UTILIZING RECOMMENDER SYSTEMS AS AN ANALYSIS TOOL TO MEASURE NETWORK DYNAMICS

A Dissertation
Submitted to the Graduate Faculty
of the
North Dakota State University
of Agriculture and Applied Science

By

Muhammad Usman Shahid Khan

In Partial Fulfillment of the Requirements
for the Degree of
DOCTOR OF PHILOSOPHY

Major Department:
Electrical and Computer Engineering

March 2015

Fargo, North Dakota

North Dakota State University
Graduate School

Title

Utilizing recommender systems as an analysis tool to measure network dynamics

By

Muhammad Usman Shahid Khan

The Supervisory Committee certifies that this ***disquisition*** complies with North Dakota State University's regulations and meets the accepted standards for the degree of

DOCTOR OF PHILOSOPHY

SUPERVISORY COMMITTEE:

Samee U. Khan
Chair

Jacob S. Glower

Scott Smith

Ying Huang

Approved:

May 13, 2015
Date

Scott Smith
Department Chair

ABSTRACT

Recommender systems apply numerous knowledge discovery techniques to suggest the preferred products, information, or service on contextual data. In our study, we utilize the recommender system for analyzing and measuring the network dynamics. The dynamic factors such as change in network shape or data size affect the performance of the networks and make it harder for the optimization techniques to be used for finding the optimum solution. In our research, we focused on the monitoring and analyzing the dynamic factors involved in two networks: (a) body area networks and (b) road networks; and based on the study proposed the efficient solution for mitigating the negative effects of the dynamic factors involved using recommender systems.

In body area networks, we monitored the dynamically changing body area sensors data and studied the correlation between the sensors' location and activity recognition. We proposed a cloud based framework that has employed a feature descriptor called Local Energy-based Shape Histogram (LESH) to preserve the maximum information of local energy. We have also used the Wearable Action Recognition Database (WARD) dataset to perform the experiments. Based on our study we proposed the best combination of sensors for various activities recognition.

In road networks, we monitored the congestion during large-scale emergency evacuation and proposed efficient route recommendation service that helps in fast and safe evacuation. To respond to emergencies in a fast and an effective manner, it is of critical importance to have efficient evacuation plans that lead to minimum road congestion. The existing approaches, mostly based on multi-objective optimizations, are not scalable enough when involve numerous time varying parameters, such as traffic volume, safety status, and weather conditions. In this study, we propose a new scalable emergency evacuation service that recommends the evacuees with the most preferred routes towards safe locations during a disaster. The evacuees are directed towards those routes that are safe and have least congestion resulting in decreased evacuation time. The results indicated the improved efficiency of our service for average evacuation times and evacuation delays.

ACKNOWLEDGEMENTS

First and foremost thanks to ALLMIGHTY ALLAH/GOD Who has helped me throughout the course of my studies. All of my knowledge, strength, courage, health, and abilities are His blessings upon me and there is no way to fulfill His right to thank Him.

Special thanks to Dr. Samee U. Khan, my advisor, for his help, guidance and innovative ideas. I offer my sincere and deep hearted gratitude to my advisor who always encouraged me, and persistently conveyed the spirit and guidance required for the research. Without his kind guidance and continuous efforts, this disquisition would not have been possible.

Special thanks to my committee members, Dr. Jacob S. Glower, Dr. Scott Smith, and Dr. Ying Huang for their support, guidance and helpful recommendations. Thanks to the Electrical and Computer Engineering staff members Jeffrey Erickson, Laura D. Dallman, and Priscilla Schlenker for all the unconditional help and favor.

I would like to thank my family. Their continuous support is always a source of motivation and encouragement for me. I especially like to thank my father and mother, who are the only and every reason for whatever I am today and whatever I achieved in my life. I also would like to thank my loving wife, my son Abdullah, and my daughter Abeera for their patience, time, and support. They have shown extreme patience and cooperation that resulted in successful completion of this disquisition.

Finally, I owe my heartiest thanks to all my friends and colleagues here in the US and Pakistan, who always helped me in the time of need.

DEDICATION

I would like to dedicate this thesis to my family, especially to my parents and my wife for all the inexplicable love, support, and motivation.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1. INTRODUCTION

1.1. Introduction

In this chapter, we discuss the introduction of the research we have performed during Ph.D. We carried out our research on the monitoring and analysis of dynamically changing factors on different networks for finding efficient solutions using recommender systems. In our research studies, we took the cases of two different networks: body area networks and road networks. In our first case study, we analyzed the body area network and monitored the effect of location of sensor on the activity recognition. Based on our study, we recommended the best possible combination of sensors' locations for different activities recognition. In the second phase of our study, we took the scenario of large scale evacuation. We monitored and analyzed the congestion being created due to the dynamic road conditions such as road densities and provide efficient route recommendation service architecture.

1.2. Activity Recognition in Body Area Networks

The rapid advancements in Information and Communication Technologies (ICTs) have significantly influenced the healthcare domain. The capabilities of conventional clinical healthcare and patient monitoring systems are being enhanced to improve the quality of care by offering remote patient monitoring services and pervasive access to health information [1.1], [1.2]. The remote patient monitoring services are effective means of communicating the symptoms and vital signs to the doctors remotely so that the patients could be informed early about the physiological deteriorations and possible remedies [1.3]. To obtain the information from patients about the observed conditions and for subsequent transmission to the physicians, Wireless Sensor Networks (WSNs) and Body Area Networks (BANs) are employed. Recently the use of BANs for patient monitoring and remote care delivery has increased because of their ability of integrating the on-body sensors and wearable devices, smart phones, Personal Digital Assistants (PDAs), and various other computing devices into a unified system [1.4],[1.5]. The wearable devices and smart band-aids are attached to different body locations to determine certain metrics , such as daily energy expenditure as a result of patients' activities, body temperatures, oxygen saturation, blood pressure, and heart rate monitoring [1.6], [1.7].

In addition, the research has also emphasized on the recognition of the activities performed by the human subjects, such as walking and falling of elderly people [1.8]. Various devices, such as pedometers and accelerometers that are equipped with sensors are being used for monitoring the activities like motion, walk, and exercise. Nonetheless, the location of sensors and wearable devices on human body plays a significant role in accurately recognizing the human activities. The signals that are captured while performing the activities are highly dependent on the location of the sensors on the body [1.9]. It is possible that the same activity may generate different signals when the location of the on-body sensors is changed. Usually, the sensor location and human activity recognition are considered as two separate problems. However, it is important to contemplate the relationship between the sensor location and activity recognition because the correct placement of on-body sensors is a factual characterization for the diagnosis and subsequent treatment [1.10]. For example, the temperature of different areas of human body even measured with the same device varies. Therefore, in healthcare settings requiring the clinical and medication procedures to be adopted based on the observations provided by wearable monitoring devices, the incorrect placement of on-body sensing devices in each use can misguide the clinicians and doctors. Therefore, it is important that the sensors should be located at appropriate locations on body to appropriately utilize the collected data. Moreover, combination of sensors at different body locations also affects the recognition accuracy. Therefore, finding the best combination of sensors at different locations on body is also an important task.

Currently, the BANs are utilizing the smart phones and handheld devices to ensure the ubiquitous access to the healthcare information and services. However, due to the architectural limitations in terms of CPU speed and memory[1.11], the mobile devices seem inadequate to handle huge volumes of data being generated unceasingly. Therefore, it is the high time to integrate the BANs with the Mobile Cloud Computing (MCC) to not only offer pervasive health services but also to manage the large volumes of data. Cloud computing services are already being used in the healthcare domain due to the key benefits, such as scalability, availability, and cost effectiveness [1.12]. The cloud computing model liberates the organizations of the protracted tasks of infrastructure management and development. All of the computational and storage tasks are delegated to the third-party cloud servicers and the users only pay for using the cloud services [1.13]. As stated earlier that the BANs generate huge volumes of data every second, the data generated by several BANs is difficult to handle using the ordinary computing devices. Therefore, it is quite feasible that the data generated by sensing devices is stored to the cloud environment and can be made available to the doctors and physicians on demand [1.11]. Storing the data to the cloud servers will also reinforce the reliability. For example, if the mobile device loses some data due the low battery or any other reason, the data backup is available on the cloud and even if the mobile device is switched off, the execution of the application is not interrupted because it is running in the cloud.

This paper proposes a cloud based framework for activity recognition and to investigate the effects of the on-body sensors' location on the performance of activity recognition. The signals generated as a result of human activities are largely dependent upon the location of the on-body sensors due to the high likelihood of generating a different signal for the same activity on a different body location. Therefore, a true characterization of human activities plays a pivotal role in remotely monitoring the patients requiring therapies and treatments for orthopedic and sports injuries. The sensor data generated by the wearable devices is transmitted to the mobile devices that are subsequently forwarded to the cloud server. The application running in the cloud is responsible for classifying the actions by extracting the features from the continuous data streams. To make the process of activity recognition more effective, an important task is to extract features from the sensor data comprising of human subjects' movements and activities. The data transformation to feature space causes the dimensionality reduction. As a result, data with multiple dimensions can be represented in few or even one dimension in feature space. The local energy response varies with the change in position and the energy generated as a result of movements is distributed in the body. Therefore, to preserve the maximum information of local energy, we employ a methodology called Local Energy-based Shape Histogram (LESH). The LESH features are primarily used in computer vision tasks, such as shape based image retrieval, object tracking, object recognition. The reduction of the data to a fixed size LESH feature helps in detecting the changes in the data as the small changes appears up as large in the fixed size vector. We represent the sensor data by means of vectors where each vector represents the underlying information or energy of the local event with very high level of transformation. The goal of the local feature extraction technique is to represent the local regions in the sensor data matrix/vector effectively and comprehensively. To handle the local energy responses that change with the activity, we generate local histogram by summing the local energy along with the filter orientation in a local region of the sensor data matrix/vector. The feature extraction phase is followed by the applying the classification algorithm on the features data. We employed three classifiers namely, Simple Logistic Regression (SLR), Naïve Bayes classifier, and Sequential Minimal Optimization (SMO) to perform the task of activity recognition. Moreover, to investigate effects of sensor location on correct identification of activities, we used the Wearable Action Recognition Database (WARD) dataset [1.8]. The WARD dataset contains the data for thirteen activities collected from five body sensors located at different positions of 20 different persons.

1.3. Route Recommendation during Large-Scale Evacuation

Natural and man-made disasters, such as tsunamis, earthquakes, floods, and epidemics pose a significant threat to human societies. In response to the growing number of recent disasters, such as the Colorado flood, Oklahoma tornado, Japan’s earth quake, Katrina hurricane, and in particular, the Red River crest that causes flood almost every year in Fargo, North Dakota, the importance and scope of emergency evacuation systems have grown tremendously over the past decade [1.14]. Well-planned evacuation operations and identification of appropriate rescue routes before and during a disaster play a significant role in saving lives and minimizing casualties.

Generally, transportation planning departments consider the peak traffic demands during normal workdays and on special occasions [1.15], [1.16]. However, it is almost impossible to conceive transportation plans for emergency situations, due to which large volumes of traffic involved in mass evacuations is likely to exceed the capacity of road networks that may lead to loss of human lives. For example, due to the lack of proper evacuation plan, 25 people lost their lives in the first 30 minutes while attempting to flee their Oakland Hills neighborhood in California during a wildfire in the year 1991. Moreover, reports indicate that the inefficient evacuation planning in case of the Katrina and Rita hurricane resulted in a heavy traffic jam on the interstates. A similar traffic jam occurred for 20 hours after a winter storm in Atlanta, USA, in January 2014, as the transportation network was incapable of handling the traffic congestion caused by snow and accidents. To prevent such incidents, emergency evacuation plans must be developed to ensure the availability of safest and most efficient evacuation routes for the residents of a structure, region, or city.

The objective of this paper is to develop a scalable service that can guide evacuees towards safe and least congested routes during a disaster. With the integration of Intelligent Transportation System (ITS), the proposed MacroServ service is capable of computing the efficient traffic flows leading to minimum congestion of the roads during an emergency evacuation.

Several works, such as [1.17]–[1.20], have applied multi-objective optimization in evacuation modeling. Generally, optimization-based evacuation models consider several assumptions to optimize parameters, such as route length, shelter locations, and evacuation times. However, as discussed in [1.15], most of the times such assumptions are performance limiting or unrealistic, and do not precisely depict the dynamics of real-life emergency situations. Moreover, the following are the limitations of most of the optimization-based evacuation models that negatively affect the performance of such systems [1.15].

A few evacuation models simulate the traffic flow with static road network characteristics that do not truly depict the real emergency scenarios [1.18], [1.21]. For instance, numerous time varying behavioral, managerial, and stochastic factors, such as number of evacuees and traffic conditions, are involved during an evacuation [1.21]. Such factors may lead to congestion of the paths that were otherwise suggested as optimal by the evacuation modeling approaches.

If time factor is added to optimization problems, such that the static network is expanded over the planning horizon for every time interval, then the corresponding problem space becomes extremely large and there are no known polynomial algorithms for solving such problems [1.19].

Evacuation modeling in most of the optimization-based approaches is formulated as a network flow optimization problem [1.19], [1.22]. However, such approaches are not scalable for the real-world large-sized evacuation networks, due to the high computational complexities. Moreover, such problems are also considered to be NP-hard because of the multi-commodity nature, as evacuees are differentiated by the origin-destination pairs [1.20]. Therefore, solving for the travel demand rates and route flow rates requires simulation, as a closed form expression cannot be captured with optimization models [1.23].

As mentioned earlier, the optimization-based evacuation models consider assumptions for various parameters, such as road capacities, traffic volumes, route distances, and population sizes [1.15]. However, such assumptions can become invalid during a real emergency scenario due to variations in weather conditions, unforeseen conditions of traffic, and possible destruction of transportation infrastructure.

To address the abovementioned limitations, in this paper we propose a scalable service that is capable of performing real-time simulation of dynamic large-scale transportation networks during emergency scenarios. Simulation based evacuation planning by emergency management agencies require faster execution of large-scale vehicular traffic flows. Therefore, we utilize parallel computing to achieve the required scale, size, and speed of the computations. Our proposed service integrates with the ITS to obtain real-time traffic data and utilizes our proposed algorithm to compute the maximum flow of routes and route costs among disaster sites and safety locations [1.24]. Based on the route costs, our proposed service redirects the traffic on alternate preferred routes before the congestion can occur. In this way, evacuees are guided towards the most preferred routes that have the minimum possible risk and the least amount of congestion.

Massive evacuations involve many stochastic factors, such as, degree of compliance of evacuees to evacuation calls, rate of evacuees departing from each household/ area, behavior of drivers, unforeseen traffic loads, and road conditions on transportation network. To depict such factors in our model, we make use of probability distributions, such as (a) Poisson distribution [1.25] and (b) Weibull distribution [1.25]. The aforementioned probability distributions allow us to model emergency evacuation scenarios that closely match with the realistic scenarios.

As a case study, we performed our simulations on the real map of City of Fargo, ND, USA where the Red River crest causes flood almost every year. The gradient (slope) of the Red River averages five inches per mile of length, and drops to 1.5 inches per mile in the region of Drayton-Pembina [1.26]. Due to lack of slope, the Red River tends to pool and cause floods. To model our system, we obtained the data, including road capacities, traffic volumes, speed limits, contours' elevations, historic crest levels of Red River, and historic flood affected areas, from the City of Fargo [1.27] and North Dakota Department of Transportation (NDOT) [1.28]. For our simulations, we considered the population of size 108,000 living at Red River flood zones that needs to be evacuated during a flood. Moreover, the transportation network consists of 7,370 road links and 2,800 intersections.

Our simulation results indicate that the traditional evacuation plans devised by the disaster management agencies are inefficient to handle sudden loads of traffic during an emergency. The sudden evacuations result in traffic jams due to which evacuation time increases. When the evacuees are directed towards the preferred routes using our proposed service, the overall evacuation time significantly decreases. Moreover, the simulation results indicate that the evacuation performance measures are largely dependent on the highway network structure and the number of vehicles produced in an emergency planning zone. In summary, our proposed service is designed to: (a) act as a decision making tool that will enable transportation departments to evaluate and review the emergency evacuation plans by simulating various disaster scenarios, and (b) recommend preferred and efficient routes to the evacuees during the course of a disaster by making use of high-end sensors and the ITS.

1.4. Research Goals and Objectives

The objective of our research is to utilize the recommender systems as an analysis tool to measure the network dynamics and to provide efficient solutions for the respective networks. For case studies, we took body area networks and road networks. In body area network, we picked our goal as the finding the correlation between the sensors' locations and activity recognition. Based on correlation, we proposed the best combinati on of the sensors' locations for efficient recognition of different activities. In road networks, we aimed at the goal of providing efficient, safe, and fast route recommendation during large scale evacuation.

1.5. References

[1.1] Y. Ren, R. W. N. Pazzi, and A. Boukerche, "Monitoring patients via a secure and mobile healthcare system," *Wireless Communications*, IEEE, 17, no. 1, 2010, pp. 59-65.

