Artificial Intelligence:
Concepts, Methodologies, Tools, and Applications

Information Resources Management Association
USA

Volume III

Chapter 67
Opportunistic Neighbour Prediction Using an Artificial Neural Network

Fraser Cadger
Intelligent Systems Research Centre, University of Ulster, UK

Jose Santos
Intelligent Systems Research Centre, University of Ulster, , UK

Kevin Curran
Intelligent Systems Research Centre, University of Ulster, UK

Sandra Moffet
Intelligent Systems Research Centre, University of Ulster, UK

ABSTRACT

Device mobility is an issue that affects both MANETs and opportunistic networks. While the former employs conventional routing techniques with some element of mobility management, opportunistic networking protocols often use mobility as a means of delivering messages in intermittently connected networks. If nodes are able to determine the future locations of other nodes with reasonable accuracy then they could plan ahead and take into account and even benefit from such mobility. Location prediction in combination with geographic routing has been explored in previous literature. Most of these location prediction schemes have made simplistic assumptions about mobility. However more advanced location prediction schemes using machine learning techniques have been used for wireless infrastructure networks. These approaches rely on the use of infrastructure and are therefore unsuitable for use in opportunistic networks or MANETs. To solve the problem of accurately predicting future location in non-infrastructure networks, the authors have investigated the prediction of continuous numerical coordinates using artificial neural networks. Simulation using three different mobility models representing human mobility has shown an average prediction error of less than 1m in normal circumstances.

1. INTRODUCTION

Mobile ad-hoc network (MANET) mobility can lead to dynamic behaviour and problems such as sub-optimal routing or link breaks caused by nodes changing their position. Potentially intermittent connectivity poses a significant challenge in performing end-to-end routing. Location prediction for mobility

management has been studied in geographic routing protocols such as Chen et al. (2006), Chou et al. (2008), and Son et al. (2004), and has also been used for QoS; Shah and Nahrstedt (2002), Stojmenovic et al. (2000), and Cadger et al. (2011). Geographic routing is localized and does not build end-to-end paths instead performing forwarding on a hop-by-hop basis. It does however still rely on continuous connectivity to an extent; if a node is unable to find a suitable next hop using the forwarding condition then the packet will be dropped. Although this is common in conventional infrastructure as well as ad-hoc routing, it is not always desirable. Opportunistic networking is a subfield of Delay Tolerant Networking (DTN). DTNs acknowledge the possibility that connections between network nodes may be intermittent; both in terms of end-to-end connection (which might never be possible using conventional routing) or even nodes being completely disconnected from all other devices for large periods of time. Existing ad-hoc routing protocols generally rely on continuous connectivity for building paths, and are unable to cope with prolonged periods of isolation or intermittent connectivity.

Opportunistic networking is an attempt to provide communications in DTNs and other scenarios with intermittent and potentially unpredictable connectivity. Opportunistic networks make forwarding decision on a hop-by-hop basis using local information. Where opportunistic networking differs from geographic routing is its use of the store-and-forward paradigm; if a node does not have any suitable neighbours or is completely disconnected from the network then it will store the message until it finds a suitable next hop. Mobility is therefore a significant factor in most opportunistic networking protocols, being both the cause of and solution to intermittent connectivity. If devices are able to accurately predict where either they or their current neighbours would be at a particular time then they could make forwarding decisions with the aim of increasing reception potential or saving energy. In Li and Shatz (2008) a geographic routing protocol with some similarities to opportunistic networking allows nodes to favour neighbours who are moving towards the destination over static nodes.

Wireless infrastructure networks such as WLANs and cellular networks have employed techniques from the field of machine learning such as artificial neural networks (NN) (Capka and Boutaba, 2004) and Hidden Markov Models (Prasad and Agrawal, 2010) for location prediction, which boast high accuracy and success. However, all of these works are reliant on either WLAN or cellular infrastructure as they formulate location prediction as a discrete classification problem in which the aim is to classify a user's location in terms of proximity to a wireless access point or cell and which makes them unsuitable for use in ad-hoc or opportunistic networks. An approach which is able to predict locations in a manner which is not dependent on the existence of infrastructure and requires knowledge of the area is therefore highly desirable. This led to the consideration of continuous regression techniques instead of discrete classification for location in prediction in MANETs. In Cadger et al. (2012) three location prediction algorithms were used to predict future device locations using MANET mobility traces with a NN performing best. The difference between Cadger et al. (2012) and previous works in infrastructure networks such as Capka and Boutaba (2004) and Prasad and Agrawal (2010) is that Cadger et al. is able to take previous coordinates and use these to predict the future coordinates for that device, without the need for any infrastructure or area-specific knowledge which makes it suitable for opportunistic networks.

