivity among them, owing to the fact that human mobility pattern is proved to exhibit intrinsic periodicity [18], [19]. The challenge in pervasive networks is to determine whether a node can pass the data (more specifically, when and whom to have the best possible hit rate/coverage and minimize latency, or the set of nodes that will collaborate to have successful realization of the application task. In other words, the number of nodes that a node comes in contact with at any time is the backbone of many application tasks. Thus, predicting the future interactions are core to the success of these applications; and thus analyzing the contact history is an integral process in predicting future contact patterns. We consider the mobility pattern underlying the pervasive networks as the basis of our proposed prediction model.

To set the stage for our novel contributions, we first present an overview of the literature on contact prediction using contact patterns, and describe the motivation behind our work.

A. Related Work and Motivation

In order to discover significant statistical patterns of human mobility, walk traces of humans from mobile networks have been analyzed in the literature. For example, it is established that individuals (hence the associated mobile devices) return to a few highly frequented locations [20], [18]. This implies

that they come in contact with other mobile devices frequently; thus inducing connectivity and collaboration, and influencing connection times and opportunities between participants.

A common notion to model the inherently regular contact patterns is to use the *Poisson* process models, as in [21] that replicate the existing interactions and predict the future interactions using homogeneous Poisson process (HPP) model. Contact patterns in reality have non-uniform periods hence the assumption of regular contact pattern and HPP model fail to capture the associated non-homogeneity.

Further, to capture the periodic contact pattern, the authors in [22], [23] used doubly periodic property with short (daily) and long (weekly) duration contact events and proposed stochastic double chain hidden Markov model (HMM) to analyze the communication/contacts over node pairs using email and MIT [24] and UCSD [25] data sets, respectively.

Among the statistical and soft computing based techniques used as prediction methodology, the autoregressive integrated moving average (ARIMA) (besides the statistical HMMs) and artificial neural network (ANN) based models are the most widely used. The stochastic ARIMA process models are used in time series analysis due to Box and Jenkins [26], though the series are assumed to be generated from linear processes and as a result might be inappropriate for most real-world problems that are nonlinear [27], [28]. Recall that ANNs are known to provide competitive results for various traditional time series models such as ARIMA [29] where the time series data has nonlinearity. In the literature, there is widespread use of both ARIMA and ANN models in forecasting, such as in stock market, electricity consumption and tariff, road traffic etc. For example, the authors in [30] used ARIMA and genetic algorithms for short term traffic flow prediction, whereas in [31] the authors used ANN for real time vehicle speed prediction. The dynamic, time-varying nature of real world problems along with the demand for prediction accuracy is the driving force behind different prediction models.

In order to encompass the real world requirements, this paper makes the following novel contributions.

B. Summary of Contributions

First, we use seasonal ARIMA models with uncertain parameter values to reflect real world applications and consider their effectiveness in capturing periodic contact pattern and prediction ability over future contacts. Next, we consider a variant of ANNs, called Recurrent Neural Networks (RNNs) in learning the dependencies from the temporal data, capturing the non linearities involved and predicting contacts over future time intervals. Finally, we evaluate and compare the efficiency of the proposed models, with existing models for contact prediction. Here we consider both short term and long term predictions, and use three distinct temporal data sets to validate the effectiveness of prediction in our models.

The remainder of the paper is structured as follows. Section II formally defines our problem. Section III gives an overview of the ARIMA model in general along with the ARIMA model we considered for contact prediction. Section IV presents the

general overview of ANN model along with the proposed recurrent version. Section V discusses the three distinct data sets used and the parameter estimation in both the proposed models. Section VI reports the experimental results and finally conclusions are offered in Section VII.

To the best of our knowledge, ours is the first work to consider Seasonal ARIMA with uncertain parameter values and Recurrent Neural Networks over time-varying graphs of contact patterns (i.e., over time series data) in predicting future contacts.

II. PROBLEM DEFINITION

We consider the underlying time-varying network graph over a time interval $[0, T]$ as $G_T(V, E, [0, T])$, where V and E represent the set of vertices and edges respectively, and $[0, T]$ is the observation period. The subgraph at any time period δt is represented as $G_{\delta t}(V_{\delta t}, E_{\delta t})$, where $V_{\delta t}$, and $E_{\delta t}$ are the corresponding vertices and edges. Corresponding to real world applications, where a contact is established between a pair of mobile devices/nodes within the communication range, an edge is established in the underlying network graph between a pair of nodes, i.e., $(v_i, v_j) \in E$ at time instance t , called *contact time* (or contact initiation time), where v_i, v_j represent the respective mobile nodes in the application network. The contact pattern over the set of $G_{\delta t}$ s in G_T represent the time series data of the contacts among mobile devices.

Let the contact remain active in time $[t, t + \delta t)$, then δt is called the contact period/duration for the corresponding node pair, where $t \geq 0, \delta t > 0$. The number of contacts that a node v_i has in time interval δt , i.e., $\forall v_j : (v_i, v_j) \in E_{\delta t}$ is called its neighbors. In fact, for any node v_i , we predict its number of contacts during time period $[t, t + \delta t)$.

III. FORECASTING WITH ARIMA MODEL

In this section we discuss the ARIMA model in general and the proposed variant of this ARIMA model for contact prediction. Before proceeding further, let us define time series data, which is equally applicable to both ARIMA and ANN models.

Time Series Data: A set of values of a variable (observation variable outcomes) during consecutive time increments is termed as time series data. The time increments themselves vary over different series. In other words, time series can be generated according to hourly, daily, weekly, monthly, quarterly, or annually with respect to the data or according to another time scale. Over any time series we generally represent the observed data at any time t by O_t . Time series modeling (or the stochastic modeling) techniques are used to analyze time series data and provide a systematic and empirical method for simulation and forecasting. In most time series data, the observations exhibit significant serial correlation, which is effectively captured by the ARIMA model. In addition, there is also non-linear correlation, both of which are well handled by the ANN model variants.

A. ARIMA Prediction Process: General Model

The most comprehensive of all popular statistical methods used for time series forecasting are Box-Jenkins technique, which is based on discrete and linear stochastic processes. The usual statistical forecasting models in this methodology are: autoregressive (AR), moving average (MA), autoregressive-moving average (ARMA), and autoregressive integrated moving average (ARIMA). While $AR(k)$, $MA(m)$ and a combination of these $ARMA(k, m)$ are used in stationary series, $ARIMA(k, d, m)$ models are used for non-stationary series, where k is the number of autoregressive terms, d is the number of non-seasonal differences needed for stationarity, m is the number of lagged forecast errors in the prediction equation. The ARIMA process model is composed of three components: an AR model as a combination of past values, an MA component that uses past forecast errors in a regression like model, and an integration (I) part that refers to the reverse process of differencing to produce the predictions. To capture the cyclic features (seasonality) in time series, $SARIMA(k, d, m)^s$ model is used, where (k, d, m) belongs to the non-seasonal part as before and K is the number of seasonal autoregressive (SAR) terms, D is the number of seasonal differences, M is the number of seasonal moving average (SMA) terms, and s represents the number of periods per season. In ARIMA model, it is assumed that the future value of a variable is assumed to be a linear function of several past observations and random errors.