[1.2] C. Doukas, T. Pliakas, and I. Maglogiannis, "Mobile healthcare information management utilizing Cloud Computing and Android OS," In *32nd Annual International Conference of the IEEE Engineering in Medicine and Biology Society* (EMBC), 2010, pp. 1037-1040.

[1.3] L. Clifton, D. A. Clifton, M. A. F. Pimentel, P. Watkinson, and L.Tarassenko, "Predictive Monitoring of Mobile Patients by Combining Clinical Observations with Data from Wearable Sensors," *IEEE Journal of Biomedical and Health Informatics*, 18, No. 3, 2014, pp. 722-730.

[1.4] O. Diallo, J. Rodrigues, M. Sene, and J. Niu, "Real-time query processing optimization for cloud-based wireless body area networks," *Information Sciences*, 284, 2014, pp. 84-94.

[1.5] S. Pandey, W. Voorsluys, S. Niu, A. Khandoker, and R. Buyya, "An autonomic cloud environment for hosting ECG data analysis services," *Future Generation Computer Systems*, 28, no. 1, 2012, pp. 147-154.

[1.6] M. Quwaider, and Y. Jararweh, "An efficient big data collection in Body Area Networks," In *5th IEEE International Conference on Information and Communication Systems* (ICICS), 2014, pp. 1-6.

[1.7] N. Amini, M. Sarrafzadeh, A. Vahdatpour, and W. Xu, "Accelerometer-based on-body sensor localization for health and medical monitoring applications," *Pervasive and mobile* *computing* 7, no. 6, 2011, pp. 746-760.

[1.8] A. Y. Yang, R. Jafari, S. S. Sastry, and R. Bajcsy, "Distributed recognition of human actions using wearable motion sensor networks," *Journal of Ambient Intelligence and Smart Environments* 1, no. 2, 2009, pp. 103-115.

[1.9] W. Xu, M. Zhang, A. A. Sawchuk, and M. Sarrafzadeh, "Co-recognition of human activity and sensor location via compressed sensing in wearable body sensor networks," In *9th IEEE International Conference on Wearable and Implantable Body Sensor Networks* (BSN), 2012, pp. 124-129.

[1.10] M. Fraz, M. S. Sarfraz, "Text Detection in Still Images and CCTV Videos using LOacal Energy based Shape Histogram Features" Conference: - In *6th International Conference on Computer Vision Theory and Applications* (VISAPP 2011), Vilamoura, Algarve, Portugal, 5-7 March, 2011, pp. 433-436.

[1.11] J. Wan, C. Zou, S. Ullah, C.-F. Lai, M. Zhou, and X. Wang, "Cloud-enabled wireless body area networks for pervasive healthcare," *IEEE Network*, 27, no. 5, 2013, pp. 56-61.

[1.12] A. Abbas and S. U. Khan, "A Review on the State-of-the-Art Privacy Preserving Approaches in E-Health Clouds," *IEEE Journal of Biomedical and Health Informatics*, vol. 18, no. 4, pp. 1431-1441, 2014.

[1.13] A. Abbas, K. Bilal, L. Zhang, and S. U. Khan, "A cloud based health insurance plan recommendation system: A user centered approach, "*Future Generation Computer Systems*, 2014, DOI: DOI: 10.1016/j.future.2014.08.010.

[1.14] J. Li, Q. Li, C. Liu, S. U. Khan, and N. Ghani, "Community-based collaborative information system for emergency management" *Computers & Operations Research*, vol. 42, pp.116–124, 2014.

[1.15] G. Galindo and R. Batta, "Review of recent developments in OR/MS research in disaster operations management," *European Journal of Operational Research*, vol. 230, pp. 201–211, 2013.

[1.16] O. Khalid, M. U. S. Khan, S. U. Khan, and A. Y. Zomaya, "OmniSuggest: A Ubiquitous Cloud based Context Aware Recommendation System for Mobile Social Networks," *IEEE Transactions on Services Computing*, vol. 7, no. 3, pp. 401-414, 2014.

[1.17] Z. Fang, Q. Li, Q. Li, D. Han, and S. Shaw, "A space–time efficiency model for optimizing intra-intersection vehicle–pedestrian evacuation movements," *Transportation Research Part C: Emerging Technologies*, vol. 31, pp. 112–130, 2013.

[1.18] J. Coutinho-Rodrigues, L. Tralhão, L. Alçada-Almeida, "Solving a location-routing problem with a multi-objective approach: the design of urban evacuation plans," *Journal of Transport Geography*, vol. 22, pp. 206–218, 2012.

[1.19] G. J. Lim, S. Zangeneh, M. R. Baharnemati, and T. Assavapokee, "A capacitated network flow optimization approach for short notice evacuation planning," *European Journal of Operational Research*, vol. 223, pp. 234–245, 2012.

[1.20] A. Stepanov and J. M. Smith, "Multi-objective evacuation routing in transportation networks," *European Journal of Operational Research*, vol. 198, pp. 435–446, 2009.

[1.21] M. Saadatseresht, A. Mansourian, and M. Taleai, "Evacuation planning using multiobjective evolutionary optimization approach," *European Journal of Operational Research*, vol. 198, no. 1, pp.305-314, 2009.

[1.22] S. Bretschneider and A. Kimms, "Pattern-based evacuation planning for urban areas," *European Journal of Operational Research*, vol. 216, pp.57–69, 2012.

[1.23] A. Pel, M. Bliemer, and S. Hoogendoorn, "Modelling Traveller Behaviour under

Emergency Evacuation Conditions," *European Journal of Transport and Infrastructure Research*, vol. 11, no. 2, pp. 166-193, 2011.

[1.24] Intelligent Transportation System, http://www.its. dot. gov/, accessed on March 3, 2014.

[1.25] S. M. Ross. "Introduction to probability models, 9th Edition." ISBN-13: 978-0-12-598062-3, *Academic Press*, 2006.

[1.26] Red River Information, http://www.ndsu.edu/fargo_geology/whyflood.htm, accessed on March 3, 2014.

[1.27] City of Fargo website: https://www.cityoffargo.com /Maps/, accessed on March 3, 2014

[1.28] North Dakota Department of Transportation (NDDOT), https://www.dot.nd.gov/, accessed on March 3, 2014.

2. RELATED WORK

2.1. Correlation between Sensor Locations and Activity Recognition

In this section, we present some of the research works that are related to sensor location and human activity recognition in pervasive healthcare monitoring systems. Yang *et al.* [2.1] used low-bandwidth wearable sensors for the classification of continuous human action. In the proposed framework, ℓ^1-minimization based approach is used for classification of valid action segments and rejection of the outlying actions. Contrary to the past approaches that recognize single action, the approach by Yang *et al.* [2.1] is capable of recognizing multiple human actions and utilizes five wearable sensors to recognize thirteen action categories. The approach presented in [2.1] is distributed in the sense that a sensor node becomes active only when an event is detected and transmits the local information to the server. The server utilizes a global classifier to receive the data from the sensors and optimizes the local sensor decisions. However, the approach requires the sensor to be installed on a fixed location on human body that may limit the accuracy in case when the sensor is misplaced.

Amini *et al.* [2.2] presented a technique to automatically recognize the location of on-body wearable devices to ensure the correct and accurate measurements in health monitoring systems. Accelerometers are used to capture the motion data and the device location is estimated through the supervised and unsupervised time series methods for data analysis. Another approach that co-recognizes the sensor location and human activities is presented in [2.3]. The authors used the compressed sensing theory to reconstruct the sparse signals captured from inertial human activities and subsequently classified the activities signals to determine the location of the signals on human body. The approach presented in [2.3] uses only one sensor on the body to recognize the human activity without necessitating a fixed location for the sensor. Zhang *et al.* [2.4] also used compressed sensing theory. The activity signals from all of the classes of training set are represented as the sparse linear combinations via ℓ^1-minimization. In [2.5], a Linear Discriminant Analysis (LDA) based approach for feature extraction is applied and classification is performed using Artificial Neural Networks (ANNs). The group features, such as autoregressive model coefficients, tilt angle, and signal magnitude area are extracted. The approach is claimed to have accuracy of 97.76% for activities classification and transitions. Our approach is different from the above approaches in the sense that we use the local energy to determine the activities performed by the subjects. Moreover, contrary to the approach presented in [2.3] that relies on one body sensor, we use five on-body sensors and our approach is capable accurately recognizing the activities with multiple sensors simultaneously.

2.2. Route Recommendation during Evacuation

Numerous studies conducted in the past addressed various perspectives of emergency evacuation modeling, such as route finding [2.6], shelter site selection [2.7], evacuees' behavior [2.8], and traffic control strategies [2.9]. In recent years, there has been a growing interest in the multi-objective optimization techniques for evacuation route finding problem.

The authors in [2.10] studied demand-based strategies for aggregate-level routing with and without congestion. The authors proposed a network flow model that optimized an evacuation specific lexicographic objective function. The function computes the time dependent evacuation routes for each of the source. However, being a combinatorial optimization problem, the proposed approach is difficult to be solved for large realistic networks. Therefore, the authors utilized two heuristics to solve the problem, but with a tradeoff of solution quality. Lim *et al*. [2.6] modeled evacuation problem as network flow optimization problem. The static network is expanded over the planning horizon for each time interval. However, this makes the optimization problem extremely large to solve. Therefore, the authors proposed a heuristic based solution that utilized Dijkstra's algorithm to compute evacuation paths, and a greedy algorithm to find the maximum flow and exit schedule for each path at each time interval. In a similar study, the authors in [2.11] utilized mixed integer programming for a dynamic network flow optimization problem. The authors proposed a heuristic solution that was applied over the time expanded transportation network, where the time horizon was divided into intervals of equal length. However, time expansion of the network made optimization problem infeasible for large scale evacuation scenarios. Coutinho-Rodrigues *et al.* [2.7] proposed a multi-objective optimization problem to find evacuation paths and the location of shelters for urban evacuation planning. The authors considered many objectives for optimization, such as path lengths, path risks, evacuation times, lengths of backup paths, and number of shelters. The set of primary routes between disaster site and shelter locations were generated with a bi-objective shortest path approach by considering the path lengths and path risks. The model was tested on a smaller network with limited roads and intersections.

Stepanov *et al.* [2.12] proposed an integer programming formulation for route assignment that utilized M/G/c/c state dependent queues to address congestion and time delays on road links. The authors computed a set of evacuation routes with kth shortest path algorithm, and then utilized M/G/c/c model to evaluate the travel time along the shortest paths. A drawback in such approach is that the shortest paths may become congested during real evacuation scenarios due to the presence of numerous unforeseen random factors, such as traffic accidents and weather conditions. The authors in [2.8] developed a traffic simulation framework that assigns evacuees with the predefined routes at the beginning of evacuations. During the journey the evacuees were able to change the routes. The authors studied the effect of non-compliance of evacuation orders by evacuees during evacuations. However, the architectures and implementation details of the proposed framework were not discussed. El-Sergany *et al.* [2.13] proposed a framework for flood disaster management and a transport distribution model for evacuations. The authors utilized linear programming on a small scale scenario with trip distribution matrix among the affected sites and destination shelters.

Huang *et al.* [2.14] presented a centralized traffic control framework for emergency vehicles. The authors utilized A* algorithm to compute three types of paths: (a) primary path between source and destination, (b) secondary path that is disjoint of the primary path, and (c) a path that connects both the primary and the secondary paths. However, authors did not mention any details about the implementation and test data of their framework.

Abdelgawad *et al.* [2.15] presented a multi-objective optimization framework that combines the vehicular traffic and mass transit system for emergency evacuation. The paper investigated the three objectives: (a) minimum in-vehicle travel time, (b) minimum at-origin waiting time, and (c) minimum fleet cost in case of mass transit. However, the authors have not included the real-time changing parameter such as road conditions that affect the real-world evacuation.

It is noteworthy to mention that most of the aforementioned optimization-based evacuation models are unable to scale well for large-scale evacuation scenarios. Therefore, most of the techniques employ various heuristics to reduce the solution space, which results in sub-optimal route recommendations. In contrast to the optimization-based approaches, there also exist some commercial/non-commercial traffic simulation packages, such as INDY [2.16], PARAMICS [2.17], DynusT [2.18], and TransCAD [2.19]. Among the aforementioned, the PARAMICS [2.17] is commercial software and has been utilized mostly for micro-scale simulations. However, a common problem that most of such packages suffer from is the lack of scalability, especially when the network size is large and different from the network under normal conditions. Therefore, to address scalability, we utilized parallel computing in our proposed simulation framework.

2.3. References

[2.1] A. Y. Yang, R. Jafari, S. S. Sastry, and R. Bajcsy, "Distributed recognition of human actions using wearable motion sensor networks," *Journal of Ambient Intelligence and Smart Environments* 1, no. 2, 2009, pp. 103-115.

[2.2] M. Fraz, M. S. Sarfraz, "Text Detection in Still Images and CCTV Videos using LOacal Energy based Shape Histogram Features", In *6th International Conference on Computer Vision Theory and Applications* (VISAPP 2011), Vilamoura, Algarve, Portugal, 5-7 March, 2011, pp. 433-436.

[2.3] W. Xu, M. Zhang, A. A. Sawchuk, and M. Sarrafzadeh, "Co-recognition of human activity and sensor location via compressed sensing in wearable body sensor networks," In *9th IEEE International Conference on Wearable and Implantable Body Sensor Networks* (BSN), 2012, pp. 124-129.

[2.4] M. Zhang, and A. A. Sawchuk, "Human daily activity recognition with sparse representation using wearable sensors," *IEEE Journal of Biomedical and Health Informatics*, 17, no. 3, 2013, pp. 553-560.

[2.5] A. Khan, Y.-K. Lee, S. Lee, and T.-S. Kim, “A triaxial accelerometer based physical-activity recognition via augmented-signal features and a hierarchical recognizer,” *IEEE Transactions on Information Technology in Biomedicine*, vol. 14, no. 5, pp. 1166–1172, Sep. 2010.

[2.6] G. J. Lim, S. Zangeneh, M. R. Baharnemati, and T. Assavapokee, “A capacitated network flow optimization approach for short notice evacuation planning,” *European Journal of Operational Research*, vol. 223, pp. 234–245, 2012.

[2.7] J. Coutinho-Rodrigues, L. Tralhão, L. Alçada-Almeida, “Solving a location-routing problem with a multi-objective approach: the design of urban evacuation plans,” *Journal of Transport Geography*, vol. 22, pp. 206–218, 2012

[2.8] A. Pel, M. Bliemer, and S. Hoogendoorn, “Modelling Traveller Behaviour under Emergency Evacuation Conditions,” *European Journal of Transport and Infrastructure Research*, vol. 11, no. 2, pp. 166-193, 2011.

[2.9] N. T. N. Anh, Y. Chevaleyre, and Jean Daniel Zucker, “Optimizing the Placement of Evacuation Signs on Road Network with Time and Casualties in Case of a Tsunami,” *IEEE 21st International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises* (WETICE), pp. 394-396, 2012.

[2.10] D. R. Bish and H. D. Sherali, “Aggregate-level demand management in evacuation planning,” *European Journal of Operational Research*, vol. 224, pp. 79–92, 2013.

[2.11] S. Bretschneider and A. Kimms, “Pattern-based evacuation planning for urban areas,” *European Journal of Operational Research*, vol. 216, pp.57–69, 2012.

[2.12] A. Stepanov and J. M. Smith, “Multi-objective evacuation routing in transportation networks," *European Journal of Operational Research*, vol. 198, pp. 435–446, 2009.

[2.13] A. T. El-Sergany and S. Alam, "Trip Distribution Model for Flood Disaster Evacuation Operation," *ITE Journal*, vol. 82, pp. 43-37, 2012.

[2.14] C. Huang, C. Kung, C. Yang, C. Tseng, C.-H.A. Chou, "A Centralized Traffic Control Mechanism for Evacuation of Emergency Vehicles Using the DSRC Protocol," *4th International Symposium on Wireless Pervasive Computing*, 2009.

[2.15] H. Abdelgawad, B. Abdulhai, and M. Wahba, "Multiobjective optimization for multimodal evacuation," *Transportation Research Record: Journal of the Transportation Research Board* 2196, no. 1, pp.21-33, 2010.

[2.16] http://www.floodsite.net/html/cd_task17-19/indy.html, accessed on March 3, 2014.

[2.17] PARAMICS, http://www.paramics-online.com/, accessed on March 3, 2014.

[2.18] DynusT, http://dynust.net/, accessed on March 3, 2014.

[2.19] TransCAD, http://www.caliper.com/tcovu.htm, accessed on March 3, 2014.

3. ON THE CORRELATION OF SENSOR LOCATION AND HUMAN ACTIVITY RECOGNITION IN BODY AREA NETWORKS (BANS)

This paper[1] is submitted to *IEEE System Journal (ISJ)*.