Although the NN algorithm used in Cadger et al. (2012) performed well, these tests were performed in Matlab using mobility traces obtained from ns-2 simulations. Instead of receiving coordinates from neighbouring nodes and predicting their future location, the algorithm was simply provided with a complete series of previous locations. Similarly, the experiment did not actually simulate a network itself. Therefore, although the NN algorithm achieved a high level of prediction accuracy (average Mean Square Error (MSE) of 0.102) it was not evaluated under network conditions. This paper presents

an implementation of a NN for predicting neighbour locations inside a geographic routing protocol in ns-2. Simulation of mobile networks of varying sizes using three different mobility scenarios of human movement has shown that a NN trained using data from only one of these mobility models is able to predict neighbouring device coordinates often obtaining an average of less than 1m error. The differences between the work described in this paper and Cadger et al. (2012) are: in this paper the NN algorithm is running within a routing protocol, predictions are made based on location updates sent by neighbouring nodes (and are thus susceptible to missing/out of date information), and actual network/ mobility behaviour is simulated. While this experiment (and Cadger et al. 2012) uses Cartesian coordinates, the general approach could be extended to GPS or any other form of numerical coordinates. Thus the approach described in this paper is suitable for use with either conventional location systems such as GPS or Galileo or localization based on alternative means, and can therefore be used in both indoor and outdoor environments depending on the location system.

2. PREVIOUS WORK

Research into opportunistic networking and MANETs have common roots; both are concerned with networks in which there is limited or no infrastructure and mobility is a factor. Where opportunistic networks and MANETs differ is in their approach to routing. Although MANET protocols attempt to offer solutions to the lack of infrastructure and effects of mobility, they still largely rely on the traditional networking concept of end-to-end communications in which a packet neatly proceeds along a link from source to destination in a timely manner. A few approaches such as the aforementioned geographic routing do not have any concept of a path at all and simply perform forwarding on a hop-by-hop basis. However, such approaches still have the common limitation of needing to make a decision immediately; once a packet is under consideration the protocol will attempt to find a suitable next-hop, and if it is unable to do so they packet will be dropped altogether. While some QoS-intensive applications rely on maintaining a low level of delay, other applications can be considered delay-tolerant and in these instances receiving a packet at *some point* is more important than receiving it within a particular time frame.

These applications therefore require an alternative paradigm in which packets that cannot be forwarded immediately do not necessarily have to be dropped. Opportunistic routing, therefore aims to approach the issue of routing in infrastructure-less mobile networks from the perspective of using mobility and intermittent contact to forward packets in instances where delay is permissible (DTNs). The term opportunistic network (or oppnet) was first introduced in Lilien and Gupta et al. (2006) and discussed in Lilien and Kamal et al. (2006) originally in the context of disaster recovery networks. However, the idea of opportunistic networking can be characterised as being part of the wider area of pervasive/ubiquitous computing and are now being explored outside of the area of disaster recovery. A survey of various approaches to opportunistic networking is provided in Nguyen and Giordano (2009) who identify three main categories of opportunistic routing protocols; context-oblivious, mobility-based, and social context-based. Context-oblivious protocols are described as being dependant on flooding techniques and includes protocols that utilise both blind (network-wide) and limited flooding (flooding limited based on some criteria) (Nguyen and Giordano, 2009). The latter category includes protocols such as epidemic routing (Vahdat and Becker, 2000) as well as Spray and Wait (Spyropoulos et al., 2005) and network coding schemes (Wang et al., 2005). Mobility-based protocols, use mobility patterns and information to make decisions on whether to forward or store a message. Opportunistic protocols that can be classified as

mobility-based include PRoPHET (Lindgren et al., 2003) which calculates a device's delivery probability based on its previous contact with other nodes and only forwards packets to nodes with a higher delivery probability and CAR (Musolesi and Mascolo, 2006) which uses Kalman filters to predict context information for message placement. Finally, social context-based protocols attempt to extend mobility-based protocols by taking into account the social context responsible for mobility and contact between devices (Nguyen and Giordano, 2009). Examples of social context-based protocols include HiBOP (Boldrini et al., 2007) and Propicman/SpatioTemp (Nguyen et al., 2007; Nguyen and Giordano, 2008).