In ARIMA model, the forecasting equation can be constructed as follows. Let o represents the d^{th} difference of O , that is:

- If $d = 0$: $o_t = O_t$
- If $d = 1$: $o_t = O_t - O_{t-1}$
- If $d = 2$: $o_t = (O_t - O_{t-1}) - (O_{t-1} - O_{t-2})$
 $= O_t - 2O_{t-1} + O_{t-2}$

Now to simplify things, we consider stationary time series with $d = 0$; thus, the underlying process that generates the time series has the following form.

$$o_t = \theta_0 + \theta_1 o_{t-1} + \theta_2 o_{t-2} + \dots + \theta_k o_{t-k} + \phi_0 + \phi_1 \epsilon_{t-1} + \phi_2 \epsilon_{t-2} + \dots + \phi_m \epsilon_{t-m}, \quad (1)$$

where o_t and ϵ_t are respectively the actual value and random error at time period t ; θ_i ($i = 0, 1, \dots, k$) are the non-seasonal autoregressive coefficients corresponding to a stable polynomial, o_{t-i} s are the associated lags to the degree of the autoregressive polynomial k and ϕ_j ($j = 0, 1, \dots, m$) are nonseasonal moving average coefficients, with both θ_i s and ϕ_i s being counted as the model parameters. Note that k and m are integers and are referred as model orders. Random errors or associated lags, ϵ_t are assumed to be independently and identically distributed with mean 0 and constant variance σ^2 to the degree of the non-seasonal moving average polynomial m , with ϵ_t s as the past forecast error between real and predicted o_t s. In fact Equation 1, becomes an AR model of order k when $m = 0$ and reduces to an MA model of order m when $k = 0$. Determining the appropriate model order is one of

the central task of ARIMA model building in general, which has fundamental effect on time series analysis and forecasting applications. It consists of three iterative steps of model identification, parameter estimation and diagnostic checking. Model identification builds on the idea that if a time series is generated from ARIMA process, then it should have some theoretical autocorrelation properties. The matching of theoretical and empirical autocorrelation patterns enables the identification of viable models for the given time series. Once a suitable model is specified, the next step is to estimate the model parameters so that the overall error measure is minimized and it is done with nonlinear optimization procedure. In the last step the model assumptions about errors are checked for model adequacy, else another tentative model with the above steps is repeated.

In the following subsection, we describe the class of ARIMA model used here (a variant of *fuzzy-ARIMA* model) and the steps involved with an example.

B. Proposed ARIMA Model

In formulating the variant of ARIMA model the following steps are followed.

- Step 1: A class of models is formulated assuming certain hypotheses.
- Step 2: A model is identified for the observed data.
- Step 3: The model parameters are estimated.
- Step 4: If model hypotheses are met: Go to Step 5. Else continue with step 2 to refine the model.
- Step 5: The desired prediction model is generated.

Each of the above steps is described below.

Step 1:

Here a general ARIMA process is selected to model the time series data. This selection is carried out by careful inspection of the main characteristics of the time series data (say, hourly data values). In most of the pervasive or opportunistic network data sets, the respective time series present high frequency, varying mean and variance, and multiple periodicity (corresponding to daily, weekly, monthly) among others. If k_t denotes the data value at time t , the proposed general ARIMA formulation is given by,

$$\theta(B)k_t = \phi(B)\epsilon_t, \quad (2)$$

where $\theta(B)$ and $\phi(B)$ are functions of the backshift operator B such that $B^l p_t = p_{t-l}$, and ϵ_t is the error term. Functions $\theta(B)$ and $\phi(B)$ have special forms. They contain factors of polynomial functions of the form $\theta(B) = 1 - \sum_{l=1}^{\theta} \theta_l B^l$, $\phi(B) = 1 - \sum_{l=1}^{\phi} \phi_l B^l$ and $(1 - B^s)$, where several values of ϕ_l and θ_l can be set to 0. For example, the function $\theta(B)$ may have the following form:

$$\theta(B) = (1 - \theta_1 B^1 - \theta_2 B^2)(1 - \theta_{24} B^{24} - \theta_{48} B^{48}) \times (1 - \theta_{168} B^{168})(1 - B)(1 - B^{24}). \quad (3)$$

The above process model example does not reflect a standard ARIMA model, rather it is sufficient to capture the main features of the time series data over the said application

domain. Further, certain hypotheses on the error term ϵ_t must be imposed. In Step 1, this term is assumed to be randomly drawn from a normal distribution with mean 0 and constant covariance σ^2 , which is a white noise process. In Step 3, a diagnosis check is used to validate these model assumptions.

Step 2:

A tentative model as in Equation 4 below, must be identified for the time series data. In order to make the underlying process stationary, a transformation of original data and the inclusion of factors of the form $(1 - B^s)$ may become necessary. In this step a logarithmic transformation is usually applied to the time series data to attain a more stable variance. Further, to attain a more stable mean, factors of the form $(1 - B)$, $(1 - B^{24})$, $(1 - B^{168})$ may be necessary depending on the specific application type of time series data. After the underlying process is accepted as being stationary, the structure of functions $\theta(B)$ and $\phi(B)$ in Equation 4 must be detected. In the first trial, the observation of autocorrelation and partial autocorrelation plots of the time series data help make this selection. Over successive trials, the observations from the difference between observed value and predicted value obtained in Step 4 is used to refine the structure of the functions in the proposed model.

Step 3:

Once the functions of the model have been specified, the parameters need to be estimated. The parameter estimation is based on maximizing a likelihood function for the available data. A conditional likelihood function is selected in order to get a good starting point to obtain an exact likelihood function. Further, an option to detect and adjust possible unusual observations or outliers is selected. It is necessary to have a procedure to detect and minimize outliers is necessary, as these events are not known initially. A better understanding of the data set, a better modeling and estimation, and finally a better forecasting performance is achieved with these adjustments.

Step 4:

Here, a diagnostic check is used to validate the model assumption in Step 1. The diagnosis ensure that the difference between the predicted and actual values satisfy zero mean, constant variance, uncorrelated process and normal distribution with the help of autocorrelation and partial autocorrelation plots.

Step 5:

In this step, the model from Step 3 is used to predict future values over the time series. As an outcome of these five steps, the resulting functions in the prediction model will be given by:

$$\begin{aligned} & (1 - \theta_1 B^1 - \theta_2 B^2 - \theta_3 B^3 - \theta_4 B^4 - \theta_5 B^5) \\ & \times (1 - \theta_{23} B^{23} - \theta_{24} B^{24} - \theta_{47} B^{47} - \theta_{48} B^{48} - \\ & \theta_{72} B^{72} - \theta_{96} B^{96} - \theta_{120} B^{120} - \theta_{144} B^{144}) \\ & \times (1 - \theta_{168} B^{168} - \theta_{336} B^{336} - \theta_{504} B^{504}) \log p_t \\ & = c + (1 - \phi_1 B^1 - \phi_2 B^2)(1 - \phi_{24} B^{24}) \\ & (1 - \phi_{168} B^{168} - \phi_{336} B^{336} - \phi_{504} B^{504}) \epsilon_t \end{aligned} \quad (4)$$