3.1. Introduction

The rapid advancements in Information and Communication Technologies (ICTs) have significantly influenced the healthcare domain. The capabilities of conventional clinical healthcare and patient monitoring systems are being enhanced to improve the quality of care by offering remote patient monitoring services and pervasive access to health information [3.1], [3.2]. The remote patient monitoring services are effective means of communicating the symptoms and vital signs to the doctors remotely so that the patients could be informed early about the physiological deteriorations and possible remedies [3.3]. To obtain the information from patients about the observed conditions and for subsequent transmission to the physicians, Wireless Sensor Networks (WSNs) and Body Area Networks (BANs) are employed. Recently the use of BANs for patient monitoring and remote care delivery has increased because of their ability of integrating the on-body sensors and wearable devices, smart phones, Personal Digital Assistants (PDAs), and various other

[1] The material in this chapter was co-authored by Muhammad Usman Shahid Khan, Assad Abbas, Mazhar Ali, Muhammad Jawad, Samee U. Khan, Keqin Li, and Albert Y. Zomaya. Muhammad Usman Shahid Khan had primary responsibility for conducting experiments and collecting results. Muhammad Usman Shahid Khan was the primary developer of the conclusions that are advanced here. Muhammad Usman Shahid Khan also drafted and revised all versions of this chapter.

computing devices into a unified system [3.4],[3.5]. The wearable devices and smart band-aids are attached to different body locations to determine certain metrics , such as daily energy expenditure as a result of patients' activities, body temperatures, oxygen saturation, blood pressure, and heart rate monitoring [3.6], [3.7].

In addition, the research has also emphasized on the recognition of the activities performed by the human subjects, such as walking and falling of elderly people [3.8]. Various devices, such as pedometers and accelerometers that are equipped with sensors are being used for monitoring the activities like motion, walk, and exercise. Nonetheless, the location of sensors and wearable devices on human body plays a significant role in accurately recognizing the human activities. The signals that are captured while performing the activities are highly dependent on the location of the sensors on the body [3.9]. It is possible that the same activity may generate different signals when the location of the on-body sensors is changed. Usually, the sensor location and human activity recognition are considered as two separate problems. However, it is important to contemplate the relationship between the sensor location and activity recognition because the correct placement of on-body sensors is a factual characterization for the diagnosis and subsequent treatment [3.10]. For example, the temperature of different areas of human body even measured with the same device varies. Therefore, in healthcare settings requiring the clinical and medication procedures to be adopted based on the observations provided by wearable monitoring devices, the incorrect placement of on-body sensing devices in each use can misguide the clinicians and doctors. Therefore, it is important that the sensors should be located at appropriate locations on body to appropriately utilize the collected data. Moreover, combination of sensors at different body locations also affects the recognition accuracy. Therefore, finding the best combination of sensors at different locations on body is also an important task.

Currently, the BANs are utilizing the smart phones and handheld devices to ensure the ubiquitous access to the healthcare information and services. However, due to the architectural limitations in terms of CPU speed and memory[3.11], the mobile devices seem inadequate to handle huge volumes of data being generated unceasingly. Therefore, it is the high time to integrate the BANs with the Mobile Cloud Computing (MCC) to not only offer pervasive health services but also to manage the large volumes of data. Cloud computing services are already being used in the healthcare domain due to the key benefits, such as scalability, availability, and cost effectiveness [3.12]. The cloud computing model liberates the organizations of the protracted tasks of infrastructure management and development. All of the computational and storage tasks are delegated to the third-party cloud servicers and the users only pay for using the cloud services [3.13]. As stated earlier that the BANs generate huge volumes of data every second, the data generated by several BANs is difficult to handle using the ordinary computing devices. Therefore, it is quite feasible that the data generated by sensing devices is stored to the cloud environment and can be made available to the doctors and physicians on demand[3.11]. Storing the data to the cloud servers will also reinforce the reliability. For example, if the mobile device loses some data due the low battery or any other reason, the data backup is available on the cloud and even if the mobile device is switched off, the execution of the application is not interrupted because it is running in the cloud.

This paper proposes a cloud based framework for activity recognition and to investigate the effects of the on-body sensors' location on the performance of activity recognition. The signals generated as a result of human activities are largely dependent upon the location of the on-body sensors due to the high likelihood of generating a different signal for the same activity on a different body location. Therefore, a true characterization of human activities plays a pivotal role in remotely monitoring the patients requiring therapies and treatments for orthopedic and sports injuries. The sensor data generated by the wearable devices is transmitted to the mobile devices that are subsequently forwarded to the cloud server. The application running in the cloud is responsible for classifying the actions by extracting the features from the continuous data streams. To make the process of activity recognition more effective, an important task is to extract features from the sensor data comprising of human subjects' movements and activities. The data transformation to feature space causes the dimensionality reduction. As a result, data with multiple dimensions can be represented in few or even one dimension in feature space. The local energy response varies with the change in position and the energy generated as a result of movements is distributed in the body. Therefore, to preserve the maximum information of local energy, we employ a methodology called Local Energy-based Shape Histogram (LESH). The LESH features are primarily used in computer vision tasks, such as shape based image retrieval, object tracking, object recognition. The reduction of the data to a fixed size LESH feature helps in detecting the changes in the data as the small changes appears up as large in the fixed size vector. We represent the sensor data by means of vectors where each vector represents the underlying information or energy of the local event with very high level of transformation. The goal of the local feature extraction technique is to represent the local regions in the sensor data matrix/vector effectively and comprehensively. To handle the local energy responses that change with the activity, we generate local histogram by summing the local

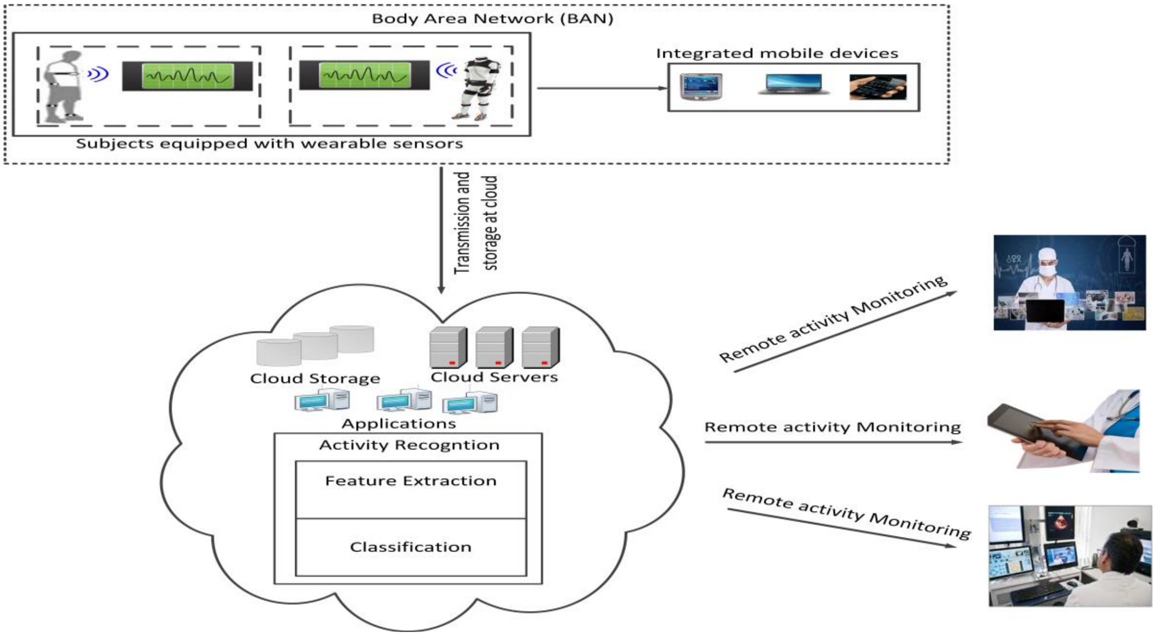

Body Area Network (BAN)
Integrated mobile devices
Subjects equipped with wearable sensors
Transmission and storage at cloud
Cloud Storage
Cloud Servers
Applications
Activity Recogntion
Feature Extraction
Classification
Remote activity Monitoring
Remote activity Monitoring
Remote activity Monitoring

Fig.3.1. Proposed System Architecture for Activities Recognition

energy along with the filter orientation in a local region of the sensor data matrix/vector. The feature extraction phase is followed by the applying the classification algorithm on the features data. We employed three classifiers namely, Simple Logistic Regression (SLR), Naïve Bayes classifier, and Sequential Minimal Optimization (SMO) to perform the task of activity recognition. Moreover, to investigate effects of sensor location on correct identification of activities, we used the Wearable Action Recognition Database (WARD) dataset [3.8]. The WARD dataset contains the data for thirteen activities collected from five body sensors located at different positions of 20 different persons. The main contributions of the proposed framework are as follows:

- We successfully applied image feature extraction approach called LESH to extract the features from the signals generated as a result of human activities. To the best of our knowledge, the LESH approach has not been used in past for feature extraction in BANs.

- The performance in terms of activity recognition is evaluated by applying classifiers, such as the SLR, Naïve Bayes, and the SMO.
- We also investigate the effects of sensor location and the best combination of sensors on activity recognition using the WARD dataset.
- We propose the assimilation of cloud computing services to handle large amount of sensors data with the mobile devices to ensure the ubiquitous access to health data for remote monitoring.

The paper is organized as follows. Section 3.2 presents the proposed framework for activity recognition. Section 3.3 presents a detailed analysis by elaborating the effects of sensor location on activity recognition. The related work is discussed in Section 3.4 whereas Section 3.5 concludes the paper.

3.2. The Proposed Framework

The proposed framework comprises of two major modules namely, (a) the preprocessing component and (b) the activity recognition module. Each of the modules is presented below. The architecture of the proposed framework is presented in Fig. 3.1.

3.2.1. Preprocessing Component

The first step in the proposed framework for activity recognition is to capture the motion signals through the deployment of on-body sensors. We adopted the sensor deployment methodology presented by Yang *et al.* [3.8] to effectively model and monitor the full body motion. The details about the location of the five sensors used are given in Table 3.1.

In the proposed framework, the activities performed by the subjects cause the generation of signals. The framework uses the integrated devices, such as the smart phones, PDAs, and laptops to receive the sensor data in the form of signals. However, due to the intrinsic limitations of the aforementioned integrated devices in terms of processing, storage, and memory, the data received from the wearable sensors is forwarded to the cloud servers.

Table 3.1. Locations of On-Body Sensors

Sensor #	Body Location
Sensor 1	Outside center of the lower left forearm joint. The y-axis of the gyroscope points to the hand.
Sensor 2	Outside center of the lower right forearm joint. The y-axis of the gyroscope points to the hand.
Sensor 3	Front center of the waist. The x-axis of the gyroscope points down.
Sensor 4	Outside center of the left ankle. The y-axis of the gyroscope points to the foot.
Sensor 5	Outside center of the right ankle. The y-axis of the gyroscope points to the foot.

3.2.2. Activity Recognition

The proposed framework causes the generation of large volumes of data from the wearable sensors as a result of the human activities that is subsequently forwarded to the cloud environment. The use of cloud computing services in the proposed environment is appropriate due to the ability to procure on-demand computing and storage services [3.14]. Therefore, it is certainly desirable to delegate the task of activity recognition to the cloud. The client application running in the cloud is responsible for performing the task of activity recognition by first extracting the features from the continuous data streams and then using the classification algorithms to classify the activities. The tasks of feature extraction and classification are presented in Section 2.2.1.

3.2.2.1. Feature Extraction and Classification

To increase the classification accuracy of activity recognition in BANs, an important task is feature extraction from the sensor data. Various studies, such as [3.7] and [3.9] have focused on extracting features from the activities like walking or non-walking. The aforementioned approaches lack in extracting the robust features to correctly identify the activities involving frequent transitions, for example walking forward, turning left, and jogging. Consequently, the recognition accuracy for the activities involving inter-posture transitions might be low. Therefore, robust feature extraction approaches are needed to attain the higher recognition accuracy.

Local feature extraction is a method to describe a local region/part of a matrix/vector by using a specific measure or transformation rules. The method calculates a unique solution of the local region for any specific event. Whenever the same event occurs, same contents of matrix/vector will encounter at the same place. Therefore, same unique solution will be found again. However, the solution has certain degree of invariance related to the noise present in the sensor. The goal of the local feature extraction technique is to represent local regions in the sensor data matrix/vector effectively and comprehensively. In body sensor networks, the sensor data is represented by either a high dimensional vector containing joint movements data during an activity (that is also called as holistic representation) or a set of vectors where each vector represents the underlying information of the local event with the very high level of transformation (local representation). The features in the aforesaid histogram have good capabilities to differentiate between the disparities among various activities.

In this paper, we use a feature descriptor named Local Energy-based Shape Histogram (LESH) that uses local energy model for features perception [3.15]. The method has been used successfully in the applications, such as image processing [3.16] and data mining [3.10]. The local energy (information) of the matrix is obtained by Gabor Filtering method [3.17]. The local energy model was first proposed by Morrone and Owens in 1987 [3.18]. The model states that features are preserved at points where the local frequency components are in the phase for maximum amount of time during an event, as shown in Eq. (3.1).

$$E(X) = \max_{\emptyset(X)\in[0,2\pi]} - \frac{\sum_n A_n cos(\emptyset_n(x) - \overline{\emptyset}(x))}{\sum_n A_n} \tag{3.1}$$

In Eq. (3.1), $\mathrm{A_n}$ and $\emptyset_\mathrm{n}$ are the magnitude and phase of the n-th Fourier component. The magnitude and phase information of the local region is extracted by convolving the sensor data matrix/vector with a bank of Gabor wavelet kernels. The produced result is a complex value that gives magnitude and phase of each of the local regions of the sensor data matrix/vector.

$$G_{u,v}(e_n, o_n) = M * \aleph_{u,v}(z), \tag{3.2}$$

where the bank of Gabor kernel is $\aleph_{\mathrm{u,v}}$, M is the sensor data matrix and u, and v are the scale and orientation that are selected as 5 and 8, respectively. The discrete cosine transformation is applied as a filter to get the proper weighting of each region and a noise cancellation factor T. Therefore, the local energy is calculated as:

$$E = \frac{\sum W(x)\lfloor A_n(x)\big(\cos\big(\varphi_n(x) - \bar{\varphi}(x)\big) - \big|\sin\big(\varphi_n(x) - \bar{\varphi}(x)\big)\big|\big) - T\rfloor}{\sum_n A_n(x) + \varepsilon} \tag{3.3}$$

The value of E is normalized by summing all of the frequency components in the region that make the local magnitude and phase of the region independent of the overall magnitude and phase of the matrix. Motivated by the described facts of local energy response that changes with the change in activity, we generate a local histogram by summing the local energy along with the filter orientation in a local region of the sensor data matrix/vector. The local histograms are extracted for each region of the matrix and then concatenated together to form a feature descriptor vector for each activity and sensor. The features represent the local energy model during various activities and postures. We also assign a label tag to each region of the matrix/vector. The labels are useful in obtaining the information of the local region having the highest energy among all of the local regions. The local histogram h is calculated as:

$$h_{a,b} = \sum w_a \times E \times \delta(L - b) \tag{3.4}$$

Where b is the current bin of the histogram, L is the label tag vector, E is the local energy described in Eq. (3.3), and w is the Gaussian weighted function. Each sensor has a tri-axial accelerometer and a bi-axial gyroscope. Each axis in the accelerometer and the gyroscope is reported as a 12-bit value to the node, indicating the values in the range of ±2g and ±500◦/s for the accelerometer and gyroscope, respectively. The size of the sensor data is 445×5, where 5 corresponds to the sensor readings from 3-axis accelerometer and 2-axis gyroscope on one sensor node, and t=445 represents the length of the trial sequence. As the frequency of the sensor is 100Hz, the time step will be 0.01second and total activity time is 4.45 seconds. Motivated by the fact that the local energy response varies with respect to the change in position and rotation, the position changing pattern measured by the accelerometer and rotation changing pattern measured by gyroscope is unique for every activity in our case. In Eq. 3.3, the normalization by adding the amplitudes of all of the elements, makes it independent of the variation in the data of accelerometer and gyroscope, and also makes it invariant to the sensor noise Therefore, to preserve maximum information of local energy, we segment our sensor data into sixteen regions and measure local energy from the accelerometer and gyroscope data by convolving with the Gabor filter of scale five (5) and orientation eight (8), as shown in Eq. (3.3). As the column length of the sensor data is five, the scale length in Gabor filter is selected as five. The orientation of the filter is selected by hit and trial method. To extract the maximum information from the sensor data, we made eight bins local histogram for each of the sixteen regions corresponding to 8-filter orientation. Each bin in the histogram defines a unique energy level and we place all the energy data points obtained from Eq. (3.3) into corresponding energy level. That makes a unique 8 bins energy histogram of the local region for any specific event. As stated earlier, we divided our sensor data matrix M into sixteen equal parts that makes a total of 16 histograms. Therefore, the histograms are combined to generate a 16×8 = 128-dimensional feature vector for each sensor data. Moreover, 128-dimensional feature vector is easy to use in recognition rather than whole 445×5 sensor data. Furthermore, the LESH feature vector is unique for every activity. Therefore, the use of LESH in activity recognition improves overall accuracy of the algorithm.