Research on location prediction in wireless networks can be split into two broad categories; research involving MANETs, and research involving wireless infrastructure networks. The majority of location prediction research in MANETs has involved some form of geographic (or location-aware routing). The earliest work on combining geographic routing with some form of location prediction was performed by Chen et al. (2006) who devised a connection time metric based on node mobility. A similar metric is that of the motion stability metric proposed in Shah and Nahrstedt (2002) which is defined as the degree of mobility variation by a neighbour, with the authors believing a large degree of motion instability is a factor in high levels of jitter. Unlike Stojmenovic et al. (2000), the method devised in Shah and Nahrstedt (2002) actually predicts coordinates through the use of a simple prediction scheme based on two previous coordinates. A similar approach to predicting coordinates is used in Stojmenovic et al. (2000). However Stojmenovic et al. claims that it does not make assumptions about direction (whereas Shah and Nahrstedt (2002) assumes linearity) and works regardless of direction of movement. Despite the success of the various MANET location prediction schemes most of them have been somewhat basic. This was observed in Cadger et al. (2011) where it was speculated that more advanced location prediction schemes (which in turn would yield more accurate results) would lead to further improvements in routing performance. More accurate location prediction schemes are also desirable as they will allow nodes to have greater awareness of the state (and future state) of both themselves and their neighbouring nodes which is particularly important for opportunistic networks.

In the other area of location prediction, wireless infrastructure networks have sought the use of more advanced location predictions schemes in the form of machine learning algorithms such as artificial neural networks (Capka and Boutaba, 2004) and Hidden Markov Models (Prasad ad Agrawal, 2010) have. However these approaches are unsuitable for use without infrastructure as they formulate the problem of location prediction in terms of predicting location as proximity to and usage to network infrastructure. For instance, Prasad and Agrawal (2010) trains a HMM algorithm using traces obtained from real WLANs which show the sequence of APs nodes connect to. Once trained, the algorithm can then predict the next AP a node will connect to based on its previous connections. Both Capka and Boutaba (2004) and Prasad and Agrawal (2010) are examples of classification algorithms. It is however still possible to use classification approaches for non-infrastructure networks so long as a discrete list of locations is available. However such approaches would be inflexible and imprecise.

As classification-based methods do not appear particularly well-suited to opportunistic network location prediction it is perhaps worth considering the possibility of using regression to predict future device locations. The use of regression-based machine learning algorithms to predict neighbour locations in MANETs was initially proposed in Cadger et al. (2012). In Cadger et al., the authors suggested that instead of performing discrete classification, previous (non-machine learning) methods of location prediction in which continuous coordinates are the output should be modified to use machine learning algorithms. This approach was implemented in the form of mobility traces from six ns-2 scenarios of varying mobility which were used as the training and testing data to evaluate three machine learning

algorithms (decision trees, NN and support vector regression) in Matlab. The results indicated that the neural network was the overall best performer in terms of accuracy.

This paper follows on from the work of Cadger et al. (2012) by implementing a NN algorithm inside a geographic routing protocol in ns-2. Unlike Cadger et al. in which all testing was done using inputs obtained from previous simulations, the experiments detailed in this paper document a NN algorithm performing predictions from inside a routing protocol during mobile network simulations. Although the work described in this paper still relies on the use of simulation and mobility models, it is important to acknowledge that designing and implementing a machine learning algorithm for location prediction on mobile devices is a time consuming endeavour. Thus, these experiments should serve as a reasonable indication of whether the NN algorithm is able to perform in real mobility scenarios and based on this, whether or not to proceed with a real-world implementation of the NN algorithm.

3. PREDICTING NEIGHBOUR LOCATIONS USING NEURAL NETWORKS

This paper is primarily motivated by two factors; to determine whether NNs can produce results similar to Cadger et al. (2012) when applied to a geographic routing protocol in a mobile network scenario, and to improve the accuracy of location prediction schemes for use with geographic routing protocols in such scenarios. A potential weakness of the NN algorithm is its 'black box' nature, which means that it is difficult for humans to observe exactly how a NN operates. The NN is fed a series of inputs and then outputs the predicted values without a clear means for the human to determine how the output was obtained. This makes performance analysis somewhat difference, as although the results can be observed it is not obvious how these results were achieved. This makes the design, configuration, and adjustment of NNs somewhat difficult and there is no universal optimum configuration. The design of NN typically varies depending on the task and data, as well as the computational restrictions.

The main adjustable parameters in NN are the number of layers, the number of neurons, and the algorithm used to train the network. All NNs contain at least two layers; the input and output layers. Most NNs also contain at least one hidden layer. The number of hidden layers is a matter of much debate with some believing that more layers increases prediction accuracy, while others arguing that one layer is sufficient for most applications. There is no consensus as to optimum number of layers, and often the decision depends on the processing power available as more layers increases computational cost. Similarly, the number of hidden neurons is also subject to some debate. One belief (Sarle, 2012) is that number of hidden neurons can only be determined through experimentation with different configurations. If the number is too low then there is a risk of predictions being inaccurate and the NN unable to generalise, while too many can still lead to poor generalisation as a result of overfitting and high variance (Sarle, 2012). Unlike the number of hidden neurons, the number of input layers is typically determined by the number of variables to be used as inputs, while the number of output neurons is determined by the number of desired outputs. In Cadger et al. (2012) seven inputs were used to represent the current time, two most recent coordinates and timestamp, and the previous coordinates and timestamp, while there were two outputs (the predicted x and y coordinates). The choice of training algorithm on the other hand, is often limited by the availability of algorithms for the particular software used to implement the NN, although hardware resources are still a factor. The NN configuration used in Cadger et al. (2012) is as follows; one hidden layer, 10 hidden neurons (sigmoid) and the training algorithm was Levenburg-Marquadt.