Equation 4 reflects the use of last five hours for prediction over the subsequent hour and does not include the differentiation over hourly, daily, weekly time periods. Including these factors $((1 - B)(1 - B^{24})(1 - B^{168}))$ into the above will give us the following:

$$\begin{aligned} & (1 - \theta_1 B^1 - \theta_2 B^2 - \theta_3 B^3 - \theta_4 B^4 - \theta_5 B^5) \\ & \times (1 - \theta_{23} B^{23} - \theta_{24} B^{24} - \theta_{47} B^{47} - \theta_{48} B^{48} - \\ & \theta_{72} B^{72} - \theta_{96} B^{96} - \theta_{120} B^{120} - \theta_{144} B^{144}) \\ & (1 - \theta_{167} B^{167} - \theta_{168} B^{168} - \theta_{169} B^{169} - \theta_{192} B^{192}) \\ & \times (1 - B)(1 - B^{24})(1 - B^{168}) \log p_t \\ & = c + (1 - \phi_1 B^1 - \phi_2 B^2) \\ & \times (1 - \phi_{24} B^{24} - \phi_{48} B^{48} - \phi_{72} B^{72} - \phi_{96} B^{96}) \\ & \times (1 - \phi_{144} B^{144}) \\ & \times (1 - \phi_{168} B^{168} - \phi_{336} B^{336} - \phi_{504} B^{504}) \epsilon_t \end{aligned} \quad (5)$$

The above can be generalized over to the Seasonal ARIMA version to capture the multi periodicity with SARIMA $(k, d, m)(K, D, M)_s$ with parameters $\theta_1, \dots, \theta_k, \phi_1, \phi_2, \dots, \phi_m, \Theta_1, \dots, \Theta_k, \Phi_1, \dots, \Phi_m$, which have exact values. In real world we often have to predict future with limited amount of data over short period of time and the above model addresses the limitations of real world applications. As described above, the model uses the concept of measurement error to handle the difference between predicted and observed data values. But these data values are correct and do not include measurement errors. Instead of exact values, we propose using fuzzy parameter values and define the new set of parameters in the form of triangular fuzzy numbers as: $\tilde{\theta}_1, \dots, \tilde{\theta}_k, \tilde{\phi}_1, \dots, \tilde{\phi}_m, \tilde{\Theta}_1, \dots, \tilde{\Theta}_k, \tilde{\Phi}_1, \dots, \tilde{\Phi}_m$.

Thus the resulting fuzzy functions for the SARIMA model will be as follows.

$$\tilde{\theta}(B)\tilde{\Theta}(B^s)W_t = \tilde{\beta}_0 + \tilde{\phi}(B)\tilde{\Phi}(B^s)a_t, \quad (6)$$

where a_t has fuzzy parameter γ and $W_t = (1 - B)^d(1 - B^s)^D Z_t$.

Thus, we can derive $\tilde{W}_t$ as:

$$\begin{aligned} \tilde{W}_t &= \tilde{\beta}_0 + \sum_{i=1}^k \tilde{\phi}_i W_{t-i} + \sum_{i=1}^K \tilde{\Theta}_i W_{t-i} - \tilde{\gamma} a_t \\ &- \sum_{i=1}^p \tilde{\phi}_i \Phi_1 W_{t-s-i} - \sum_{i=1}^K \tilde{\phi}_i \Phi_2 W_{t-i-2s} \\ &- \dots - \sum_{i=1}^k \tilde{\phi}_i \Phi_K W_{t-Ks-i} + \tilde{\gamma} a_t \\ &- \sum_{i=1}^m \tilde{\theta}_i a_{t-i} - \sum_{i=1}^Q \tilde{\Theta}_i a_{t-is} \\ &+ \sum_{i=1}^m \tilde{\theta}_i \Theta_1 a_{t-s-i} + \sum_{i=1}^m \tilde{\theta}_i \Theta_2 a_{t-2s-i} \\ &+ \dots + \sum_{i=1}^m \tilde{\theta}_i \Theta_Q a_{t-Ms-i}, \end{aligned} \quad (7)$$

where Z_t are observations and the center value of γ is 1. Now we rewrite Equation 7 with fuzzy parameters in the form of triangular fuzzy numbers as follows.

$$\begin{aligned} \tilde{W}_t = & \tilde{\beta}_0 + \sum_{i=1}^k \tilde{\beta}_i W_{t-i} + \sum_{i=1}^K \tilde{\beta}_{k+i} W_{t-is} \\ & - \sum_{j=1}^K \sum_{i=1}^k \tilde{\beta}_i \tilde{\beta}_{k+j} W_{t-js-i} + \tilde{\beta}_{k+K+1} a_t \\ & - \sum_{i=1}^m \tilde{\beta}_{k+K+1+i} a_{t-i} \\ & - \sum_{i=1}^M \tilde{\beta}_{k+K+m+1+i} a_{t-is} + \sum_{j=1}^M \sum_{i=1}^m \\ & \tilde{\beta}_{k+K+1+i} \tilde{\beta}_{k+K+m+1+j} a_{t-js-i}. \end{aligned} \quad (8)$$

The membership function of the fuzzy set represented by parameter β_i, α_i . The center of the fuzzy number given by,

$$\mu_{\tilde{B}_i}(\beta_i) = \begin{cases} 1 - |\alpha_i - \beta_i|/c_i & \text{if } \alpha_i - c_i \leq \beta_i \leq \alpha_i + c_i \\ 0 & \text{otherwise,} \end{cases}$$

where c_i is the spread around the fuzzy number.

Using approximation formula from Dubois et al. [32], we get fuzzy multiplication $A_i \otimes A_j \cong (c_i c_j, a_i a_j, b_i b_j)$, where $A_i = (c_i, a_i, b_i)$ and $A_j = (c_j, a_j, b_j)$ are triangular fuzzy numbers. In Equation 8 the fuzzy multiplication is given by $\tilde{\beta}_i \tilde{\beta}_{p+j} = (c_i c_{p+j}, \alpha_i \alpha_{p+j}, c_i c_{p+j})$.

Using fuzzy parameters β_i in the form of triangular fuzzy numbers, the membership of W in Equation 8 is given by,

$$\mu_{\tilde{W}}(W_t) = \begin{cases} 1 - |W_t - E_t|/F_t & \text{for } W_t \neq 0, a_t \neq 0, \\ 0 & \text{otherwise,} \end{cases}$$

where

$$\begin{aligned} E_t = & \alpha_0 + \sum_{i=1}^k \alpha_i W_{t-i} + \sum_{i=1}^K \alpha_{k+i} W_{t-is} \\ & - \sum_{j=1}^K \sum_{i=1}^k \alpha_i \alpha_{k+j} W_{t-js-i} + a_t \\ & - \sum_{i=1}^m \alpha_{k+K+1+i} a_{t-i} - \sum_{i=1}^M \alpha_{k+K+m+1+i} a_{t-is} \\ & + \sum_{j=1}^M \sum_{i=1}^m \alpha_{p+P+i} \alpha_{p+P+q+j} a_{t-js-i} \end{aligned} \quad (9)$$

$$\begin{aligned} F_t = & c_0 + \sum_{i=1}^k c_i |W_{t-i}| + \sum_{i=1}^K c_{k+i} |W_{t-is}| \\ & + \sum_{j=1}^K \sum_{i=1}^k c_i c_{k+j} |W_{t-js-i}| + c_{k+K+1} |a_t| \\ & + \sum_{i=1}^m c_{k+K+1+i} |a_{t-i}| + \sum_{i=1}^M c_{k+K+m+1+i} |a_{t-is}| \\ & + \sum_{j=1}^M \sum_{i=1}^m c_{k+K+1+i} c_{k+K+m+1+j} |a_t - js - i| \end{aligned} \quad (10)$$

Here, Z_t represents the t^{th} observation and h -level is the threshold value representing the degree to which the model should satisfy all the data points $Z_1, Z_2, \dots, Z_t$. The choice of h value influences the width c of the fuzzy parameters, i.e., $Z_t \geq h$ for $t = 1, 2, \dots, k$. The index t refers to the number of non fuzzy data used for constructing the model. Whereas, the fuzziness S included in the model is defined by Equation 10.