To investigate the effects of sensor location on the activity recognition, the task following the feature extraction using the LESH based approach is the classification of activities. Therefore, we used three classifiers namely the Simple Logistic Regression (SLR), Naïve Bayes classifier, and Sequential Minimal Optimization (SMO). The performance of each of the classifiers in terms of activity recognition and their correlation with the sensor location are detailed in Section 3.3.

Table 3.2. Activity Classes in the WARD Dataset

Activity	Action Class	Activity	Action Class
1	Rest at Standing (ReSt)	8	Turn right
2	Rest at Sitting (ReSi)	9	Go upstairs
3	Rest at Lying (ReLi):	10	Go downstairs
4	Walk forward (WaFo)	11	Jog (Jog)
5	Walk forward left-circle (WaLe)	12	Jump (Jump)
6	Walk forward right-circle (WaRi)	13	Push wheelchair
7	Turn left (TuLe)		

3.3. Performance Evaluation

We used the WARD dataset to evaluate the performance in terms of activity recognition. The brief description of each of the classifiers used is given below.

Simple Logistic Regression (SLR): The SLR is a standard classification method used to predict the class labels of categorical dependent variable on the basis of one or more features [3.19]. A logistic function is used to model the probabilities that describe the possible outcomes.

Naïve Bayes: Naïve Bayes is a probabilistic classifier with high predictive capabilities. The model operates on the assumption of high attribute or feature independence to exhibit high prediction capabilities [3.20].

Sequential Minimal Optimization (SMO): The SMO is the learning algorithm for Support Vector Machines (SVMs) and utilizes the analytic Quadratic Programming (QP). The SMO works in an iterative manner by breaking the problem in a series of small subproblems that are subsequently solved analytically. Besides the disease classification, the SMO has been used for human activity recognition [3.21]. The performance was evaluated by using a common accuracy metric called F-measure. The F-measure is the harmonic mean of both the precision and recall values. F-measure is calculated as below:

$$\mathrm{F-measure} = \frac{2\mathrm{TP}}{2\mathrm{TP} + \mathrm{TP}\ + \mathrm{FN}} \tag{3.5}$$

We used k-fold cross validation with k=10 to determine the classification accuracy. In k-fold cross validation method, the dataset is divided into k-folds. One of the folds is selected as the testing fold whereas the remaining k-1 folds work as the training folds [3.22]. An advantage of repeating the process k-times is that all of the examples in the dataset are utilized for analysis.

Results exhibit that the location of sensors on body plays significant role in the activity recognition with precision. The accuracy in terms of F-measure obtained from the three classifiers for the thirteen different activities listed in Table 3.2 and for five different sensor locations are presented in Fig. 3.2 (a) – Fig.3.2 (m).

We tested the activity recognition accuracy by applying, (a) one sensor, (b) three sensors, and (c) all five sensors. It can be observed from Fig. 3.2 (a) that for Activity 1 (ReSt), by using all of the five sensors resulted in more than 86% F-measure score for the Naïve Bayes. The F-measure score for the SLR and SMO with all the five sensors were also observed approximately 81% for Activity 1. However, using the Naïve Bayes classification technique with only Sensor 1, Activity 1 was recognized with 85% score. Interestingly, by using different combinations of sensors, the recognition of Activity 1 is even more increased. For example, with Naïve Bayes and the combination of Sensor 1, Sensor 2, and Sensor 3, the F-measure score increases up to 95% (as shown in Table 3.4). Similarly, for the same activity with Sensor1, Sensor 3, and Sensor 4, the recognition rate of the SMO classification technique was 91%. The complete F-measure scores for all the activities by using different sensors for the Naïve Bayes, the SLR, and the SMO are presented in Table 3.4, Table 3.5, and Table 3.6, respectively.

The F-measure scores for recognition of Activity 2 (ReSi) are presented in Fig. 3.2 (b). Using all of the five sensors for the three classifiers resulted in recognition accuracy of 68% for Activity 2. However, when tested with different combinations of three sensors, the results increased to 78% with the Naïve Bayes and 74% with the SLR. Each of the sensors namely, Sensor 1, Sensor 2, and Sensor 5 played important role in the highly precise recognition of Activity 2 using the Naïve Bayes classification technique. However, it was observed that using Sensor 4 with any combination of sensors resulted in a reduced recognition rate. A similar observation was made for Sensor 2 with the SLR classifier.

For Activity 3 (ReLi), it can be observed from Fig. 3.2 (c) and Table 3.4—Table 3.6 that not only the combination of Sensor 2, Senor 3, and Sensor 4 results in high recognition but they are also able to individually recognize the Activity 3 with the high F-measure score. However, the F-measure score using the Naïve Bayes remains up to 77% with the combination of Sensor 1, Senor 2, and Sensor 3. On the other hand, the combination of Sensor 2, Sensor 3, and Sensor 4 turned out with approximately 92% F-measure score using the SMO classification technique. The score in recognizing the Activity 4 (WaFo) were reasonably high using all of the five sensors at the same time with the SLR and SMO techniques whereas the accuracy for the same activity by applying the Naïve Bayes was observed extremely low as shown in Fig 3.2 (d). However, with Naïve Bayes, Sensor 3 individually exhibited high accuracy of 41% while low accuracy results were observed with all of the combinations of different sensors. For Activity 4, the combination of Sensors 1, Sensor 3, and Sensor 4 turned out with the maximum F-measure score of 55% using the Naïve Bayes classification technique.

Fig. 3.2 (e) presents the F-measure scores for the Activity 5 (WaLe). Again, the Naïve Bayes classification was observed significantly low in recognizing the Activity 5. Using the Naïve Bayes, the highest F-measure score of 45% was observed with the combination of Sensor 1, Senor 4, and Sensor 5 (as shown in Table 3.4). The SLR classification technique recognized the Activity 5 with 68% F-measure score with the combination of Sensor 1, Sensor 3, and Sensor 5 (as shown in Table 3.5). For Activity 5, the SMO produced the highest accuracy (63%) only when all of the five sensors were used together.

The F-measure scores for the recognition of Activity 6 (WaRi) are presented in Fig. 3.2 (f). The SLR and the SMO classification techniques produced the highest score of 59% when all the sensors were used together. The Naïve Bayes classification technique recognizes the Activity 6 with 52% accuracy when the combination of Senor 1, Sensor 4, and Sensor 5 is used.

The Naïve Bayes classification technique performed better in recognizing the Activity 7 (TuLe) as shown in Fig. 3.2 (g) with 65% F-measure score. The SLR classification technique recognized the Activity 7 with approximately 71% with the combination of Senor 3, Sensor 4, and Sensor 5 as shown in Table 3.4. The SMO recognized the Activity 7 with 67% accuracy with the combination of Sensor 1, Sensor 4, and Sensor 5 (as shown in Table 3.6).

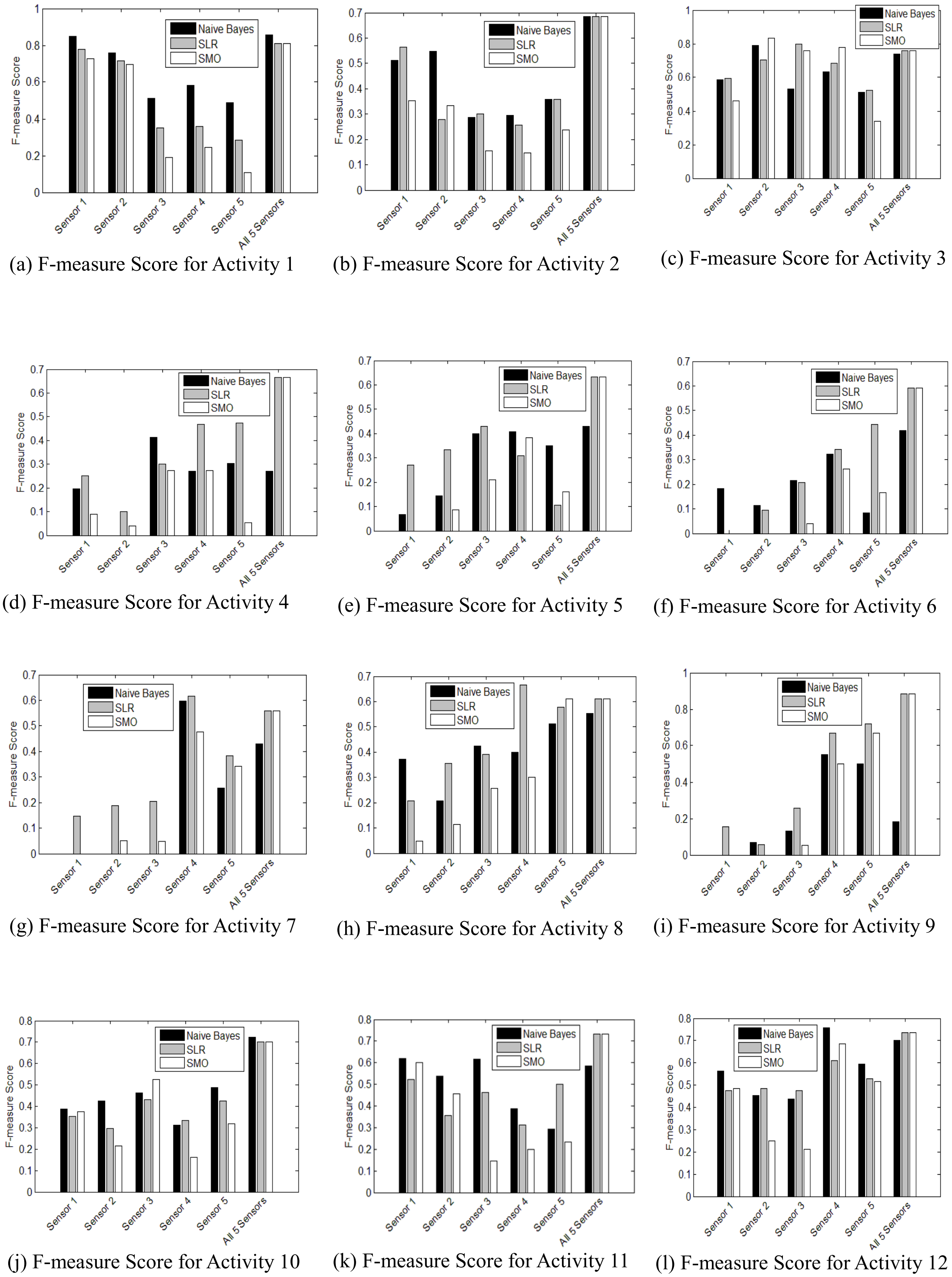
Activity 8 (TuRi) is best recognized using combination of Senor 3, Sensor 4, and Sensor 5 with 74% F-measure as shown in Table 3.4 using the SLR. The SMO recognized the Activity 8 with 70% accuracy with different combinations of three sensors as shown in Table 3.5. With only Sensor 5, the SMO can recognize the activity with 60% score. The Naïve Bayes classifier with the combination of Sensor 1, Sensor 4, and Sensor 5 as shown in Fig. 3.2 (h) recognized the Activity 8 with 62% accuracy.

Fig. 3.2 (i) shows the F-measure scores for recognizing the Activity 9 (Up) using the three classification techniques. The SLR and SMO recognize the Activity 9 with high ac curacy when all of the sensors are used. The F- measure score using the SLR reaches up to 92% with the combination of Sensor 3, Sensor 4, and Sensor 5.

Fig. 3.2 (j) presents the F-measure score for recognition of Activity 10 (Down). The three classification techniques recognized the Activity 10 with high accuracy when all of the sensors were used. However, the for Naïve Bayes, the F-measure score reaches up to 77% with the combination of Sensor 1, Sensor 3, and Sensor 4 . The SLR recognized the Activity 10 with 74% accuracy when using the combination of Sensor 1, Sensor 2, and Sensor 3. Similarly, the SMO recognized the Activity 10 with 70% accuracy with combination of Sensor 1, Sensor 2, and Sensor 3. It can be observed for Fig. 3.2 (k), that using all the five sensors together yielded high recognition accuracy for Activity 11 (Jog). The F-measure score for the SMO was observed the highest (81%) when the combination of Sensor 1, Sensor 3, and Sensor 4 is used. The Naïve Bayes recognized the Activity 11 with 74% score when the combination of Sensor 1, Sensor 3, and Sensor 4 was used. The recognition accuracy of the SLR was approximately 76% with the combination of Sensor 1, Sensor 3, and Sensor 4.

Fig. 3.2 (l) shows the F-measure scores of the three classification techniques with individual sensors and using all the five sensors together for Activity 12 (Jump). Each of the three classifiers exhibited recognition accuracy for Activity 12 with the Sensor 4. However, using the combination of three sensors, such as Sensor 1, Sensor 2, and Sensor 4, the SMO achieved the accuracy of 84%. Similarly, the Naïve Bayes exhibited accuracy of 81% with the combination of Sensor 1, Sensor 4, and Sensor 5 that is slightly better than that of Sensor 4 as shown in Table 3.3. On the other hand, the SLR was able to achieve the highest accuracy of 76% with the combination of Sensor 1, Sensor 2, and Sensor 5 for Activity 12.

Fig. 3.2 (m) shows the F-measure scores for the recognition of Activity 13 (Push) using the three different classification techniques. The SMO accurately recognizes the Activity 13 with 90% accuracy using the Sensor 1 alone. However, using the combination of Sensor 1, Sensor 2, and Sensor 3, the F-measure scores increased to 92%. The SLR recognized the Activity 13 with 90% accuracy using the combination of Sensor 2, Sensor 3, and Sensor 4. Similarly, the Naïve Bayes recognized the activity with 85% score using the combination of Sensor 1, Sensor 2, and Sensor 3. It can be observed from the analysis in terms of activity recognition accuracy as depicted in Table 3.4—Table 3.6 and Fig 3.2—Fig. 3.14 that neither a single classification technique can precisely recognize all of the activities nor any single combination of sensors works best for all the activities. However, we reach an interesting conclusion that considering the transient nature of the most of the activities in the WARD dataset, using more than one sensor simultaneously yielded high recognition accuracy. Table 3.3 summarizes the results that represent the best combination of sensors and classification technique to recognize the particular activity.



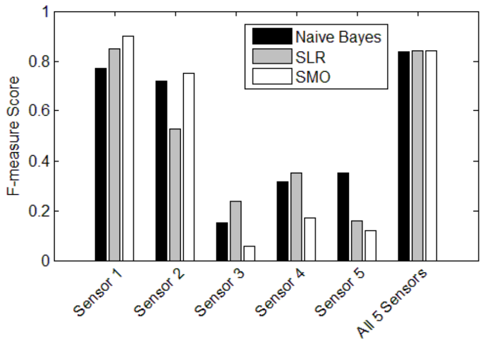
(a) F-measure Score for Activity 1 (b) F-measure Score for Activity 2 (c) F-measure Score for Activity 3

(d) F-measure Score for Activity 4 (e) F-measure Score for Activity 5 (f) F-measure Score for Activity 6

(g) F-measure Score for Activity 7 (h) F-measure Score for Activity 8 (i) F-measure Score for Activity 9

(j) F-measure Score for Activity 10 (k) F-measure Score for Activity 11 (l) F-measure Score for Activity 12

Fig.3.2. F-measure Scores for Different Activities (a) to (m)



(m) F-measure Score for Activity 13

Fig.3.2. F-measure Scores for Different Activities (a) to (m) (continued)

Table 3.3. Summarized Results Depicting the Best Combinations of Sensors for Activity Recognition

Activity	Combination of Sensors	Technique	Highest F-Measure score
a1	1,2,3	Naïve Bayes	0.95
a2	1,2,5	Naïve Bayes	0.78
a3	2,3,4	SMO	0.92
a4	All 5 Sensors	SMO	0.67
a5	1,3,5	SLR	0.68
a6	1,4,5	SLR	0.6
a7	3,4,5	SLR	0.71
a8	3,4,5	SLR	0.74
a9	3,4,5	SLR	0.92
a10	1,3,4	Naïve Bayes	0.77
a11	1,3,4	SMO	0.81
a12	1,2,4	SMO	0.83
a13	1,2,3	SMO	0.92

3.4. Related Work

In this section, we present some of the research works that are related to sensor location and human activity recognition in pervasive healthcare monitoring systems. Yang *et al.* [3.8] used low-bandwidth wearable sensors for the classification of continuous human action. In the proposed framework, ℓ^1-minimization based approach is used for classification of valid action segments and rejection of the outlying actions. Contrary to the past approaches that recognize single action, the approach by Yang *et al.* [3.8] is capable of recognizing multiple human actions and utilizes five wearable sensors to recognize thirteen action categories. The approach presented in [3.8] is distributed in the sense that a sensor node becomes active only when an event is detected and transmits the local information to the server. The server utilizes a global classifier to receive the data from the sensors and optimizes the local sensor decisions. However, the approach requires the sensor to be installed on a fixed location on human body that may limit the accuracy in case when the sensor is misplaced.