4. NEURAL NETWORK DESIGN AND IMPLEMENTATION

The NN algorithm was designed and tested using FANN (Li and Shatz, 2008) which is an open-source C library for creating multilayer perceptron NNs. FANN was chosen for its lightweight architecture and desirable performance levels, as well as the fact it is written in C and provides a C++ wrapper as ns-2 routing protocols are written in C. Following experimentation with various architectures, the following architecture was chosen; 7 input neurons, 15 hidden neurons, 1 hidden layer, and 2 output neurons the hidden transfer function was sigmoid-symmetric while the output transfer function was linear. The iRPROPR (Igel and Husken, 2000) based on the RPROP algorithm (Riedmiller and Braun, 1993) was used for training. After testing, the NN algorithm was implemented in a routing protocol in ns-2. The protocol used was the Greedy Perimeter Stateless Routing algorithm (GPSR) (Karp and Kung, 2000) using code from Kiess (2012). GPSR is a hybrid geographic routing algorithm which combines greedy and face routing. However, for the purpose of this work the NN implementation was simulated in greedy-only mode. GPSR like most other geographic routing protocols uses the beaconing system for discovering (and keeping alive) links with other devices and disseminating location information. Prior to implementation of the NN algorithm GPSR was modified as in Cadger et al. (2011) to store the two most recent (instead of just the current) coordinates and their timestamps. Location prediction is then performed when GPSR is determining which of its neighbours is closest to the destination; instead of using the coordinates from the last update, the NN implementation passes the two previous locations and their time to the NN prediction method which inputs these parameters along with the current time into the NN and then receives the predicted x and y coordinates as outputs.

5. EXPERIMENT CONFIGURATION

Testing was performed by running simulations of the modified GPSR in ns-2 to determine prediction accuracy. A total of 27 simulations were run using three different mobility models and three different numbers of nodes as well as varying the beacon interval. The mobility models used were Random Waypoint Model (RWP) (Johnson and Maltz, 1996), Reference Point Group Mobility (RPGM) (Hong et al., 1999) and an implementation of the Gauss-Markov model (GM) (Liang and Haas, 1999) based on Camp et al. (2002). The reasons for selecting these three mobility models were as follows; RWM and RPGM were the mobility models used in Li and Shatz (2008) so in order to compare the results from the ns-2 simulation of the NN algorithm with the results from Cadger et al. (2012) it was necessary to use these two mobility models. RWM is generally considered a simplistic (and possibly inaccurate) representation of mobility as it is purely random and does not take into account either history or surrounding nodes. On the other hand, RPGM is based on group mobility (but where individual mobility is also permissible) and is therefore considered more realistic. In addition GM was chosen because it allows varying degrees of randomness and memory-based decisions depending on the value of the alpha parameter (Liang and Haas, 1999). Both RPGM and GM trace files for ns-2 were generated using the BonnMotion tool (Aschenbruck et al., 2010). While RWP traces were generated using the setdest tool that comes with ns-2. For RPGM the probability of a node joining a group was set to 0.75, while for GM the update frequency was set to 1, the angle standard deviation 0.5, and the speed standard deviation 0.5 (the use of 0.5 for both parameters means that the model is balanced in the middle between memoryless random and memory-based mobility).

Traces for all mobility models were produced for 10, 50, and 100 node scenarios and all scenarios ran for 600 seconds on an area of size 1500mx400m (with the exception of the GM which ran on 1400mx300m due to a known bug in which traces are produced containing destinations outside of the specified area causing the simulation to crash). Similarly, all scenarios had a maximum velocity of 2.5m/s and a maximum pause time of 20s. The maximum velocity of 2.5m/s is the same as the one used in Cadger et al. (2012) where it was decided to use this value as it was deemed the most appropriate maximum speed for ordinary human movement. The traffic pattern used for the 50 and 100 node scenarios was as follows; constant bit rate (CBR) UDP traffic containing 30 streams with a packet send rate of 0.5. For the 10 node scenario contained 6 streams of CBR traffic with a send rate of 0.5. While it is important to recognize that opportunistic networks might not feature constant streams of traffic, the lack of established opportunistic networking applications makes the modelling of traffic somewhat difficult. Therefore the decision to use CBR traffic was taken as it would allow for the possibility of congestion (at some nodes) and the resulting loss of location updates.