To evaluate the parameters of the proposed model, we propose the approach comprising of two basic phases that let the number of constraint functions remain the same as the number of observations. The two phases are as follows:

Phase I: Determining the type of model using observations. Fitting the periodic ARIMA model with the available information about the center points of the observations, i.e., the input data are considered precise. The results of Phase I, the optimum solution of the parameter $\alpha^* = (\alpha_1, \alpha_2, \dots, \alpha_{k+K+m+M})$, is used as one of the data sets in Phase II.

Phase II: Determining the minimal fuzziness by using the above criterion of $Z(Z_t) \geq h$, but without α being a vector of decision variables. The proposed model is in fact an interval prediction model. The model given by: $\tilde{W}_t$, as a combination of Equation 9 and Equation 10. In order for the model to include all conditions, when data include a significant difference, the data is processed to produce c_j with wide spread.

IV. FORECASTING WITH ARTIFICIAL NEURAL NETWORK

Reflecting the functioning of human brain, Artificial Neural Network (ANN), a nonlinear modeling technique is a combination of layers made of neurons, with the multi-layer perceptron (MLP) being widely used for forecasting. When the MLP used for time series forecasting, determining the input, output and hidden neurons or variables is important, although these parameters are problem dependent. Below we present a variant of ANN that very well captures the recurrent patterns and is call Recurrent Neural Network (RNN) model.

A. RNN Prediction Process: General Model

The number of input neurons is one of the most important parameters of an MLP for forecasting. It is not easy to determine the suitable number of input neurons when dealing with a time series forecasting problem. The proper number of neurons is usually achieved through heuristic approaches. The number

of hidden layers and neurons are significant to the efficiency of ANN, as they bring out the defining properties within data and help establish the non-linear relationship between the input and output, in order to help achieve better accuracy. Though there can be more than one hidden layers in ANN, usually one hidden layer structure is used for forecasting purpose [27]. Though determining the number of hidden neurons is also a critical task, fewest neurons are preferred, as generalization is more attainable with this, rather than memorization which happens with the network having more hidden neurons. In addition, the number of output neurons also affect the performance of MLP. The number of output neurons is directly dependent on the problem and is determined based on the length of the forecast period. For example, if the forecasting horizon is one period, then the number of output neuron is one. Whereas, in a multi-periodic forecast, the suitable number of output neurons depend on the underlying approach. For example, in an iterative forecast, the number of output neurons is one, where as for a direct forecast, the number of output neurons is equal to the number of forecast periods. In fact, an iterative method is used to forecast only one period at a time, and the value of the next period is predicted using the predicted value as an input. The disadvantage is the decrease in accuracy when the number of forecasting period is large, where the estimated values rather than the observed data are used as inputs. On the other hand, the direct method only utilizes past data to forecast the future. For time series forecasting tasks, the prediction model has the general form as follows:

$$X_t = f(X_{t-1}, X_{t-2}, \dots, X_{t-p}) + e_t, \quad (11)$$

where X_t s are the set of time series data with X_t as the forecasted value for subsequent time and e_t is the error term. A variant of the ANN are the Recurrent Neural Networks (RNN) that has the ability to learn sequences as time evolves with the help of feed back loops, thus the network state over previous time step playing an important role in the network output in the current time step [33], [34], [35]. More so, as events from the past can be retained and used in current computation, the resulting network is able to produce complex, time-varying outputs in response to static inputs. In fact RNNs rarely settle in local minima without even precautions.

The general block diagram representation of RNN, with β as the damped feedback in the range of 0 to 1.

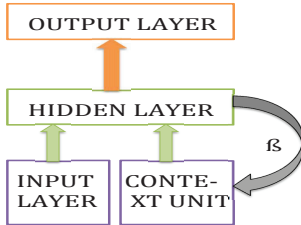


Fig. 1: Block Diagram of RNN

B. Proposed RNN Model

Here we propose a variant of RNN with recurrent back propagation. Let us describe its training methodology with a

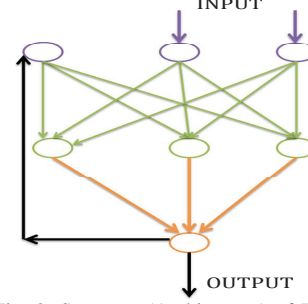


Fig. 2: Structure (Architecture) of RNN

simple example. For example, small network with two input neurons, one hidden layer having three neurons, and one output neuron. For back propagation, a neuron takes input from the output layer and connected to hidden layer. The corresponding diagram is presented in Figure 2.

Initially weights are assigned to the propagators as small random numbers. The inputs are fed to the input units and the output from the network is calculated as in a feed forward (artificial) neural network. The process is repeated over all the input. After finding the system error, the network is trained based on the steepest gradient method using ordered partial derivatives explained below. Ordered partial derivatives are derived from the concept of ordered set of equations. Let $[X_1, X_2, \dots, X_n]$ be a set of n variables, whose values are determined by a set of n equations. This set of equations is ordered if each variable X_i is a function of the previous $i - 1$ variables $[X_1, X_2, \dots, X_{i-1}]$. For example, $X_1 = 1$, $X_2 = 3X_1$, $X_3 = X_1 + 2X_2$. When calculating a partial derivative it is necessary to specify which of the variables are held constant and which ones are allowed to vary. In an ordered partial derivative, the constant and varying terms are determined using an ordered set of equations, denoted as $\partial^+ X_j / \partial X_i$, with constant terms $[X_1, X_2, \dots, X_{i-1}]$ and varying terms $[X_i, X_{i+1}, \dots, X_n]$. Thus, if $j \leq i$, $\frac{\partial^+ X_j}{\partial X_i} = 0$; if $j = i + 1$, $\frac{\partial^+ X_j}{\partial X_i} = \frac{\partial X_j}{\partial X_i}$; if $j > i + 1$, $\frac{\partial^+ X_j}{\partial X_i} = \frac{\partial X_j}{\partial X_i} + \sum_{k=i+1}^{j-1} \left(\frac{\partial^+ X_j}{\partial X_k} * \frac{\partial X_k}{\partial X_i} \right)$, or $\frac{\partial^+ X_j}{\partial X_i} = \frac{\partial X_j}{\partial X_i} + \sum_{k=i+1}^{j-1} \left(\frac{\partial X_j}{\partial X_k} * \frac{\partial^+ X_k}{\partial X_i} \right)$. Thus, for the above example, $\frac{\partial^+ X_3}{\partial X_1} = \frac{\partial X_3}{\partial X_1} + \frac{\partial X_3}{\partial X_2} * \frac{\partial X_2}{\partial X_1} = 1 + 2 * 3 = 7$. For training the network, once the system error is calculated, the weights are computed as per Equation 12.