Amini *et al.* [3.10] presented a technique to automatically recognize the location of on-body wearable devices to ensure the correct and accurate measurements in health monitoring systems. Accelerometers are used to capture the motion data and the device location is estimated through the supervised and unsupervised time series methods for data analysis. Another approach that co-recognizes the sensor location and human activities is presented in [3.9]. The authors used the compressed sensing theory to reconstruct the sparse signals captured from inertial human activities and subsequently classified the activities signals to determine the location of the signals on human body. The approach presented in [3.9] uses only one sensor on the body to recognize the human activity without necessitating a fixed location for the sensor. Zhang *et al.* [3.23] also used compressed sensing theory. The activity signals from all of the classes of training set are represented as the sparse linear combinations via ℓ^1-minimization. In [3.24], a Linear Discriminant Analysis (LDA) based approach for feature extraction is applied and classification is performed using Artificial Neural Networks (ANNs). The group features, such as autoregressive model coefficients, tilt angle, and signal magnitude area are extracted. The approach is claimed to have accuracy of 97.76% for activities classification and transitions. Our approach is different from the above approaches in the sense that we use the local energy to determine the activities performed by the subjects. Moreover, contrary to the approach presented in [3.9] that relies on one body sensor, we use five on-body sensors and our approach is capable accurately recognizing the activities with multiple sensors simultaneously.

3.5. Conclusions

We presented a cloud based framework that integrates the cloud computing services with the BANs for human activity recognition. We also investigated the effects of on-body sensor locations on accuracy of activity recognition. Due to the varying energy responses that change with the position and rotation, we employed a methodology called Local Energy Based Shape Histogram (LESH) for feature extraction. The purpose of feature extraction was to represent the local regions in the sensor data matrix more comprehensively that ultimately results in the improved recognition accuracy. The LESH approach preserves maximum information about local energy by segmenting the sensors data in various regions. Three classifiers namely, the SLR, Naïve Bayes, and the SMO were used to study the relationship between the sensor location and human activity recognition. A detailed analysis shows that the task of activity recognition is highly dependent on the location of on-body sensors. Particularly, for the activities that are more dynamic and transient, using more than one sensor on different body locations simultaneously yields high recognition accuracy. The presented framework effectively integrates the cloud computing services with the BANs and has potential to be adopted practically for remote patient monitoring and therapies.

Table 3.4. Detailed Results for Different Combinations of Sensors for Naïve Bayes

Class	S 1	S 2	S 3	S 4	S 5	All 5	S 123	S 124	S125	S 134	S 135	S145	S 234	S 235	S 245	S 345
a1	0.78	0.71	0.35	0.36	0.29	0.81	0.85	0.87	0.78	0.91	0.79	0.88	0.6	0.58	0.55	0.42
a2	0.56	0.28	0.3	0.26	0.36	0.68	0.62	0.68	0.6	0.74	0.69	0.53	0.5	0.49	0.42	0.35
a3	0.6	0.7	0.8	0.68	0.52	0.76	0.74	0.73	0.83	0.86	0.77	0.73	0.87	0.78	0.74	0.83
a4	0.25	0.1	0.3	0.47	0.47	0.67	0.41	0.59	0.45	0.61	0.6	0.63	0.44	0.59	0.53	0.46
a5	0.27	0.33	0.43	0.31	0.11	0.63	0.54	0.37	0.32	0.62	0.68	0.5	0.43	0.45	0.49	0.65
a6	0	0.1	0.21	0.34	0.44	0.59	0.22	0.47	0.44	0.48	0.5	0.6	0.43	0.48	0.43	0.46
a7	0.15	0.19	0.21	0.62	0.38	0.56	0.39	0.51	0.36	0.52	0.4	0.64	0.47	0.38	0.59	0.71
a8	0.21	0.36	0.39	0.67	0.58	0.61	0.5	0.51	0.55	0.6	0.67	0.67	0.47	0.55	0.67	0.74
a9	0.15	0.06	0.26	0.67	0.72	0.88	0.4	0.5	0.56	0.62	0.74	0.75	0.74	0.82	0.9	0.92
a10	0.35	0.3	0.43	0.33	0.43	0.7	0.74	0.41	0.56	0.54	0.7	0.51	0.48	0.5	0.65	0.6
a11	0.52	0.36	0.46	0.31	0.5	0.73	0.63	0.71	0.68	0.76	0.7	0.65	0.75	0.65	0.45	0.73
a12	0.47	0.49	0.47	0.61	0.53	0.74	0.6	0.74	0.76	0.7	0.62	0.75	0.65	0.59	0.7	0.75
a13	0.85	0.53	0.24	0.35	0.16	0.84	0.78	0.84	0.84	0.73	0.84	0.84	0.9	0.78	0.72	0.39

Table 3.5. Detailed Results for Different Combinations of Sensors for the SLR

Class	S 1	S 2	S 3	S 4	S 5	All 5	S 123	S 124	S 125	S 134	S 135	S 145	S 234	S 235	S 245	S 345
a1	0.78	0.71	0.35	0.36	0.29	0.81	0.85	0.87	0.78	0.91	0.79	0.88	0.6	0.58	0.55	0.42
a2	0.56	0.28	0.3	0.26	0.36	0.68	0.62	0.68	0.6	0.74	0.69	0.53	0.5	0.49	0.42	0.35
a3	0.6	0.7	0.8	0.68	0.52	0.76	0.74	0.73	0.83	0.86	0.77	0.73	0.87	0.78	0.74	0.83
a4	0.25	0.1	0.3	0.47	0.47	0.67	0.41	0.59	0.45	0.61	0.6	0.63	0.44	0.59	0.53	0.46
a5	0.27	0.33	0.43	0.31	0.11	0.63	0.54	0.37	0.32	0.62	0.68	0.5	0.43	0.45	0.49	0.65
a6	0	0.1	0.21	0.34	0.44	0.59	0.22	0.47	0.44	0.48	0.5	0.6	0.43	0.48	0.43	0.46
a7	0.15	0.19	0.21	0.62	0.38	0.56	0.39	0.51	0.36	0.52	0.4	0.64	0.47	0.38	0.59	0.71
a8	0.21	0.36	0.39	0.67	0.58	0.61	0.5	0.51	0.55	0.6	0.67	0.67	0.47	0.55	0.67	0.74
a9	0.15	0.06	0.26	0.67	0.72	0.88	0.4	0.5	0.56	0.62	0.74	0.75	0.74	0.82	0.9	0.92
a10	0.35	0.3	0.43	0.33	0.43	0.7	0.74	0.41	0.56	0.54	0.7	0.51	0.48	0.5	0.65	0.6
a11	0.52	0.36	0.46	0.31	0.5	0.73	0.63	0.71	0.68	0.76	0.7	0.65	0.75	0.65	0.45	0.73
a12	0.47	0.49	0.47	0.61	0.53	0.74	0.6	0.74	0.76	0.7	0.62	0.75	0.65	0.59	0.7	0.75
a13	0.85	0.53	0.24	0.35	0.16	0.84	0.78	0.84	0.84	0.73	0.84	0.84	0.9	0.78	0.72	0.39

Table 3.6. Detailed Results for Different Combinations of Sensors for the SMO

Class	S 1	S 2	S 3	S 4	S 5	All 5	S 123	S 124	S 125	S 134	S 135	S 145	S 234	S 235	S 245	S 345
a1	0.73	0.7	0.19	0.24	0.11	0.81	0.74	0.77	0.81	0.65	0.64	0.65	0.59	0.49	0.53	0.3
a2	0.35	0.33	0.15	0.15	0.24	0.68	0.65	0.6	0.68	0.5	0.5	0.51	0.51	0.46	0.46	0.25
a3	0.46	0.83	0.76	0.78	0.34	0.76	0.81	0.84	0.87	0.9	0.7	0.8	0.92	0.76	0.81	0.81
a4	0.09	0.04	0.27	0.27	0.05	0.67	0.3	0.54	0.22	0.65	0.49	0.6	0.56	0.43	0.37	0.41
a5	0	0.09	0.21	0.38	0.16	0.63	0.36	0.38	0.24	0.44	0.36	0.48	0.42	0.42	0.41	0.5
a6	0	0	0.04	0.26	0.17	0.59	0.12	0.45	0.2	0.5	0.27	0.48	0.32	0.5	0.49	0.5
a7	0	0.05	0.05	0.48	0.34	0.56	0.17	0.62	0.39	0.51	0.49	0.67	0.51	0.46	0.59	0.64
a8	0.05	0.11	0.26	0.3	0.61	0.61	0.32	0.33	0.51	0.45	0.7	0.7	0.47	0.55	0.58	0.7
a9	0	0	0.05	0.5	0.67	0.88	0.07	0.56	0.6	0.5	0.54	0.67	0.52	0.63	0.68	0.82
a10	0.38	0.22	0.53	0.16	0.32	0.7	0.7	0.53	0.67	0.68	0.68	0.62	0.59	0.65	0.48	0.59
a11	0.6	0.46	0.15	0.2	0.23	0.73	0.65	0.71	0.77	0.81	0.75	0.74	0.62	0.65	0.44	0.55
a12	0.49	0.25	0.21	0.69	0.52	0.74	0.61	0.83	0.75	0.81	0.74	0.83	0.67	0.69	0.69	0.68
a13	0.9	0.75	0.06	0.17	0.12	0.84	0.92	0.9	0.9	0.73	0.88	0.9	0.87	0.89	0.83	0.23

3.6. References

[3.1] Y. Ren, R. W. N. Pazzi, and A. Boukerche, “Monitoring patients via a secure and mobile healthcare system,” *Wireless Communications*, IEEE, 17, no. 1, 2010, pp. 59-65.

[3.2] C. Doukas, T. Pliakas, and I. Maglogiannis, “Mobile healthcare information management utilizing Cloud Computing and Android OS,” In *32nd Annual International Conference of the IEEE Engineering in Medicine and Biology Society* (EMBC), 2010, pp. 1037-1040.

[3.3] L. Clifton, D. Clifton, M. Pimentel, P. Watkinson, and L. Tarassenko, “Predictive Monitoring of Mobile Patients by Combining Clinical Observations with Data from Wearable Sensors”, *IEEE Journal of Biomedical and Health Informatics*, 18, No. 3, 2014, pp. 722-730.

[3.4] O. Diallo, J. Rodrigues, M. Sene, and J. Niu, “Real-time query processing optimization for cloud-based wireless body area networks,” *Information Sciences*, 284, 2014, pp. 84-94.

[3.5] S. Pandey, W. Voorsluys, S. Niu, A. Khandoker, and R. Buyya, “An autonomic cloud environment for hosting ECG data analysis services,” *Future Generation Computer Systems*, 28, no. 1, 2012, pp. 147-154.

[3.6] M. Quwaider, and Y. Jararweh, “An efficient big data collection in Body Area Networks,” In *5th IEEE International Conference on Information and Communication Systems* (ICICS), 2014, pp. 1-6.

[3.7] N. Amini, M. Sarrafzadeh, A. Vahdatpour, and W. Xu, “Accelerometer-based on-body sensor localization for health and medical monitoring applications,” *Pervasive and mobile computing* 7, no. 6, 2011, pp. 746-760.

[3.8] A. Y. Yang, R. Jafari, S. S. Sastry, and R. Bajcsy, “Distributed recognition of human actions using wearable motion sensor networks,” *Journal of Ambient Intelligence and Smart Environments* 1, no. 2, 2009, pp. 103-115.

[3.9] W. Xu, M. Zhang, A. A. Sawchuk, and M. Sarrafzadeh, “Co-recognition of human activity and sensor location via compressed sensing in wearable body sensor networks,” In *9th*

IEEE International Conference on Wearable and Implantable Body Sensor Networks (BSN), 2012, pp. 124-129.

[3.10] M. Fraz, M. S. Sarfraz, “Text Detection in Still Images and CCTV Videos using LOacal Energy based Shape Histogram Features”, In *6th International Conference on Computer Vision Theory and Applications* (VISAPP 2011), Portugal, 5-7 March, 2011, pp. 433-436.

[3.11] J. Wan, C. Zou, S. Ullah, C.-F. Lai, M. Zhou, and X. Wang, “Cloud-enabled wireless body area networks for pervasive healthcare,” *IEEE Network*, 27, no. 5, 2013, pp. 56-61.

[3.12] A. Abbas and S. U. Khan, “A Review on the State-of-the-Art Privacy Preserving Approaches in E-Health Clouds,” *IEEE Journal of Biomedical and Health Informatics*, vol. 18, no. 4, pp. 1431-1441, 2014.

[3.13] A. Abbas, K. Bilal, L. Zhang, and S. U. Khan, “A cloud based health insurance plan recommendation system: A user centered approach”, *Future Generation Computer Systems*, 2014, DOI: DOI: 10.1016/j.future.2014.08.010.

[3.14] M. Ali, R. Dhamotharan, E. Khan, S. U. Khan, A. V. Vasilakos, K. Li, and A. Y. Zomaya, "SeDaSC: Secure Data Sharing in Clouds," *IEEE Systems Journal*. (Forthcoming.)

[3.15] M. S. Sarfraz, O. Hellwich, “Head pose estimation in face recognition across pose scenarios”, *International conference on computer vision theory and applications*, 2008.

[3.16] M. Jawad, M. Yasin, M. S. Sarfraz, “License Plate Detection using NMF with Sparseness Constraints through still Images,” *20th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision in cooperation with EUROGRAPHICS*, 2012, pp. 335-340.

[3.17] J. Daugman, “Complete Discrete 2-D Gabor Transforms by Neural Networks for Image Analysis and Compression,” *IEEE Trans on Acoustics, Speech, and Signal Processing*, Vol. 36. No. 7. July 1988, pp. 1169–1179

[3.18] M. C. Morrone, D. C. Burr, “Feature detection in human vision: A phase-dependent energy model,” *Proceedings of the Royal Society of London*, 1988, pp. 221-245.

[3.19] J. G. Liao, and K. –V. Chin, “Logistic regression for disease classification using microarray data: model selection in a large p and small n case,” *Bioinformatics* 23, no. 15, 2007, pp.1945-1951.

[3.20] S. Palaniappan, and R. Awang, “Intelligent heart disease prediction system using data mining techniques,” In *IEEE/ACS International Conference on In Computer Systems and Applications*, 2008, pp. 108-115.

[3.21] M. A. Labrador, and O. D. L. Yejas, Human Activity Recognition: Using Wearable Sensors and Smartphones, *CRC Press*, 2013

[3.22] C. Porcel, A. T. Lorente, M. A. Martínez, and E. H. Viedma, “A hybrid recommender system for the selective dissemination of research resources in a Technology Transfer office,” *Information Sciences* 184, no. 1, 2012, pp. 1-19.

[3.23] M. Zhang, and A. A. Sawchuk, “Human daily activity recognition with sparse representation using wearable sensors,” *IEEE Journal of Biomedical and Health Informatics*, 17, no. 3, 2013, pp. 553-560.

[3.24] A. Khan, Y.-K. Lee, S. Lee, and T.-S. Kim, “A triaxial accelerometer based physical-activity recognition via augmented-signal features and a hierarchical recognizer,” *IEEE Transactions on Information Technology in Biomedicine*, vol. 14, no. 5, pp. 1166–1172, Sep. 2010.

4. MACROSERV: A ROUTE RECOMMENDATION SERVICE FOR LARGE-SCALE EVACUATIONS

This paper[1] is submitted to *IEEE Transactions on Service Computing (TSC)* and is in the second round of review.

4.1. Introduction

Natural and man-made disasters, such as tsunamis, earthquakes, floods, and epidemics pose a significant threat to human societies. In response to the growing number of recent disasters, such as the Colorado flood, Oklahoma tornado, Japan's earth quake, Katrina hurricane, and in particular, the Red River crest that causes flood almost every year in Fargo, North Dakota, the importance and scope of emergency evacuation systems have grown tremendously over the past decade [4.1]. Well-planned evacuation operations and identification of appropriate rescue routes before and during a disaster play a significant role in saving lives and minimizing casualties.

4.1.1. Motivation

Generally, transportation planning departments consider the peak traffic demands during normal workdays and on special occasions [4.2], [4.3]. However, it is almost impossible to conceive transportation plans for emergency situations, due to which large

[1] The material in this chapter was co-authored by Muhammad Usman Shahid Khan, Osman Khalid, Ying Huang, Rajiv Ranjan, Fan Zhang, Junwei Cao, Bharadwaj Veeravalli, Samee U. Khan, Keqin Li, and Albert Zomaya. Muhammad Usman Shahid Khan had primary responsibility for conducting experiments and collecting results. Muhammad Usman Shahid Khan was the primary developer of the conclusions that are advanced here. Muhammad Usman Shahid Khan also drafted and revised all versions of this chapter.

volumes of traffic involved in mass evacuations is likely to exceed the capacity of road networks that may lead to loss of human lives. For example, due to the lack of proper evacuation plan, 25 people lost their lives in the first 30 minutes while attempting to flee their Oakland Hills neighborhood in California during a wildfire in the year 1991. Moreover, reports indicate that the inefficient evacuation planning in case of the Katrina and Rita hurricane resulted in a heavy traffic jam on the interstates. A similar traffic jam occurred for 20 hours after a winter storm in Atlanta, USA, in January 2014, as the transportation network was incapable of handling the traffic congestion caused by snow and accidents. To prevent such incidents, emergency evacuation plans must be developed to ensure the availability of safest and most efficient evacuation routes for the residents of a structure, region, or city.