An important difference in the experiments detailed in this paper and those described in Cadger et al. (2012) is the variation of beacon intervals. Where Cadger et al. uses a beacon interval of 0.5s for all scenarios, our experiments use three different beacon periods; 0.5s, 5s and 50s. The purpose of this is to determine how the NN algorithm is able to react when faced with less frequent updates (particularly in the case of the 50s where there is a large duration between updates), as this is a potential issue for real-world opportunistic networks and MANETs where beacons may be delayed either unintentionally (as a result of buffering) or intentionally (to reduce traffic in a congested network or save energy). Prediction accuracy was estimated as follows; a comparison of this prediction with the neighbour's actual coordinates was made using the God utility found in ns-2 which allows routing protocols to directly access the simulator and retrieve parameters.

6. RESULTS

6.1. Prediction Accuracy

Tables 1, 2, and 3 show the results of the simulations for each of the three mobility models. Note, that here error is not MSE and is the actual difference in meters between the predicted and actual coordinates. When evaluating these results, it is important to note that there is no obvious 'state of the art' to compare them with. As Cadger et al. (2012) was the first approach to propose the use of machine learning algorithms for predicting continuous coordinates, while there is no data on the accuracy of geographic routing location prediction algorithms. Therefore, establishing a threshold for prediction accuracy is somewhat difficult.

Table 1. Absolute error for all RWM scenarios

Number of Nodes	Error (0.5 Seconds)	Error (5 Seconds)	Error (50 Seconds)
10	0.123	0.53	7.55
50	0.1025	0.484	4.24
100	0.148	0.835	2.45

Table 2. Absolute error for all RPGM scenarios

Number of Nodes	Error (0.5 Seconds)	Error (5 Seconds)	Error (50 Seconds)
10	0.155	0.475	1.58
50	0.206	1.08	11.82
100	0.1385	0.727	0.145

Table 3. Absolute error for all GM scenarios

Number of Nodes	Error (0.5 Seconds)	Error (5 Seconds)	Error (50 Seconds)
10	0.182	0.775	3.345
50	0.135	0.397	1.59
100	0.188	0.894	3.97

Results show that the NN algorithm is able to predict future coordinates very accurately and with minimal error except for the 50 second beaconing scenarios. Although there is no commonly accepted threshold for location prediction error, many systems in the area of localization strive for an error of less than 2-3m therefore anything below this should be considered highly desirable and anything below 1m exceptional. In a total of 18 scenarios out of 27 an absolute error of less than 1m was observed. Furthermore an error of >3m was only observed in 5 scenarios and all of these instances occurred when beacon duration was set as 50 seconds. For the 0.5s and 5s scenarios only one of these 18 scenarios contained an error greater than 1m (1.08m recorded in RPGM 50 nodes, 5s) and none greater than 2m. This indicates a high degree of accuracy in predicting future locations when provided with frequent (0.5s or 5s) location updates, although performance does appear to decrease when faced with infrequent (50s updates).

With regards to the 50s beaconing, it is not surprising that the results found in these scenarios contain the highest level of error (11.82m is the highest error of any scenario) given the large duration between updates. At a maximum velocity of 2.5m/s it is possible that a person can move up to 125m from their previously recorded location. Similarly, when the two previous updates are received 50s apart it is also more difficult to develop a pattern in these updates than if they are received more regularly as it is possible that a person's mobility will have changed significantly between previous updates. However, the 50s scenarios are included as exceptional scenarios to demonstrate how the NN algorithm performs when faced with conditions not well-suited to location prediction. For the 50s scenarios only 4 out of 9 scenarios contained an error greater than 2m, however large errors such as 11.82 and 7.55 indicate that the algorithm can struggle when faced with irregular updates. It is also possible that the poorer results are also due to the training set used and that a training set based on less frequent updates would perform better in such a scenario. In general though, the NN algorithm performs reasonably well when faced with infrequent updates.

Performance with regards to the different mobility models seems to be fairly balanced. Although the NN algorithm was trained using only the 50 node RPGM dataset from Cadger et al. (2012) which itself used beacon periods of 0.5s, it is still able to accurately predict (often with <1m of error) future device locations in scenarios which use different mobility models, number of nodes, and beacon periods. For instance, the lowest prediction error (0.1025m) is found in the 50 node RWM scenario with a beacon interval of 0.5s, while the highest prediction error (11.82m) is from the 100 node RPGM scenario with a beacon period of 50s. Considering number of nodes there again appears to be no obvious 'perfect' scenario with the highest and second highest level of error occurring at 50 node and 10 node scenarios respectively.