$$w(k+1) = w(k) - \mu * \frac{\partial^+ E}{\partial w(k)}, \quad (12)$$

where $w(k)$: the weight in the current iteration k ; $w(k+1)$: the corresponding weight to be used for next iteration $k+1$; and $\frac{\delta^+ E}{\delta w(k)}$: the gradient of the system error with respect to the weight $w(k)$. Now, the procedure for computing the system error is described in the following.

$$\frac{\partial^+ E}{\partial w(k)} = \frac{\partial E}{\partial w(k)} + \sum_{k=1}^{N_p} \left(\frac{\partial E}{\partial f_{out}(k)} * \frac{\partial^+ f_{out}(k)}{\partial w(k)} + \frac{\partial E}{\partial w(k)} * \frac{\partial^+ w(k)}{\partial w(k)} \right), \quad (13)$$

where the terms $\frac{\partial E}{\partial w(k)} = 0$, because E is not a direct

function of the weights. Therefore, the equation reduces to the following.

$$\frac{\partial^+ E}{\partial w(k)} = \sum_{k=1}^{N_p} \left(\frac{\partial E}{\partial f_{out}(k)} * \frac{\partial^+ f_{out}(k)}{\partial w(k)} \right), \quad (14)$$

where $\frac{\partial E}{\partial f_{out}(k)} = (-2) * (a_{out}(k) - f_{out}(k))$. Now,

$$\begin{aligned} \frac{\partial^+ f_{out}(k)}{\partial w(k)} &= \frac{\partial f_{out}(k)}{\partial w(k)} + \sum_{j=0}^{k-1} \left(\frac{\partial f_{out}(k)}{\partial W_j(k)} * \frac{\partial^+ W_j(k)}{\partial w(k)} \right) + \\ &\sum_{j=0}^{k-1} \left(\frac{\partial f_{out}(k)}{\partial f_{out}(j)} * \frac{\partial^+ f_{out}(j)}{\partial w(k)} \right) \end{aligned} \quad (15)$$

The second term of Equation 15 is calculated using chain rule. Here, W indicates the weight vector. The term $\frac{\partial f_{out}(k)}{\partial w_j(k)}$ of the first summation is non-zero only when $k = j$. As a result, the first summation can be written as $\frac{\partial f_{out}(k)}{\partial w(k)}$. In addition, the first term of the second summation, $\frac{\partial f_{out}(k)}{\partial f_{out}(j)}$ is non-zero only when $(k - j) < L$, where L is the number of previous output values which are sent as inputs to the hidden layer. Thus, Equation 15 can be simplified as:

$$\frac{\partial^+ f_{out}(k)}{\partial w(k)} = \frac{\partial f_{out}(k)}{\partial w(k)} + \sum_{j=1}^L \left(\frac{\partial f_{out}(k)}{\partial f_{out}(k-j)} * \frac{\partial^+ f_{out}(k-j)}{\partial w(k)} \right). \quad (16)$$

This equation is used recursively to calculate the output gradient for all the patterns, which is initialized using $\frac{\partial^+ f_{out}(1)}{\partial w(k)} = \dots = \frac{\partial^+ f_{out}(L)}{\partial w(k)} = 0$.

The gradient of the error for a neuron j in the hidden layer and the neuron k in the output layer is calculated as follows.

$$\frac{\partial^+ f_{out}(k)}{\partial w(j,k)} = out1_j * \left(\frac{exp^{inp2_k}}{(1 + exp^{inp2_k})^2} \right), \quad (17)$$

where $inp2_k$ is the total input received by the neuron k for the particular pattern and $out1_j$ is the output of the neuron j for that pattern and $\frac{\partial f_{out}(k)}{\partial f_{out}(j)} = \left(\frac{exp^{inp2_k}}{(1 + exp^{inp2_k})^2} \right) * w_{rj}$, where w_{rj} is the weight which transfers the output of pattern j to the hidden layer. Once the values of the above two equations are known, the weights can be changed accordingly.

The gradient of the weights between the input layer neuron i and the hidden layer neuron j is calculated as follows.

$$\begin{aligned} \frac{\partial f_{out}(k)}{\partial w1_{i,j}} &= \left(\frac{exp^{inp2_k}}{(1 + exp^{inp2_k})^2} \right) * w2_{j,k} * \\ &\left(\frac{exp^{inp1_j}}{(1 + exp^{inp1_j})^2} * inp_{k,i} \right) \end{aligned} \quad (18)$$

Once the gradient of the error for all the weights is evaluated, the weights are changed and the process repeated until the errors converge to a pre-specified value or for a specified maximum number of iterations.

V. DATASETS AND PARAMETER ESTIMATION

To validate the efficacy of our proposed models we consider a distinct set of datasets. First, we consider the widely cited reality mining dataset from the MIT media laboratory [36] for our experiments. It is a data set collected over 94 subjects, provided with Nokia 6600 mobile phones, over time period from September 2004 to June 2005. Since, Bluetooth devices respond to each other within 5-10 meters, it resembles the contact establishment process. The scans performed over 5 minute intervals create logs with starting time of contact, duration of contact, the two phone numbers involved, i.e., the source and destination of contact.

Another data set [37] we consider is based on mobility traces over vehicular network. The data set consider two large GPS-based mobility traces. The data set contains mobility traces of taxi cabs in San Francisco, USA. It contains GPS co-ordinates of approximately 500 taxis collected over 30 days in the San Francisco bay area. Each taxi is equipped with a GPS receiver and sends a time stamped location update to a central server. The location updates are quite fine grained, with average time interval between two consecutive location updates being less than 10 sec, allowing us to accurately interpolate node positions between location updates. From this we obtain the connectivity graph of mobile nodes over temporal domain.

In addition, we use the data set [38], that collect wireless contacts traces. The tracing experiment lasted for 65 days, in March-May 2012 and has taken place at the Faculty of Automatic Control and Computer Science, University Politehnica of Bucharest. A total of 66 volunteers participated varying in terms of year and specialization: one first year Bachelor student, one third year Bachelor student, 53 fourth year Bachelor students, three Master students, two faculty members and six external participants (only from office environments). The experiment implied an initial start-up phase, also called pairing session, when all attendants were asked to meet and pair all devices for Bluetooth interactions, with the purpose of collecting contextual data from Android smartphones. The participants were asked to start the HYCCUPS Tracer (The HYCCUPS Tracer is an Android application designed to collect contextual data from smartphones. The application runs in background and collects multiple features pertaining to availability and mobile interaction.) each weekday between 10AM-18PM as the this time period is assumed as the interval in which most participating members are attending classes or work.

To estimate the parameters, we divide the datasets into training and test data sets. We estimate the parameters over the training data set. In Section VI we show the prediction outcomes over the test data sets.

Below we give explanation of fitting the seasonal ARIMA model with uncertainty over the MIT reality mining [36] data set (with 10 months of temporal data, of which 9 months data is used for training). We use both R and SAS package software to acquire the best model with most accurate parameter set.