The objective of this paper is to develop a scalable service that can guide evacuees towards safe and least congested routes during a disaster. With the integration of Intelligent Transportation System (ITS), the proposed MacroServ service is capable of computing the efficient traffic flows leading to minimum congestion of the roads during an emergency evacuation.

4.1.2. Research Problem

Several works, such as [4.4]–[4.7], have applied multi-objective optimization in evacuation modeling. Generally, optimization-based evacuation models consider several assumptions to optimize parameters, such as route length, shelter locations, and evacuation times. However, as discussed in [4.2], most of the times such assumptions are performance limiting or unrealistic, and do not precisely depict the dynamics of real-life emergency situations. Moreover, the following are the limitations of most of the optimization-based evacuation models that negatively affect the performance of such systems [4.2].

A few evacuation models simulate the traffic flow with static road network characteristics that do not truly depict the real emergency scenarios [4.5], [4.8]. For instance, numerous time varying behavioral, managerial, and stochastic factors, such as number of evacuees and traffic conditions, are involved during an evacuation [4.8]. Such factors may lead to congestion of the paths that were otherwise suggested as optimal by the evacuation modeling approaches.

If time factor is added to optimization problems, such that the static network is expanded over the planning horizon for every time interval, then the corresponding problem space becomes extremely large and there are no known polynomial algorithms for solving such problems [4.6].

Evacuation modeling in most of the optimization-based approaches is formulated as a network flow optimization problem [4.6], [4.9]. However, such approaches are not scalable for the real-world large-sized evacuation networks, due to the high computational complexities. Moreover, such problems are also considered to be NP-hard because of the multi-commodity nature, as evacuees are differentiated by the origin-destination pairs [4.7]. Therefore, solving for the travel demand rates and route flow rates requires simulation, as a closed form expression cannot be captured with optimization models [4.10].

As mentioned earlier, the optimization-based evacuation models consider assumptions for various parameters, such as road capacities, traffic volumes, route distances, and population sizes [4.2]. However, such assumptions can become invalid during a real emergency scenario due to variations in weather conditions, unforeseen conditions of traffic, and possible destruction of transportation infrastructure.

The immediate repercussion of the above listed limitations is the suboptimal performance of optimization-based evacuation models.

4.1.3. Methods and Contributions

To address the abovementioned limitations, in this paper we propose a scalable service, MacroServ, that is capable of performing real-time simulation of dynamic large-scale transportation networks during emergency scenarios. Simulation based evacuation planning by emergency management agencies require faster execution of large-scale vehicular traffic flows. Therefore, we utilize parallel computing to achieve the required scale, size, and speed of the computations. The MacroServ service integrates with the ITS to obtain real-time traffic data and utilizes our proposed algorithm to compute the maximum flow of routes and route costs among disaster sites and safety locations [4.11]. Based on the route costs, the MacroServ service redirects the traffic on alternate preferred routes before the congestion can occur. In this way, evacuees are guided towards the most preferred routes that have the minimum possible risk and the least amount of congestion.

Massive evacuations involve many stochastic factors, such as, degree of compliance of evacuees to evacuation calls, rate of evacuees departing from each household/ area, behavior of drivers, unforeseen traffic loads, and road conditions on transportation network. To depict such factors in our model, we make use of probability distributions, such as (a) Poisson distribution [4.12] and (b) Weibull distribution [4.12]. The aforementioned probability distributions allow us to model emergency evacuation scenarios that closely match with the realistic scenarios.

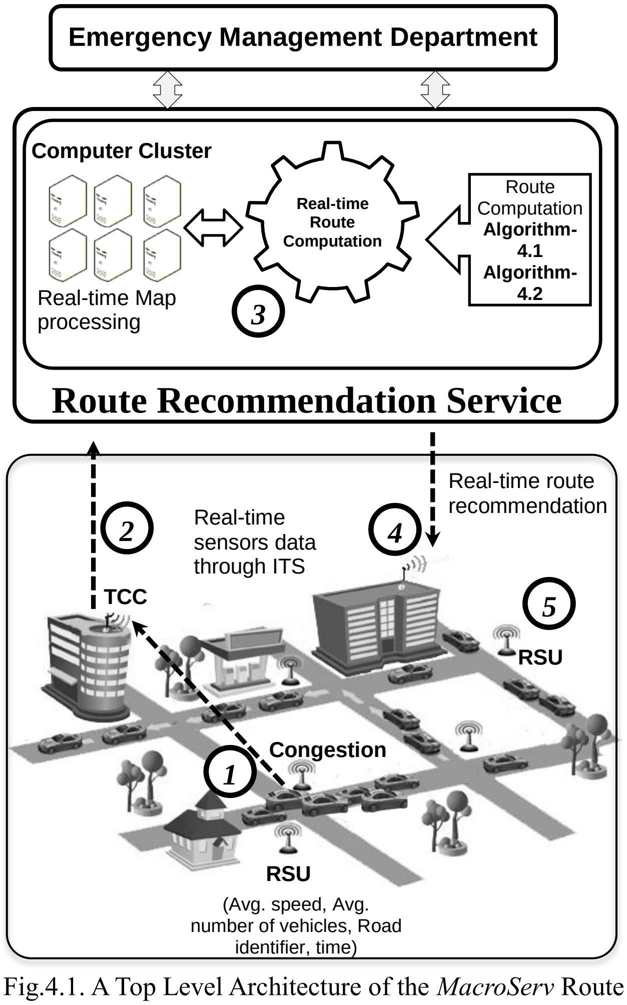
As a case study, we performed our simulations on the real map of City of Fargo, ND, USA where the Red River crest causes flood almost every year. The gradient (slope) of the Red River averages five inches per mile of length, and drops to 1.5 inches per mile in the region of Drayton-Pembina [4.13]. Due to lack of slope, the Red River tends to pool and cause floods. To model our system, we obtained the data, including road capacities, traffic volumes, speed limits, contours' elevations, historic crest levels of Red River, and historic flood affected areas, from the City of Fargo [4.14] and North Dakota Department of Transportation (NDOT) [4.15]. For our simulations, we considered the population of size 108,000 living at Red River flood zones that needs to be evacuated during a flood. Moreover, the transportation network consists of 7,370 road links and 2,800 intersections.

Our simulation results indicate that the traditional evacuation plans devised by the disaster management agencies are inefficient to handle sudden loads of traffic during an emergency. The sudden evacuations result in traffic jams due to which evacuation time increases. When the evacuees are directed towards the preferred routes using our MacroServ service, the overall evacuation time significantly decreases. Moreover, the simulation results indicate that the evacuation performance measures are largely dependent on the highway network structure and the number of vehicles produced in an emergency planning zone. In summary, the MacroServ service is designed to: (a) act as a decision making tool that will enable transportation departments to evaluate and review the emergency evacuation plans by simulating various disaster scenarios, and (b) recommend preferred and efficient routes to the evacuees during the course of a disaster by making use of high-end sensors and the ITS.

The remainder of this chapter is organized as follows. The MacroServ service architecture is described in Section 4.2. In Section 4.3, we discuss the importance and role of sensors in emergency scenarios. Section 4.4 presents the design and modeling of our approach. Experimentation results are discussed in Section 4.5. In Section 4.6, we discuss the related work, and Section 4.7 concludes the chapter.

4.2. Service Architecture

The MacroServ is designed as a route recommendation service that computes the preferred routes on a transportation network during the emergency evacuation scenarios. To make such service scalable, the parallel computing architecture is utilized on a cluster setup.



Emergency Management Department
Computer Cluster
Real-time Route Computation
Route Computation Algorithm-4.1 Algorithm-4.2
Real-time Map processing
3
Route Recommendation Service
Real-time route recommendation
2
Real-time sensors data through ITS
4
TCC
5
RSU
Congestion
1
RSU
(Avg. speed, Avg. number of vehicles, Road identifier, time)

Fig.4.1. A Top Level Architecture of the *MacroServ* Route Recommendation Service

We provide the details about various components of the service architecture with an example in the upcoming subsections.

4.2.1. Major Components

4.2.1.1. Road Side Units

As depicted in Fig. 4.1, the collection of traffic information is performed by Road Side Units (RSU). The RSUs consist of sensors deployed mostly on the intersections to capture the road characteristics and disaster related information, such as average speed of vehicles, average number of vehicles, rain intensity, flood level, and road’s extreme temperature conditions (See Section 3). The collected information is transferred to the route recommendation service, where the route computation takes place.

4.2.1.2. Route Recommendation Service

The basic purpose of route recommendation service is to perform the real-time analysis of the sensory data received through ITS and compute preferred routes for the evacuees that are least congested and at least risk. Fig. 4.1 depicts the top level components of the service, and the computational details, complexity, as well as empirical evaluation of the service are thoroughly investigated in the subsequent sections of the paper. The sensory data as input workload to the service consists of current traffic and disaster related information that is relayed to the route computation algorithm. The route computation algorithm running on cluster nodes computes a subset of routes that have the sufficient capacity to allow maximum traffic flow with minimum delay. The service relays the information about the computed set of preferred routes to the emergency management department to take appropriate decisions during evacuations, as well as to the evacuees for traffic guidance through RSU, navigation devices, and other means of communication, such as radio or smart phones.

4.2.2. An Example Scenario

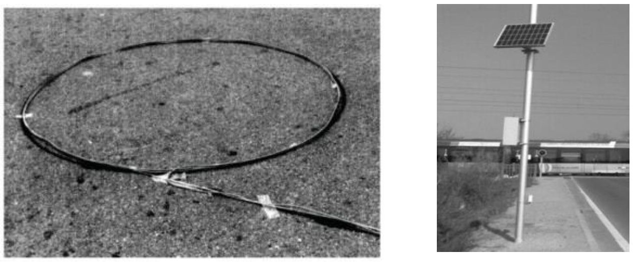
Fig. 4.1 also presents an example scenario of the proposed MacroServ service architecture. While the evacuation is in progress, the vehicles are following different routes on a city's transportation network. The road congestion information is communicated from the RSU sensors to the Traffic Control Center (TCC), as shown in Step 1 from where it is communicated to route recommendation service (Step 2). The route recommendation service utilizes computer cluster and route computation algorithm to compute the alternate routes with the maximum flow capacity (Step 3). The vehicles approaching towards congested road links are warned in advance, and are provided with alternate preferred routes on the navigation devices to prevent congestion (Step 4 and Step 5).

4.3. ITS and Disaster Management

An ITS is a combination of advanced sensing technologies used in transportation engineering to monitor traffic and road infrastructure, and to assist users for a better traffic management and safe travelling [4.16]. A basic ITS could include several essential components, such as: (a) a sensing system, (b) a communication system, (c) roadside units including traffic signal control system and movable signs, and (d) a notification system that includes car navigation and disaster warning system. In addition, advanced modeling techniques are also intensively used with a combination of an ITS for traffic predictions and guidance based on historical baseline data.

A sensing system plays a critical role and provides basis for any decisions made from the ITS. Achieving an accurate and reliable monitoring system for traffic and infrastructure behavior has attracted worldwide attention. Vehicle-based sensing systems are usually used for transportation infrastructure assessments while infrastructure-based systems can be applied for both traffic and infrastructure monitoring [4.19].

Infrastructure sensors can be installed either on the side or top of road, or can be embedded inside the roads also known as in-road reflectors [4.17], [4.18]. In the past decades, numerous infrastructure sensors were placed around cities and towns in United States, resulting in a network of ITS for better state and national traffic management.



(a) (b)

Fig.4.2. Traffic and Disaster Sensors

The collection of road traffic information can be achieved through in-road detectors. Inductive loop sensor, as shown in Fig. 4.2(a) is one example of the commonly applied infrastructure traffic sensor. Fig. 4.2(b) shows a typical road-side communication unit.

4.4. Design and Modeling

In this section, we present the design and model of the proposed route recommendation service. Due to a disaster's evolution in space and time, the network characteristics, such as vehicles' speed and road capacity vary with time. Some roads may suffer blockade over time due to congestion, whereas a few roads may become inaccessible after being hit by the disaster (e.g., flood). Capturing the important time evolving variations in the road infrastructure will only make the model more realistic.

We model the traffic at macro-scale level, where the vehicles act as intelligent agents carrying evacuees from sources to destinations. The MacroServ service computes real-time preferred routes for the evacuees. Consequently, the autonomous agents make dynamic route choices based on the congestion level on roads, distance from the destination, road safety condition, and capacity in safety shelters. The dynamic re-routing of vehicles more closely depicts a realistic scenario, as with the advancement in ITS and sensors, most of the vehicles nowadays are equipped with radio and GPS based navigation systems [4.11]. Therefore, the vehicles can interpret road conditions in advance by the help of navigation systems, as well as through the updates on radio. In the following subsections, we discuss the various phases of our proposed model.

4.4.1. Network Design

To create the transportation network, we imported the map of City of Fargo from the OpenStreetMap API [4.20] that has a database of world maps, and the regional maps can be

Intersections

Node ID	GPS point
N_1	P_1
N_2	P_2
...	...

Roads

Route ID	Start point	End point	Length	Speed limit	No. of lanes	Traffic volume
R1	P_1	P_{10}	1500	30	1	14,212
R2	P_7	P_{15}	800	35	2	21,200
...	...	...	...	...	...	...

Fig.4.3. Database of Transportation Network

exported in XML format. For each road segment (between two intersections), the information from the ITS about traffic volumes, speed limits, number of lanes, segment length, and contour elevations, has been stored in the database as indicated in Fig. 4.3. The aforementioned information has been obtained from NDOT [4.15]. Moreover, the intersections are also stored in database as nodes with unique identifiers (Fig. 4.3).

Based on the past records of flooding in the City of Fargo, the areas that are at higher risk of getting affected by the Red River flood are marked on the map as evacuation zones. Safety shelters are defined on the map locations that are not at risk of flooding.

4.4.2. Evacuee's Departure Rate

In this phase, we present a way of estimating the average number of vehicles departing from each home within a disaster affected area. Specifically, we need to find the time distribution of the evacuating vehicles. To make such estimations, it is important to know the population size of the particular area under consideration. However, the population size is a random factor that varies between day and night. People are more likely to be at work places during the day and at home during night time or weekends. In our model, we intend to introduce the maximum traffic load on the transportation network from the disaster affected area. Therefore, we assume that the people are at home when the disaster warning is announced. Moreover, we also assume that in a given time interval, vehicles originating from the homes make a discrete count, such as 0, 1, 2,…, or n number of vehicles. Therefore, to represent the vehicle departure rate, we utilized Poisson distribution [4.12] given as:

$$P[X = q] = \frac{\lambda^q \cdot e^{-\lambda}}{q!}. \tag{4.1}$$

The above equation indicates the probability that there would be q number of vehicle departures, where λ is the mean number of departures per time interval. The Poisson distribution is commonly used in queuing theory, and describes the probability that q events will occur within a time period, given that the time between two events is a random number that is independent of the time of the previous events [4.12]. For our given scenario, at a given time interval, some houses have no or a few vehicles coming out, most have some vehicles emerging, and a few houses have most vehicles departing. Moreover, the vehicle departures are also independent of each other. Under such circumstances, we considered the Poisson distribution to be more appropriate to depict the vehicular departures during an evacuation scenario. The primary reason for such selection is that just a single parameter, mean value λ, needs to be configured in simulations to evaluate the effect of varying number of departing vehicles during an evacuation depending on the number of residents at home.

4.4.3. Departure Times

The purpose of this phase is to model the rate of vehicles entering the transportation network following the evacuation instructions. For model accuracy, it is imperative to consider the evacuees' behavior in response to the evacuation orders. The evacuees' decision about when to leave depends on factors, such as, severity of the disaster, social status, and the availability of information and transport. Some people may prefer to stay behind to look after their property. In general, it is more likely that a few people leave initially, then the evacuations reach the peak value, and gradually slow down as most of the population have already evacuated. Such an evacuation behavior can be depicted with a response curve as stated in Fig. 4.4 that indicates the percentage of people departing in each time interval. The evacuation response curve can be expressed with probability density function of Weibull distribution [4.12], given as:

$$f(x; \alpha, \beta) = \begin{cases} \frac{\beta}{\alpha}\left(\frac{x}{\alpha}\right)^{\beta-1} \times e^{-\left(\frac{x}{\alpha}\right)^{\beta}}, & x \geq 0, \\ 0. & \text{otherwise} \end{cases} \tag{4.2}$$

In above equation, the parameter $\alpha > 0$ is the scale parameter, and $\beta > 0$ is the shape parameter of the distribution. If the parameter x represents the time taken in departure, then the Weibull gives a distribution that has departure rate proportional to a power of time. The values of α and β can be configured to analyze the impact of evacuees' compliance behavior during the emergency evacuations. Compared to the Poisson and Uniform distributions, the Weibull distribution more closely depicts the evacuee's behavior due to the asymmetric nature of the curve, as indicated in Fig. 4.4.