6.2. Comparison with Other Results

As previously outlined the only paper to discuss the use of NNs for predicting coordinates is Cadger et al. (2012). Tables 4 and 5 show a comparison between the errors in MSE between the results obtained from these experiments and the previous results obtained in Cadger et al. It is important to note that only the results from the 0.5 beacon interval are considered as Cadger et al. only experimented with a beacon interval of 0.5s, and that the results from the GM simulations are excluded as Cadger et al. did not use the GM model.

The purpose of this paper is to ascertain whether the approach (but not necessarily the particular implementation) proposed in Cadger et al. (2012) is viable for use in an actual opportunistic networking routing protocol. Before performing these experiments it was expected that the NN algorithm deployed inside the GPSR routing protocol would not achieve as accurate a performance as the Matlab analysis in Cadger et al. This was because the Matlab analysis was somewhat idealized and was not being performed 'live' where changes were actually occurring. Another aspect in favour of Cadger et al. was that there was no chance of missing or delayed update packets which is always a possibility in simulated and real deployments. Furthermore, the Matlab analysis used training and testing data derived from the same overall dataset. Even more importantly, testing was performed using a training set from the same scenario (for instance, training and testing for the 10 node RWM scenario was performed using training and testing sets from 10 node RWM scenarios) whereas in our simulations only the 50 node RPGM training set was used for all scenarios even when the number of nodes and/or mobility model differed. Therefore it initially appears surprising that the results from our simulations are in all scenarios better than the results from the Matlab analysis of Cadger et al. (2012). However, it is important to remember that in Cadger et al., the configuration for the NN is based on the default configuration in Matlab. On the other hand, the NN algorithm discussed in this paper was developed by training and testing different architectures and configurations using the FANN application with the best performing configuration being chosen. Although this seems like the logical explanation for this NN evaluation outperforming Cadger et al. (2012), it is important to remember the purpose of these experiments was not to devise a NN implementation that outperformed Cadger et al. but instead to prove that such an NN implementation was viable for use inside an opportunistic networking protocol. The fact that the specific configuration used in this paper outperformed Cadger et al. is further proof that the general concept of using a NN algorithm for predicting continuous coordinates is a viable and desirable approach for performing location prediction in opportunistic networks or MANETs. Overall, all these results indicate that a 2-layer

Table 4. Comparison of MSE between results from Cadger et al. (2012) and our simulations for RWM scenarios

Number of Nodes	MSE from [4]	MSE from ns-2 Simulation
10	73	0.054
50	3.31	0.0475
100	0.172	0.076

Table 5. Comparison of MSE between results from Cadger et al. (2012) and our simulations for RPGM scenarios

Number of nodes	MSE from [4]	MSE from ns-2 simulation
10	0.357	0.143
50	0.539	0.1463
100	0.5049	0.145

NN algorithm with 15 hidden neurons is able to accurately predict future device locations in various scenarios of human mobility where number of nodes, mobility model, and beacon interval are all varied. That the same NN architecture trained on a dataset from one of these scenarios (50 nodes RPGM with 0.5s beacon interval) was able to perform such accurate predictions in a wide variety of different scenarios confirms that the NN prediction approach proposed in Cadger et al. (2012) is suitable for use in mobile network location prediction.

7. CONCLUSION

The primary contribution of this paper is to demonstrate the suitability of an artificial neural network implemented inside an ad-hoc routing protocol for predicting human mobility in opportunistic network scenarios. This work builds on that of Cadger et al. (2012) where three machine learning algorithms were evaluated for their ability to accurately predict future device locations based on mobility traces obtained from mobile network simulations. In Cadger et al. the neural network was overall the best performer in terms of prediction accuracy, and this work therefore focuses on further evaluating its potential for use in mobile network location prediction. This paper illustrates the process of implementing a two-layer feedforward NN algorithm created with the FANN library inside the GPSR geographic routing protocol in ns-2. Results showing the accuracy of the NN algorithm's predictions in actual simulations are then presented. The average prediction error from all scenarios is only 1.88m. However it is important to recognize that this figure contains the errors from the 50s scenario which are significantly higher than the other scenarios' errors, and if the results from the 50s scenarios are excluded the average is less than 1m of error. Similarly, a comparison of the results from the RPGM and RWM simulations with the corresponding results from Cadger et al. (2012) showed that the NN algorithm simulated in ns-2 actually performed better than the NN algorithm analyzed in Matlab.