In fitting the model, we have $W_t = 0.269W_{t-1} + a_t - 0.7398a_{t-10}$ and $W_t = Z_t - Z_{t-1} - Z_{t-9} + Z_{t-10}$. Thus, we

get the resulting model as:

$$W_t = \langle \alpha_1, c_1 \rangle W_{t-1} + \langle 1, c_2 \rangle a_t + \langle \alpha_2, c_2 \rangle a_{t-10} \quad (19)$$

$$W_t = Z_t - Z_{t-1} - Z_{t-9} + Z_{t-10} \quad (20)$$

Setting $(\alpha_0, \alpha_1) = (-0.259, -0.74)$ and $h = 0$, we obtain the linear interval model.

$$W_t = \langle -0.26, 0.553 \rangle W_{t-1} + \langle 1, 0 \rangle a_t + \langle -0.76, 1.64 \rangle a_{t-9} \quad (21)$$

$$W_t = Z_t - Z_{t-1} - Z_{t-9} + Z_{t-10} \quad (22)$$

With Equation 21 and 22, we present the prediction outcomes in Section VI for MIT reality mining [36] data set. In a similar manner we also produce the empirical results in case of the other two data sets.

Now with RNN model over the MIT reality mining [36] data set, out of the 10 months data, 9 months data is used for training and the remaining months data is used for testing. We consider monthly contact data, with subtraction of monthly mean and dividing it by standard deviation to remove the periodicity present in the data. As the activation function is a sigmoidal function, the data sets are scaled between 0 and 1 and the resulting data sets are used for training. To start with we consider three layers, i.e. one input, one hidden and one output layer and use MATLAB and R neural network packages. For the RNN model, number of previous output values to the first hidden layer varied from 3 to 5. The number of hidden layer neurons varied from 5 to 20 and the number of output neurons was fixed as 1. From the experiment we find that instead of a model with one hidden layer of 20 neurons, a model with four layers with two hidden layer with 10 neurons per hidden layer is better suited for training. We fix the number of iterations at 5,000, as above this the mean pattern error is fluctuating instead of converging further in the experiments. To train the RNN the inputs were fed to the input neurons and the final values are obtained after training. The output values from the trained network are then back transformed to get the predicted data for the desired time interval.

VI. EXPERIMENTAL RESULTS

In order to predict number of contacts and future contact time, we analyze the statistical features over our experimental data set. For brevity, only weekly and monthly overall contact distributions are shown for MIT reality mining [36] data set

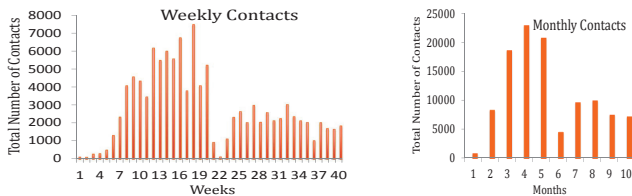


Fig. 3: Weekly and Monthly contacts: MIT Data Set
in Figure 3, for 2nd data set [37] we show daily contact distribution in first graph of Figure 4, where as 2nd graph shows the weekly contact distribution of 3rd data set in [38]. Over the MIT reality mining [36] data set, we see there is a sudden

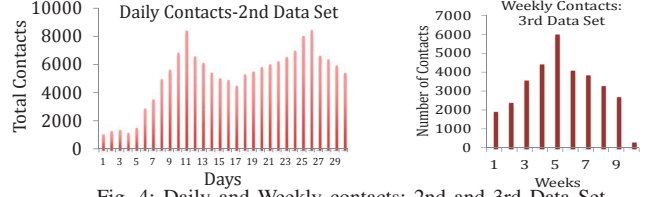


Fig. 4: Daily and Weekly contacts: 2nd and 3rd Data Set

drop of contacts over the month of September and February. Similarly, for dataset in [37] we see much less contacts over the first 6 days; where as, for data set in [38] the total number of contacts over the week 10 is lowest owing to only two remaining days.

We structure the data as groups corresponding to all users, per user and pair of users relative to hour, day, week and month. Once we have the estimated parameters over the respective organized training data set, we validate the model of its prediction accuracy over the respective test data sets. We then predict the total number of contacts, future contact time over respective time frames. Over MIT reality mining [36] data set we use 9 months data as training and the rest one month as test set; for second data set in [37] we use data over 21 days as training and predict over the remaining 9 days of test set, and over third data set [38] we train the models with 45 days of data and predict over 20 days of test set. We use the F -score to validate the test (prediction) accuracy. We also use two statistical measure i.e. R^2 values and root mean square error (RMSE) to compare the forecasts obtained from RNN with actual data values. If the observed value is O_i and predicted value $\hat{O}_i$ in i^{th} time step, then $RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (O_i - \hat{O}_i)^2}$ and $R^2 = 1 - \frac{\sum_{i=1}^n (O_i - \hat{O}_i)^2}{\sum_{i=1}^n (\hat{O}_i - 1/n \sum_{i=1}^n O_i)^2}$. RMSE gives us more information by measuring the average deviation of the estimates from the observed values. The lower the RMSE values the better fit the model is.

Figure 5, shows the accuracy of prediction for total number of contacts relative to a single participant and all participant, over hourly, daily, weekly, monthly periods for our proposed seasonal ARIMA model with uncertainty. Figure 6 shows the prediction outcome with proposed RNN model over the same data sets. From the F-score it is evident that accuracy over single participant is higher, owing to less variability or unpredictability, than that with multiple participants/users. Similarly, the accuracy decreases with higher prediction interval. More so, prediction accuracy with RNN model is higher than the SARIMA model (along with uncertainty), as it better captures the nonlinearity, dependency and the periodicity.

Similarly, we show the prediction outcome over second and third data set [37], [38] respectively, in Figures 7, 8, 9 and 10. We make similar observation as before in these cases as well.

Finally, we compare the effectiveness of our two prediction model (SARIMA (with uncertainty) and RNN) relative to the doubly recurrent model and HPP model, both our models outperforms the other two over each data set. For brevity, we predict only the total contacts per user and over all users, . In all four models the prediction accuracy is better in case of per

user evaluation, than over all users; owing to the less variability in case of single user. We show prediction outcome in terms of F-score over the three distinct data sets over distinct time intervals (for brevity, showing outcome for daily and weekly intervals only). The prediction accuracy per day is higher for all the models than its counterpart over weekly intervals. In all cases RNN superseded in prediction accuracy over all the models. More so it is also capturing the periodicity in data, with accuracy over 0.75 in all cases.