4.4.4. Safety Shelter Selection

This step computes the set of safety shelters for evacuating vehicles. The simplest approach is to assign safety shelters that are at the shortest distances from the vehicles.

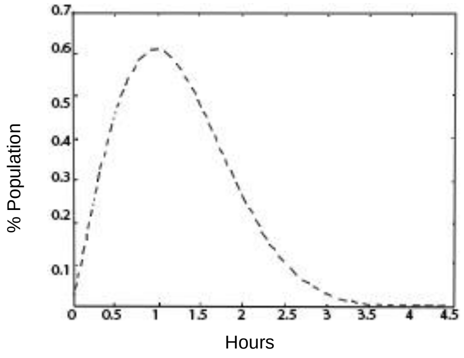


Fig.4.4. Evacuees Departure Times in Response to Emergency

However, this may result in overcrowding of shelters when most of the vehicles prefer to reach the nearest shelters. An alternate approach is to manually designate shelters for the evacuees in various areas. However, the aforementioned approach is not efficient, as the roads leading to manually designated shelters may become inaccessible due to congestion or other factors.

In our model, we adopt a probabilistic approach of assigning shelters to evacuees. We consider the real-time varying factors, such as road traffic, congestion, distance, risk level, and capacity of the shelter. In this way, evacuees are recommended a set of shelters that are most preferable in the current time interval.

4.4.5. Route Selection

In this step, the route recommendation service computes the preferred routes for the vehicles during evacuation. The service allows individuals to take decisions about route selection at road intersections. For an intersection, the service maintains a route table that contains least travel costs from the intersection to each of the destinations. Cost is based on flow capacity, maximum speed limit, density of traffic, length, and travel time of the road link. The total cost of a route between an intersection and the destination shelter is the sum of the costs of individual road segments. As a disaster has tendency to expand outwards from the epicenter, such as in the case of wild fire, tornado, and floods, the roads which are not affected by disaster yet may get hit by the disaster at a later time. Therefore, costs are computed by also taking into consideration the disaster affected roads. The information about road damage is provided at real-time by the high-end ITS sensors. As indicated in Table 4.1, if a vehicle is heading for Shelter 2, then at the intersection, the least cost towards Shelter 2 is 13 that is for the route 3. Moreover, the costs indicated in Table 4.1 are dynamic

Table 4.1. Route Cost Table Maintained by Each Intersection

Route ID	Shelter 1	Shelter 2	Shelter 3	Shelter 4	…
1	11	65	32	35	…
2	43	34	54	31	…
3	23	13	31	33	…
4	24	21	45	36	…
…	…	…	…	…	…

Table 4.2. Road Congestion Example

$\boldsymbol{T_1}$	$\boldsymbol{T_2}$	$\boldsymbol{T_3}$
$F_k = 4 \times 50 = 200$ $S_k = 200/4 = 50$ $C_k = 300/50 = 6$	$F_k = 4 \times 50 = 200$ $S_k = 200/5 = 40$ $C_k = 300/40$ $= 7.5$	$F_k = 4 \times 50 = 200$ $S_k = 200/7 = 28.5$ $C_k = 300/28.5$ $= 10.5$

Table 4.3. Notations and Description

G	Graph representing the transportation network
$H^{(r)}$	Sub-graph, representing a region r
X_{xz}	Set of routes between an intersection x and destination z
g	A group of people
l^m	Location of person m
Z	Set of shelters

and are adjusted at real-time according to the road congestions. Therefore, when more vehicles enter a road segment, the overall speed of the vehicles decrease and the cost is recalculated. In the following subsections, we present the cost calculation and route recommendation algorithm.

4.4.5.1. Route Cost Calculation

The route cost is the time it will take a vehicle to traverse a route to reach the destination shelter. When traffic capacity of a route is greater, then the vehicles will take lesser time in traversing the route, and smaller will be the route cost. Let L_k be the length of a road segment k, which has N_k number of lanes, and ℓ be the average length of vehicles passing through the road segment. The maximum possible traffic flow capacity of the road segment k is given as:

$$\mathrm{f_k} = \frac{\mathrm{L_k} \times \mathrm{N_k}}{\ell}. \tag{4.3}$$

Equation (4.3) computes the maximum number of vehicles that can traverse the road segment without congestion at a given time interval. In normal situations, inter-vehicular distance depends on speed of vehicles. More the speed, higher would be the inter-vehicular distance. However, we assume that during the time of disaster, it is almost impossible for the vehicles to maintain appropriate inter-vehicular distance, because, evacuees are trying to save their lives, so traffic laws are difficult to apply. Suppose, the number of vehicles currently travelling through the road segment k is denoted as τ_k. Then, the extent of congestion experienced by the road segment k is given by:

$$\mathrm{c_k} = \frac{\tau_\mathrm{k}}{\mathrm{f_k}}. \tag{4.4}$$

If we let $\mathrm{s_k^{max}}$ to denote the maximum allowed speed limit of the road segment k, then the free flow ($\mathrm{F_k}$) of vehicles currently traversing the road segment is given as:

$$\mathrm{F_k} = \mathrm{f_k} \times \mathrm{s_k^{max}}. \tag{4.5}$$

With τ_k number of vehicles travelling through the link, the maximum possible speed can be computed as:

$$\mathrm{S_k} = \frac{\mathrm{F_k}}{\tau_\mathrm{k}}. \tag{4.6}$$

Finally, we compute the travel cost of road segment k given by:

$$\mathrm{C_k} = \frac{\mathrm{L_k}}{\mathrm{S_k}}. \tag{4.7}$$

The cost in the above equation is dynamic and varies with time as the numbers of vehicles on the road segment vary. An Illustrative Example: We present an example of the cost computation with the support of Table 4.2 and Fig. 4.5. Suppose, at the time interval $\mathrm{T_1}$, the road segment CD has maximum capacity of four vehicles ($\mathrm{f_{CD}} = 4$) and the maximum speed limit $\mathrm{s_{CD}^{max}} = 50$ km/h. The free flow of vehicles according to (5) is given as $\mathrm{F_{CD}} = 200$. As indicated in Fig. 4.5(a), the current number of vehicles on link is $\tau_{\mathrm{CD}} = 4$, so by using (6), we get the maximum possible speed as $\mathrm{S_{CD}} = 200/4 = 50$. If the length of the link is $\mathrm{L_{CD}} = 300\mathrm{m}$, then the travel time cost is given as $\mathrm{C_{CD}} = 300/50 = 6$ (Table 4.2, Column 1). As one more vehicles enter the link CD, the total number of vehicles becomes $\tau_{\mathrm{CD}} = 5$ (Fig. 4.5(b)), and the travel cost is increased to 7.5. In Fig. 4.5(c), when the number of vehicles on link CD reaches 7, then the link cost becomes 10.5 (Table 4.2, Column 3), which is an indication of road congestion, and consequently the vehicles are travelling with reduced speeds. Therefore, the incoming vehicles are redirected to other routes with the smaller cost (indicated by arrows in Fig. 4.5(c). In the following subsection, we present our algorithm that computes the preferred routes for the evacuating vehicles.

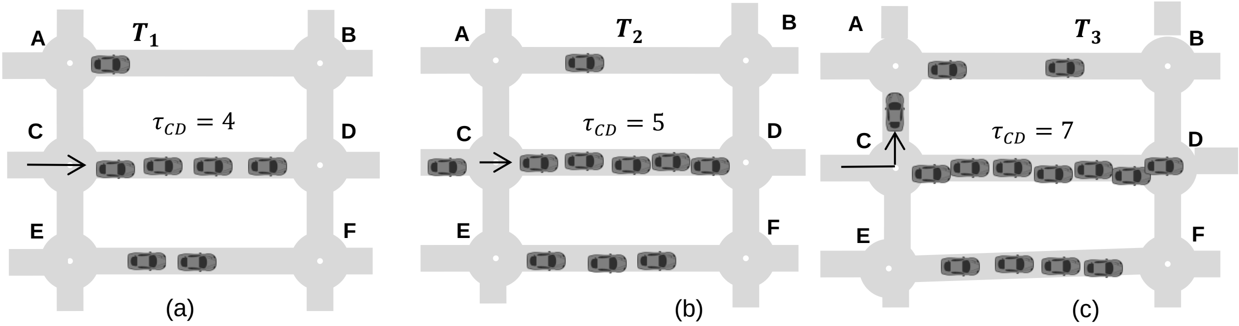


Fig.4.5. Route Cost Update: (a) No Congestion, (b) Slight Congestion, and (c) Alternate Route Selection Due to High Congestion. The Arrow Sign Shows the Directions to the Next Intersection During Evacuation at Intersection *C*.

4.4.5.2. Route Computation Algorithm

Fig. 4.6 depicts a portion of the map of City of Fargo that we considered as a case study in this paper. The three main items on the map are disaster risk areas, intersections, and safety shelters. During the evacuation, the residents in affected areas flee towards safety shelters, and are guided about the routes on intersections. We denote the road network with a graph notation: $\mathrm{G} = (\mathrm{V}, \mathrm{E})$, where V is the set of vertices representing intersections, and E is the set of links that represent roads. Table 4.3 indicates the set of notations used in this subsection. The real-time processing of graph of up to the scale of a city is very resource intensive task and is not feasible to be performed by a single computational node. Therefore, the graph G is logically split into several sub-graphs, and each sub-graph is processed on a separate node using MPI on the HPC cluster. We denote a sub-graph by $\mathrm{H}^{(\mathrm{r})}$, $\mathrm{r} = 1, 2, 3, \dots, \mathrm{n}$, such that $\mathrm{H}^{(1)} \cup \mathrm{H}^{(2)} \cup \mathrm{H}^{(3)} \cup \dots \cup \mathrm{H}^{(\mathrm{n})} = \mathrm{G}$. Between any two regions $\mathrm{H}^{(\mathrm{a})}$ and $\mathrm{H}^{(\mathrm{b})}$, we define a set of overlapping points as boundary points B_{ab}. The boundary points are the intersection nodes that are common to both the regions, such that $\mathrm{B}_{\mathrm{ab}} = \mathrm{H}^{(\mathrm{a})} \cap \mathrm{H}^{(\mathrm{b})}$. Suppose, evacuees in region $\mathrm{H}^{(\mathrm{a})}$ are recommended a shelter that is located in the region $\mathrm{H}^{(\mathrm{r})}$, where $\mathrm{r} \neq \mathrm{a}$. As a first step, the evacuees are routed towards a boundary point $\mathrm{p}_{\mathrm{i}} \in$

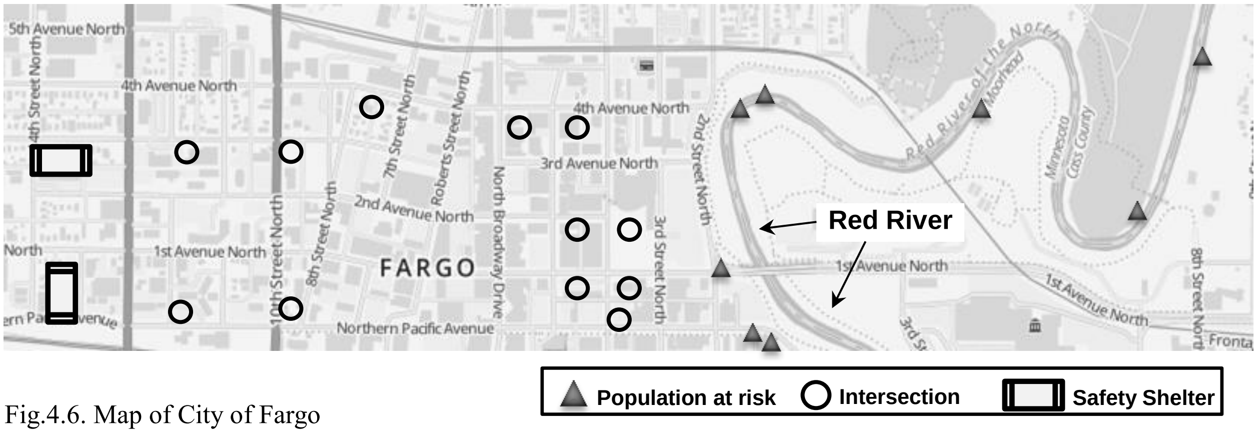


Fig.4.6. Map of City of Fargo

Algorithm 4.1. Route Recommendation

```
1:  while time interval $t \leq T_{end}$ do
2:    PARFOR region $H^{(r)} \in G$ do
3:      PARFOR intersection $x \in H^{(r)}$ do
4:        for each shelter $z \in Z$ do
5:          If z $\in H^{(r)}$ then
6:            Route cost $\xi^{(xz)} = min\{\sum_{e\in j} C_e\}$ , $\forall j \in X_{xz}$
7:          else
8:            Route cost $\xi^{(xp_i)} = min\{\sum_{e\in j} C_e\}$ , $\forall j \in X_{xp_i}$
9:          end if
10:       end for
11:     end PARFOR
12:     for each $g$ in area $A \in H^{(r)}$ do
13:       $D = GetDestination(g)$
14:       for each $m \in g$ do
15:         $move(m, D)$
16:       end for
17:     end for
18:   end PARFOR
19: end while
```

$\{\mathrm{B}_{\mathrm{ak}} = \mathrm{H}^{(\mathrm{a})} \cap \mathrm{H}^{(\mathrm{k})}\}$ that has the minimum congestion and risk at the current time interval (determined through ITS). Here, $\mathrm{H}^{(\mathrm{k})}$ is the region adjacent to $\mathrm{H}^{(\mathrm{a})}$, and located on the shortest and safest path towards the target shelter. On reaching the boundary point p_{i} using least cost paths, as the next step, vehicles are routed within the same region $\mathrm{H}^{(\mathrm{k})}$ based on route cost computations. If k=r, then the vehicles have reached the desired region, where the target shelter is located. Otherwise, the aforementioned procedure will be repeated to further route the vehicles towards new boundary point.

As indicated in Algorithm 4.1, the PARFOR loop executes in parallel for each of the regions (Line 2). Within a region, route cost tables are updated in parallel using (7) for every intersection and shelter (Line 3–Line 11). There can be more than one route possible between an intersection and a safety shelter, and each route consists of numerous road segments. If the destination shelter lies within the same region, then the Line 6 filters out a route that has the minimum travel cost between an intersection x and shelter z. Otherwise, the minimum travel cost is calculated between the shelter x and the boundary point p_i as indicated in Line 8. For various groups of evacuees in the disaster affected area (A), the preferred destination shelters are selected using GetDestination function in Line 13 that is defined in Algorithm 4.2. Algorithm 4.2 selects only those shelters from the set of shelters that still have space to accommodate more people (Line 2). We assume that shelter space information is available by the help of sensors installed at each shelter. The Line 3 to Line 7 of Algorithm 4.2 computes the minimum travel cost of each evacuee in the group g from each shelter. The shelter whose average travel cost is minimum from all the group members is considered as the one satisfying the group members and is selected as the destination shelter (Algorithm 4.2, Line 8 to Line 10). An example scenario of the aforementioned destination selection is when members of a family are at different locations in a city during the time of disaster, and they want to gather at a place that is at shortest travel costs for each member. On selection of the destination shelter, the Line 14 to Line 16 of the Algorithm 4.1 moves each group member towards destination shelter. More precisely, in the real emergency scenario, the move function is meant to recommend the evacuees with least cost route towards the destination. The move function also handles the boundary condition, such that when a vehicle reaches the boundary of a region $H^{(r)}$, the current region hands over the vehicular details to the new region. Such a case may arise when the destination shelter is not located in the same region from where the vehicle has initiated evacuation.

4.4.5.2.1. Time Complexity

The time complexity of the MacroServ is based on the time complexity of the Algorithm 4.1. From the Line 4 to Line 9, the Algorithm 4.1 calculates the cost of all the routes from a single intersection to all of the shelters and boundary points. For z shelters and j routes from an intersection to a shelter, the time complexity is O(z×j) and for p boundary points, the time complexity is O(p×j). The time complexity of the Algorithm 4.1 from the Line 4 to Line 9 is O(z×j+p×j). The number of boundary points p is much larger than the number of shelters. Therefore, the time complexity is equivalent to O(p×j). For i intersections, the time complexity is O(i×p×j). For h regions the time complexity is O(h×i×p×j). In Line 13, the destination is selected for each group by calling the Algorithm 4.2.

In Line 1 of the Algorithm 4.2, all the members of the group are retrieved with time complexity O(1). All the shelters having space are selected in the Line 2 that takes O(z). The time complexity of the Algorithm 4.2 from Line 3 to 7 is O(m×z), where m is the number of the users in the group. The time complexity of the Line 8 is O(m). Therefore, the total time complexity of the Algorithm 4.2 is O(z+m×z+m+1) that is equivalent to O(m×z).