Even when the duration, number of nodes, and mobility model was changed the NN algorithm was still able to predict neighbors' locations with a very high degree of accuracy. The paper also discussed possible future applications of the results most of which were related to developing an opportunistic networking routing protocol based on NN location prediction. It is the intention of the authors to pursue the development of such a protocol and also to implement the NN prediction algorithm on Android smartphones in order to assess its performance on realistic mobile hardware. Overall, these results are very promising in particular the consistently high prediction accuracy of the NN algorithm in a variety of differing mobile scenarios. These results are applicable to a wide variety of areas where mobility and location are important factors.

REFERENCES

Aschenbruck, N., Ernst, R., Gerhards-Padilla, E., & Schwamborn, M. (2010). BonnMotion: a mobility scenario generation and analysis tool. *Proceedings of the 3rd International ICST Conference on Simulation Tools and Techniques* (pp. 51:1-51:10). doi:10.4108/ICST.SIMUTOOLS2010.8684

Boldrini, C., Conti, M., Jacopini, J., & Passarella, A. (2007). HiBOp: a History Based Routing Protocol for Opportunistic Networks, In World of Wireless, Mobile and Multimedia Networks. Proceedings of WoWMoM 2007 IEEE International Symposium (pp. 1-12). doi:10.1109/WOWMOM.2007.4351716

Cadger, F., Curran, K., Santos, J., & Moffett, S. (2012). MANET Location Prediction Using Machine Learning Algorithms. In Y. Koucheryavy, L. Mamatas, I. Matta, & V. Tsaoussidis (Eds.), *Wired/Wireless Internet Communication* (pp. 174–185). Springer Berlin Heidelberg. doi:10.1007/978-3-642-30630-3_15

Cadger, F., Curran, K., Santos, J., & Moffett, S. (2011). An Analysis of the Effects of Intelligent Location Prediction Algorithms on Greedy Geographic Routing in Mobile Ad-Hoc Networks. *Proceedings of the 22nd Irish Conference on Artificial Intelligence and Cognitive Science*, Derry.

Camp, T., Boleng, J., & Davies, V. (2002). A survey of mobility models for ad hoc network research. *Wireless Communications and Mobile Computing*, 2(5), 483–502. doi:10.1002/wcm.72

Capka, J., & Boutaba, R. (2004). Mobility prediction in wireless networks using neural networks," Management of Multimedia Networks and Services, pp. 320–333.

Chen, Q., Kanhere, S., Hassan, M., & Lan K.-C. (2006, June). Adaptive Position Update in Geographic Routing. Proceedings of the *2006 IEEE International Conference on Communications* (pp. 4046-4051).

Chou, C., Ssu, K., & Jiau, H. (2008, February). Dynamic route maintenance for geographic forwarding in mobile ad hoc networks. *Computer Networks*, 52(2), 418–431. doi:10.1016/j.comnet.2007.10.001

Hong, X., Gerla, M., Pei, G., & Chiang, C.-C. (1999). A group mobility model for ad hoc wireless networks. *Proceedings of the 2nd ACM international workshop on Modeling, analysis and simulation of wireless and mobile systems* (pp. 53-60). doi:10.1145/313237.313248

Igel, C., & Husken, M. (2000). Improving the RPROP Learning Algorithm. *Proceedings of the Second International Symposium on Neural Computation (NCC 2000)*.

Johnson, D. B., & Maltz, D. A. (1996). Dynamic source routing in ad hoc wireless networks. Mobile Computing (pp. 153-181). doi:10.1007/978-0-585-29603-6_5

Karp, B., & Kung, H. (2000). GPSR: greedy perimeter stateless routing for wireless networks. *Proceedings of the 6th annual international conference on Mobile computing and networking* (pp. 243–254). doi:10.1145/345910.345953

Kiess, W. (2012, March 29). *HLS patch for ns-2.29 and ns-2.33*. Retrieved from http://www.cn.uni-duesseldorf.de/alumni/kiess/software/hls-ns2-patch

Li, J., & Shatz, S. M. (2008). Toward using node mobility to enhance Greedy-forwarding in geographic routing for mobile ad hoc networks. Proceedings of *The International Workshop on Mobile Device and Urban Sensing (MODUS 2008)*, *St. Louis, MO* (pp. 1-8).

Liang, B., & Haas, Z. J. (1999). Predictive distance-based mobility management for PCS networks. Proceedings of *INFOCOM '99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies* (Vol. 3, pp. 1377 -1384). doi:10.1109/INFCOM.1999.752157

Lilien, L., Gupta, A., Member, S., & Yang, Z. (2006). Opportunistic Networks for Emergency Preparedness and Response.

Lilien, L., Kamal, Z. H., Bhuse, V., & Gupta, A. (2006). Opportunistic networks: the concept and research challenges in privacy and security. Proceedings of International workshop on research challenges in security and privacy for mobile and wireless networks (WSPWN 2006), Miami, Florida (pp. 134-147).