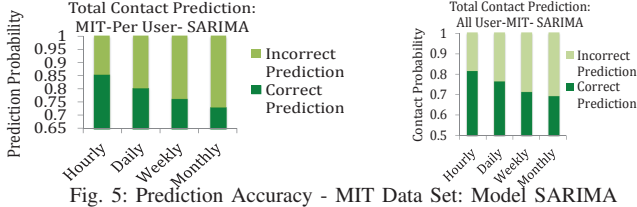


Fig. 5: Prediction Accuracy - MIT Data Set: Model SARIMA

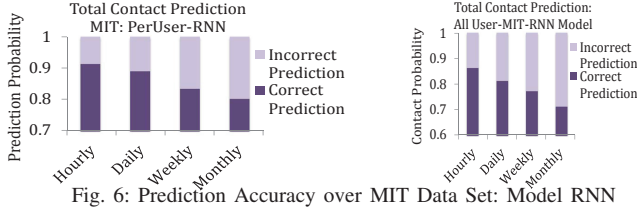


Fig. 6: Prediction Accuracy over MIT Data Set: Model RNN

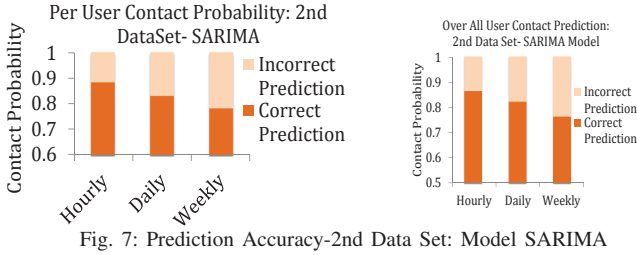


Fig. 7: Prediction Accuracy-2nd Data Set: Model SARIMA

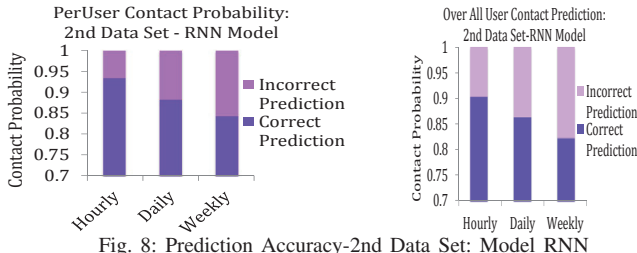


Fig. 8: Prediction Accuracy-2nd Data Set: Model RNN

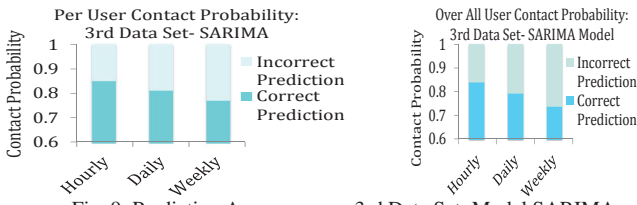


Fig. 9: Prediction Accuracy over 3rd Data Set: Model SARIMA

TABLE I: MODEL EFFECTIVENESS: RMSE AND R^2 MEASURE

Over First Data Set		
Measure	SARIMA	RNN
RMSE	163	103
R^2	0.831	0.983

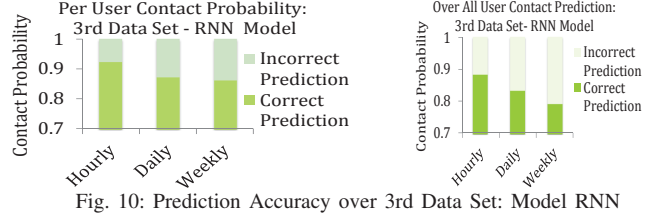


Fig. 10: Prediction Accuracy over 3rd Data Set: Model RNN

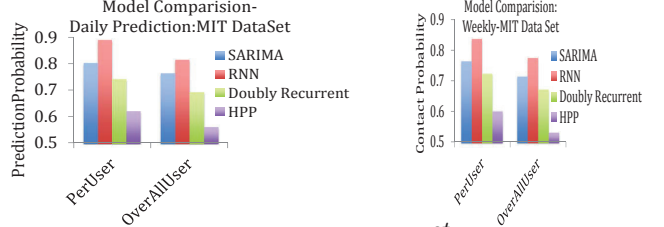


Fig. 11: Model Comparison: 1st Data Set

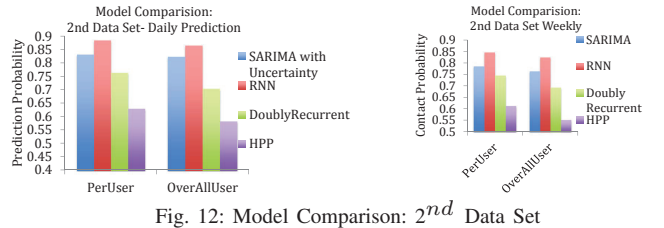


Fig. 12: Model Comparison: 2nd Data Set

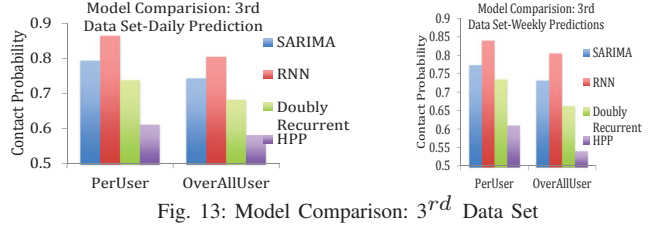


Fig. 13: Model Comparison: 3rd Data Set

TABLE II: MODEL EFFECTIVENESS: RMSE AND R^2 MEASURE

Over Second Data Set		
Measure	SARIMA	RNN
RMSE	113	63
R^2	0.4613	0.6753

TABLE III: MODEL EFFECTIVENESS: RMSE AND R^2 MEASURE

Over Third Data Set		
Measure	SARIMA	RNN
RMSE	134	84
R^2	0.6615	0.8413

In Table I, Table II, and Table III we present the RMSE and R^2 values, for the two proposed models over the three distinct data sets. The lower the RMSE value the better the prediction accuracy of the model. Thus, RNN model over the second data set has highest accuracy and the RNN model is better than the SARIMA model (with uncertainty).

VII. CONCLUSION

In an era of mobile social networks, and opportunistic networks, the prediction of future contacts is vital to the success of many applications. In this regard, we propose use of statistical models like seasonal autoregressive integrated moving average (with uncertainty) and recurrent neural network models to

evaluate the contact prediction efficacy. We observe that both these models outperforms the existing doubly recurrent model and HPP model, thus showing better ability to capture the underlying periodicity associated with human mobility and hence in contact patterns. More so, RNN is the most efficient model, owing to its ability to capture nonlinearity, dependency and hence multi-periodicity.

We propose to formulate models to evaluate prediction accuracy over contact time and contact duration, and also construct a more effective model to capture the randomness that also occurs in these type of time series data to improve the prediction accuracy further.