Lindgren, A., Doria, A., & Schelén, O. (2003). Probabilistic routing in intermittently connected networks. *SIGMOBILE Mob. Comput. Commun. Rev.*, *7*(3), 19–20. doi:10.1145/961268.961272

Musolesi, M., & Mascolo, C. (2006). A community based mobility model for ad hoc network research. *Proceedings of the 2nd International Workshop on Multi-hop Ad Hoc Networks: From Theory to Reality*, ACM, New York, NY, USA (pp. 31-38). doi:10.1145/1132983.1132990

Nahrstedt, S., & Nahrstedt, K. (2002). Predictive location-based QoS routing in mobile ad hoc networks. *Proceedings ofIEEE International Conference on Communications ICC 2002* (Vol. 1, pp. 1022-1027).

Nguyen, H.A., Giordano, S., & Puiatti, A. (2007). Probabilistic Routing Protocol for Intermittently Connected Mobile Ad hoc Networks (PROPICMAN). Proceedings of IEEE AOC, Helsinki.

Nguyen, H. A., & Giordano, S. (2009). Routing in Opportunistic Networks. *International Journal of Ambient Computing and Intelligence*, *1*(3), 19–38. doi:10.4018/jaci.2009070102

Nguyen, H. A., & Giordano, S. (2008). Spatiotemporal routing algorithm in opportunistic networks., In World of Wireless, Mobile and Multimedia Networks Proceedings of WoWMoM 2008 International Symposium (pp. 1-6). doi:10.1109/WOWMOM.2008.4594894

Nissen, S. (2003). *Implementation of a Fast Artificial Neural Network Library (FANN).* Copenhagen, Copenhagen: Univ.

Perkins, C., Belding-Royer, E., & Das, S. (2003). Ad hoc On-Demand Distance Vector (AODV) Routing. *IETF RFC 3561.*

Perkins, C. E., & Bhagwat, P. (1994). Highly dynamic Destination-Sequenced Distance-Vector routing (DSDV) for mobile computers. *Proceedings of the conference on Communications architectures, protocols and applications* (pp. 234-244). doi:10.1145/190314.190336

Prasad, P. S., & Agrawal, P. (2010). Movement Prediction in Wireless Networks Using Mobility Traces. Proceedings of the *Consumer Communications and Networking Conference (CCNC)*, *7th IEEE* (pp. 1-5). doi:10.1109/CCNC.2010.5421613

Riedmiller, M., & Braun, H. (1993). A direct adaptive method for faster backpropagation learning: the RPROP algorithm. *Neural Networks, proceedings of the IEEE International Conference on* (Vol. 1, pp. 586 -591). doi:10.1109/ICNN.1993.298623

Sarle, W. S. (Ed.). (1997). *Neural Network FAQ, part 1 of 7: Introduction.* Usenet newsgroup comp. ai.neural-nets. Retrieved from ftp://ftp.sas.com/pub/neural/FAQ.html

Son, D., Helmy, A., & Krishnamachari, B. (2004, July). The Effect of Mobility-Induced Location Errors on Geographic Routing in Mobile Ad Hoc and Sensor Networks: Analysis and Improvement Using Mobility Prediction. *IEEE Transactions on Mobile Computing*, *3*(3), 233–245. doi:10.1109/TMC.2004.28

Spyropoulos, T., Psounis, K., & Raghavendra, C. S. (2005). Spray and wait: an efficient routing scheme for intermittently connected mobile networks. *Proceedings of the 2005 ACM SIGCOMM Workshop on Delay-tolerant Networking*, New York, NY, USA (pp. 252-259). doi:10.1145/1080139.1080143

Spyropoulos, T., Psounis, K., & Raghavendra, C. S. (2007). Spray and Focus: Efficient Mobility-Assisted Routing for Heterogeneous and Correlated Mobility. Pervasive Computing and Communications Workshops Fifth Annual IEEE International Conference (pp. 79-85).

Stojmenovic, I., Russell, M., & Vukojevic, B. (2000). Depth first search and location based localized routing and QoS routing in wireless networks. *Parallel Processing Conference* (pp. 173–180). doi:10.1109/ICPP.2000.876111

Vahdat, A., & Becker, D. (2000). Epidemic Routing for Partially Connected Ad Hoc Networks.

Wang, Y., Jain, S., Martonosi, M., & Fall, K. (2005). Erasure-coding based routing for opportunistic networks. *Proceedings of the 2005 ACM SIGCOMM Workshop on Delay-tolerant Networking*, New York, NY, USA (pp. 229-236). doi:10.1145/1080139.1080140