REFERENCES

- [1] M. Weiser, "Connecting the physical world with pervasive networks," 2002.
- [2] N. Eagle, A. S. Pentland, and D. Lazer, "Inferring friendship network structure by using mobile phone data," *Proceedings of the National Academy of Sciences*, vol. 106, no. 36, pp. 15 274–15 278, 2009.
- [3] L. Pelusi, A. Passarella, and M. Conti, "Opportunistic networking: data forwarding in disconnected mobile ad hoc networks," *IEEE Communications Magazine*, vol. 44, no. 11, pp. 134–141, 2006.
- [4] I. Chlamtac, M. Conti, and J. J.-N. Liu, "Mobile ad hoc networking: imperatives and challenges," *Ad hoc networks*, vol. 1, no. 1, pp. 13–64, 2003.
- [5] L. Aalto, N. Göthlin, J. Korhonen, and T. Ojala, "Bluetooth and wap push based location-aware mobile advertising system," in *Proceedings of the 2nd International Conference on Mobile Systems, Applications, and Services*. ACM, 2004, pp. 49–58.
- [6] C. Free, G. Phillips, L. Watson, L. Galli, L. Felix, P. Edwards, V. Patel, and A. Haines, "The effectiveness of mobile-health technologies to improve health care service delivery processes: a systematic review and meta-analysis," *PLoS Med*, vol. 10, no. 1, p. e1001363, 2013.
- [7] M. N. K. Boulos, S. Wheeler, C. Tavares, and R. Jones, "How smart-phones are changing the face of mobile and participatory healthcare: an overview, with example from ecaalyx," *Biomedical engineering online*, vol. 10, no. 1, p. 1, 2011.
- [8] H. T. Dinh, C. Lee, D. Niyato, and P. Wang, "A survey of mobile cloud computing: architecture, applications, and approaches," *Wireless Communications and Mobile Computing*, vol. 13, no. 18, pp. 1587–1611, 2013.
- [9] N. Fernando, S. W. Loke, and W. Rahayu, "Mobile cloud computing: A survey," *Future Generation Computer Systems*, vol. 29, no. 1, pp. 84–106, 2013.
- [10] A. M. Kaplan and M. Haenlein, "Users of the world, unite! the challenges and opportunities of social media," *Business horizons*, vol. 53, no. 1, pp. 59–68, 2010.
- [11] E. Miluzzo, N. D. Lane, K. Fodor, R. Peterson, H. Lu, M. Musolesi, S. B. Eisenman, X. Zheng, and A. T. Campbell, "Sensing meets mobile social networks: the design, implementation and evaluation of the cenceme application," in *Proceedings of the 6th ACM conference on Embedded Network Sensor Systems*, 2008, pp. 337–350.
- [12] S. Ioannidis, A. Chaintreau, and L. Massoulie, "Optimal and scalable distribution of content updates over a mobile social network," in *IEEE INFOCOM*, 2009, pp. 1422–1430.
- [13] W. Gao, Q. Li, B. Zhao, and G. Cao, "Multicasting in delay tolerant networks: a social network perspective," in *Proceedings of the tenth ACM international symposium on Mobile ad hoc networking and computing*, 2009, pp. 299–308.
- [14] M. Conti and M. Kumar, "Opportunities in opportunistic computing," *Computer*, vol. 43, no. 1, pp. 42–50, 2010.
- [15] J. Zhao and G. Cao, "Vadd: Vehicle-assisted data delivery in vehicular ad hoc networks," *IEEE Transactions on Vehicular Technology*, vol. 57, no. 3, pp. 1910–1922, 2008.
- [16] A. Howard, M. J. Matarić, and G. S. Sukhatme, "Mobile sensor network deployment using potential fields: A distributed, scalable solution to the area coverage problem," in *Distributed Autonomous Robotic Systems 5*. Springer, 2002, pp. 299–308.
- [17] H. Hartenstein and K. P. Laberteaux, "A tutorial survey on vehicular ad hoc networks," *IEEE Communications Magazine*, vol. 46, no. 6, pp. 164–171, 2008.
- [18] M. C. Gonzalez, C. A. Hidalgo, and A.-L. Barabasi, "Understanding individual human mobility patterns," *Nature*, vol. 453, no. 7196, pp. 779–782, 2008.
- [19] E. Cho, S. A. Myers, and J. Leskovec, "Friendship and mobility: user movement in location-based social networks," in *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2011, pp. 1082–1090.
- [20] K. Lee, S. Hong, S. J. Kim, I. Rhee, and S. Chong, "Slaw: A new mobility model for human walks," in *Proc. IEEE INFOCOM*, 2009, pp. 855–863.
- [21] R. I. Ciobanu and C. Dobre, "Predicting encounters in opportunistic networks," in *Proceedings of the 1st ACM workshop on High performance mobile opportunistic systems*, 2012, pp. 9–14.
- [22] R. D. Malmgren, J. M. Hofman, L. A. Amaral, and D. J. Watts, "Characterizing individual communication patterns," in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2009, pp. 607–616.
- [23] L. Yang, H. Jiang, S. Wang, L. Wang, and Y. Fang, "Characterizing pairwise contact patterns in human contact networks," *Ad Hoc Networks*, vol. 10, no. 3, pp. 524–535, 2012.
- [24] A. Pentland, N. Eagle, and D. Lazer, "Inferring social network structure using mobile phone data," *Proceedings of the National Academy of Sciences (PNAS)*, vol. 106, no. 36, pp. 15 274–15 278, 2009.
- [25] M. McNett and G. M. Voelker, "Access and mobility of wireless pda users," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 9, no. 2, pp. 40–55, 2005.
- [26] G. E. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.
- [27] G. Zhang, B. E. Patuwo, and M. Y. Hu, "Forecasting with artificial neural networks:: The state of the art," *International Journal of Forecasting*, vol. 14, no. 1, pp. 35–62, 1998.
- [28] M. Khashei, M. Bijari, and G. A. R. Ardali, "Improvement of autoregressive integrated moving average models using fuzzy logic and artificial neural networks (anns)," *Neurocomputing*, vol. 72, no. 4, pp. 956–967, 2009.
- [29] M. Khashei and M. Bijari, "An artificial neural network (p, d, q) model for timeseries forecasting," *Expert Systems with applications*, vol. 37, no. 1, pp. 479–489, 2010.
- [30] C. Xu, Z. Li, and W. Wang, "Short-term traffic flow prediction using a methodology based on autoregressive integrated moving average and genetic programming," *Transport*, vol. 31, no. 3, pp. 343–358, 2016.
- [31] J. Park, D. Li, Y. L. Murphey, J. Kristinsson, R. McGee, M. Kuang, and T. Phillips, "Real time vehicle speed prediction using a neural network traffic model," in *The International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2011, pp. 2991–2996.
- [32] D. Dubois and H. Prade, "Operations on fuzzy numbers," *International Journal of systems science*, vol. 9, no. 6, pp. 613–626, 1978.
- [33] L. B. Almeida, "Backpropagation in perceptrons with feedback," in *Neural computers*. Springer, 1989, pp. 199–208.
- [34] A. Carling, *Introducing neural networks*. Sigma Press Wilmslow, 1992.
- [35] R. Kothari and K. Agyepong, "Induced specialization of context units for temporal pattern recognition and reproduction," in *Proceedings of the Neural Networks for Signal Processing Workshop*. IEEE, 1997, pp. 131–140.
- [36] N. Eagle and A. S. Pentland, "CRAWDAD dataset mit/reality (v. 2005-07-01)," Downloaded from <http://crawdad.org/mit/reality/20050701>, Jul. 2005.
- [37] M. Piorkowski, N. Sarafjanovic-Djukic, and M. Grossglauser, "CRAW-DAD dataset epfl/mobility (v. 2009-02-24)," Downloaded from <http://crawdad.org/epfl/mobility/20090224>, Feb. 2009.
- [38] R. I. Ciobanu and C. Dobre, "CRAWDAD dataset upb/hyccups (v. 2016-10-17)," Downloaded from <http://crawdad.org/upb/hyccups/20161017>, Oct. 2016.