The Line 13 in the Algorithm 4.1 uses Algorithm 4.2 so its time complexity is O(m×z). For g number of groups, the time complexity increases to O(g×m×z). The time

Algorithm 4.2. Get Destination Shelter

Input: A group g of people

Output: A shelter $\mathcal{S}$ that is preferred by every member of group

```
1:  $Q \leftarrow$ Retrieve group members from $g$
2:  $V' \leftarrow getShelters(\,)$ // shelters that have space
3:  for each member $m \in Q$ do
4:      for each shelter $v \in V'$ do
5:          $P[m][v] = \min c^{(j)}(l^m, v), \forall j \in X_{l^m,\ v}$
6:      end for
7:  end for
8:  $Rank[v] \leftarrow avg(P)$
9:  $\mathcal{S} = v_{min(Rank)}$
10: return $\mathcal{S}$
```

complexity of the Line 14 to Line 16 is O(m) that increases to O(g×m) for g number of groups. For Line 2 to Line 18, the time complexity becomes O(h×(i×p×j+g×m×z)). The algorithm iterates Tend times. Therefore, the total time complexity of the Algorithm 4.1 is O(Tend×h×(i×p×j+g× m×z)). By executing the algorithm in parallel, the time complexity of the Algorithm 4.1 is reduced to O(Tend×(p×j +g×m×z)).

4.5. Performance Evaluation

In this section, we evaluate the performance of the proposed MacroServ service. We implemented the route cost computation algorithm in C++ using OpenMPI library [4.21]. The experiments are performed on HPC cluster established in North Dakota State University (NDSU), Fargo, ND, USA [4.22]. The cluster nodes have the following specifications: quad-core Intel X5550 @ 2.67GHz with 48GB ECC DDR3 1333MHz (8GB DIMMs), 160GB 7.2K RPM SATA HDD, 1x Myri-10G port, and dual Gigabit Ethernet ports. Fig. 4.6 indicates the scenario considered in the simulations. The population settled at the Red River's bank needs to be evacuated towards the safety shelter locations. The road intersections are equipped with ITS sensors to send alerts to the vehicles during the evacuations. The map is divided into 3 regions (zones). The total number of evacuees is about 108,000. If the given map is converted to a graph representation, then the graph has 2,800 vertices with 7,370 edges.

4.5.1. Performance Metrics

The performance metrics considered in the simulations include: (a) evacuation times, (b) average travel time, (c) road congestion, and (d) population evacuated. The evacuation time indicates the time spent between the start of evacuation and when the last person evacuates the affected area. The average travel time computes the average of travel times of all the evacuees. The road congestion is computed by (4) and indicates the amount of congestion experienced by a road segment at a given time interval. The population evacuated indicates the number of people who have fled from disaster area in a certain time interval. The performance of the system is observed by varying the parameters of the Poisson and Weibull distributions.

4.5.2. Comparison Techniques

To compare the performance of the MacroServ service, we considered two other evacuation approaches: (a) Dedicated and (b) Shortest-Path. In the first approach, the evacuating population follows only those routes that are predefined by the Fargo department of transportation [4.15]. The criterion of selection of dedicated routes is set by department of transportation, and the primary factor is the road capacity. Therefore, we considered the dedicated routes as a set of the interstates and main roads [4.14]. We developed the shortest-path model (based on Dijkstra algorithm) that allows the evacuees to take the shortest routes from disaster site towards safety locations. The information about the changes in the road

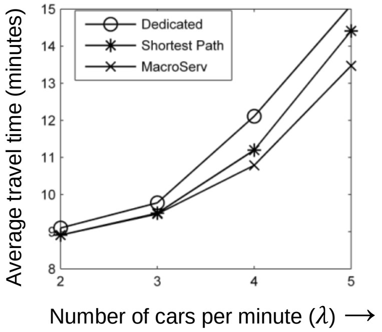


Fig.4.7. Average Travel Times with Varying Number of Departing Vehicles from Each Intersection (α=6 and β=2)

conditions is made available to both the aforementioned baseline approaches to provide a fair comparison with the proposed MacroServ.

4.5.3. Evaluation Results

In this subsection, we present the evacuation performance of our proposed scheme in comparison to the abovementioned approaches. For each data point, the simulation is repeated 20 times to obtain the statistical significance of the results. In our experiments, we also introduced the damage to the roads by the disaster to analyze the impact on total evacuation and average travel time. For that purpose, we utilized Bernoulli distribution [4.12] to randomly mark roads as damaged beginning with destroying initially those roads that are geographically near to the disaster site, and as the simulation time proceeds, the road destruction is expanded outwards to mimic damage caused by the floods.

4.5.3.1. Impact of Departure Rate

Fig. 4.7 depicts average travel time of the vehicles by varying the departure rate of vehicles from the intersections. When the departure rate is lower, less than 3 vehicles per minute, the average travel time taken is almost same for all the three approaches. However, as the departure rate increases, the average travel time of Dedicated and Shortest-Path turns

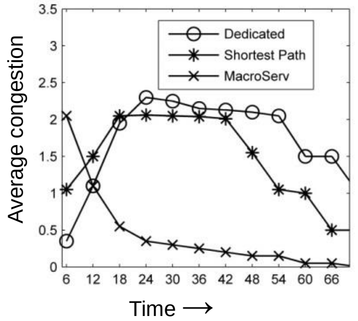
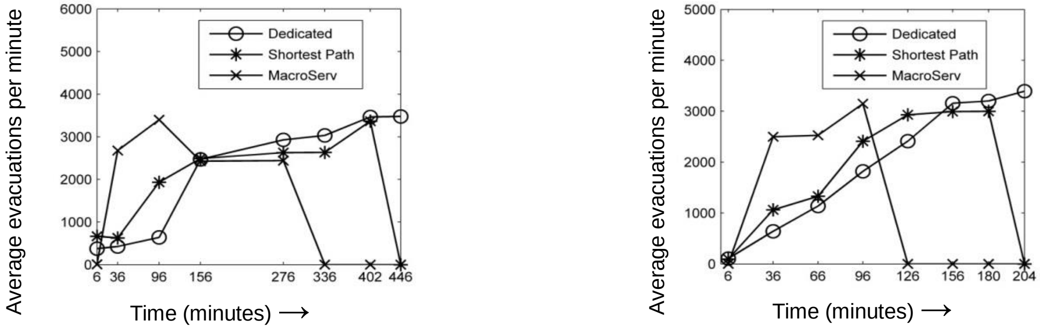


Fig.4.8. Average Congestion with Respect to Time with Damaged Road Network α=6, β=2, and λ=5

out to be higher than the MacroServ. This is an expected outcome, as the aforementioned approaches do not take into account the current traffic flow rate on roads. As people tend to adopt the shortest route for evacuations, the roads become congested and road's traffic flow rate is dropped. Same is the reason for Dedicated approach as arrival of too many vehicles on limited set of roads results in congestion of such roads.

4.5.3.2. Impact of Congestion

Fig. 4.8 compares the three approaches for congestion on the roads near the shelters with respect to time. Initially, due to heavy congestion in case of Dedicated and Shortest-Path, very few cars are able to reach the roads leading towards shelters. Therefore, the congestion on the roads is shown lesser for Dedicated and Shortest-Path in the Fig. 4.8.



(a)

(b)

Fig.4.9. Average Evacuations per Minute with Varying Shape Parameter in Weibull distribution: (a) α=6, β=1, and λ=5, and (b) α=6, β=2, and λ=5

Alternatively, the vehicles quickly reached the roads near shelters for the MacroServ scheme, which resulted in higher congestion in the first few minutes. However, the congestion decreases subsequently as most of the vehicles have reached the shelters. Moreover, congestion level decreases relatively at lower rate in the case of Dedicated and Shortest-Path. This is due to the fact that both the approaches do not consider traffic flow capacity while computing routes, and as a result evacuees are guided towards short but congested roads.

4.5.3.3. Impact of Shape (β) of Weibull Distribution

In this simulation, we inspect the impact of evacuations with respect to time. The β=1 in Fig. 4.9(a) indicates that most of the people have departed immediately after disaster warning from the affected site. The sudden significant increase in departing population resulted in an overall increase in the evacuation times. The β=2 (Fig. 4.9(b)) is the approximate mean around the scaling parameter, which creates the similar curve indicated in Fig. 4.4. As reflected from Fig. 4.9(b), the total evacuation time with β=2 is lesser than the evacuation time with β=1 in Fig. 4.9(a). The most probable reason is that with β=2, few

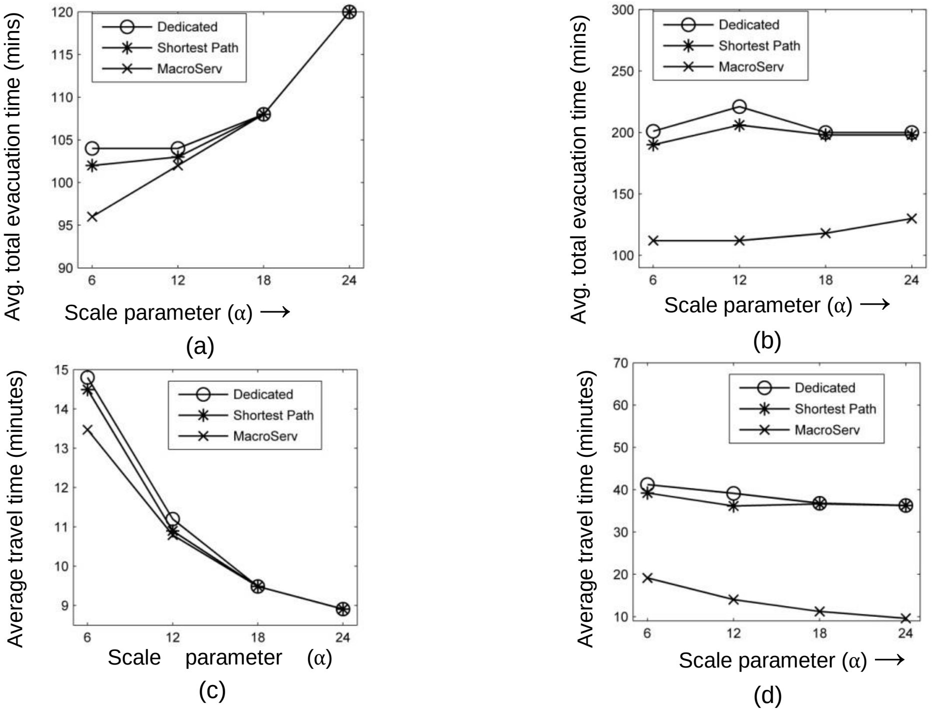


Fig. 4.10. Effect of Road Damage by Varying Departure Time (Scale Parameter α) with β=2: (a) Average Total Evacuations under Normal Conditions, (b) Average Total Evacuations under Damaged Conditions, (c) Average Travel Time under Normal Condition, and (d) Average Travel Time under Damaged Conditions.

people are expected to be evacuated initially, then evacuations go to a peak value and gradually decrease, which results in the less congestion of roads. The slow departure puts less load on the road network and results in lesser evacuation time. Fig. 4.9 indicates that an organized evacuation with gradual departure performs much better than a random immediate evacuation approach. However, whether the evacuation is organized or not, the MacroServ scheme yields shorter evacuation time and evacuates more population per minute as compared to the other approaches. Especially, within one hour of issuing evacuation comments, the MacroServ service doubles the amount of evacuees when compared with the other two approaches. This accounts for the fact that the evacuees are directed towards the roads with maximum flow rate.

4.5.3.4. Impact of Average Departure Time

Fig. 4.10 presents the results for varying the departure time that is determined by the parameter α. As it can be seen in Fig. 4.10(a), under normal road conditions, when the average departure time of the evacuees increases, the average total evacuation times become similar for MacroServ, Dedicated, and Shortest-path. The similarity in evacuation times of the three approaches is due of the low density of traffic on roads. As inter-arrival times of vehicles entering road network have increased, this leads to the lesser congestion on roads. Moreover, increased departure time also results into the increased evacuation times. Fig. 4.10(b) depicts the evacuation times under damaged road conditions. The MacroServ takes half of the time for evacuations as compared to the time taken by the other approaches. Alternatively, Shortest-Path does not consider the road condition and congestion while computing the routes. This may lead to traffic jams when vehicles move towards the damaged roads, resulting in increased congestion and total evacuation times. Fig. 4.10(c) shows the average travel time versus departure time, α. The average travel time decreases as the departure time increases under normal road conditions. Similar to Fig. 4.9, it can be seen from Fig. 4.10(c) that as people are taking more time in departure, there would be less congestion on the roads, and average travel times for the three approaches becomes similar. Under damaged road network (Fig. 4.10(d)), MacroServ outperforms the other two schemes because of the most preferred route recommendation strategy.

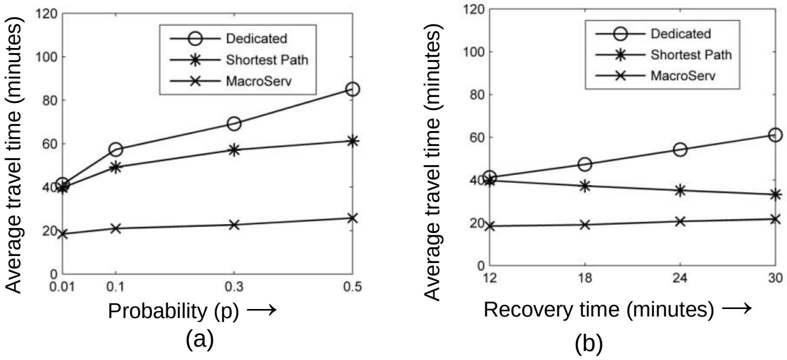


Fig.4.11. Impact of (a) Road Damage Probability and (b) Recovery Time, on Average Travel Time with α=6, β=2, λ=5, and p=0.01

4.5.3.5. Impact of Road Damage Probability

The simulations are performed to analyze the effect of road infrastructure damage on the three evacuation schemes. Fig. 4.11(a) indicates that with the increase in road damage probability, the average travel time also increases. However, the MacroServ scheme has lesser increase in travel time as compared to the other two approaches. The reason for the aforementioned behavior is that the evacuees are not directed towards damaged routes in the case of MacroServ, and consequently, the vehicle speeds are not reduced. Fig. 4.11(b) depicts the effect of damage recovery time of roads on the average travel time. When damage to the road network occurs, the vehicles are diverted towards paths that are slightly longer than the damaged ones. When the recovery time is smaller, such as 12 minutes, more vehicles arrive back on the shortest paths and cause congestion that causes more delay. With increase in the recovery time (18 to 30 minutes), more cars were diverted to longer paths, and few remaining cars came back on the shortest path, and traveled with less congestion. Overall average travel time decreases slightly in case of Shortest-Path. The MacroServ approach is already using the road network efficiently and is not allowing congestion on the road, so effect of road damage is not observed in MacroServ.

4.5.3.6. Impact of Population Growth

In this simulation run, we evaluate the effect of damaged road network if the population is increased by 2% in each coming year [4.23]. Fig. 4.12 indicates that there would be very slight effect of increase in population on the average travel time for all the approaches. However, in contrast to the Dedicated and Shortest-Path the proposed MacroServ evacuation strategy is exhibiting the least average travel times in the future years. Therefore, we may conclude that MacroServ is capable of efficiently handling the evacuations with increased population size.

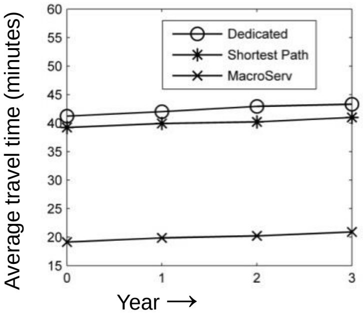


Fig.4.12. Effect of Population Increase in Future 3 Years on Average Car Travel Time on Damaged Network (α=6 β=2, and λ=5)

4.5.3.7. Scalability Analysis

The simulations are performed to analyze the scalability of the MacroServ framework. An algorithm is known to be scalable if the algorithm can maintain the execution time in desirable limits even in case of large increase in workload by using additional processors.

We split the map into smaller regions using two different techniques, Quadtree [4.32] and METIS [4.33]. Quadtree is a data structure technique most often used to partition a two-dimensional space by recursively subdividing it into four quadrants or regions. The procedure of creating the Quadtree-based partitioning begins with decomposing the region into four equal quadrants, subquadrants, and so on with each leaf node containing data corresponding to a specific sub-region based on a criteria. Each node in the tree either has exactly four children, or has no children (a leaf node) [4.32]. METIS [4.33] can partition an unstructured graph into a user-specified number k of partitions. We utilized METIS's multilevel k-way partitioning algorithm [4.33] because it provides additional capabilities, such as minimize the resulting subdomain connectivity graph, enforce contiguous partitions, and minimize alternative objectives.

Table 4.4. Quadtree vs METIS

Technique		Number of partitions			
		1	4	8	16
Quadtree	Recommendation generation time (secs)	91.17	30.951	21.984	15.344
	Number of messages	0	297	1031	2129
METIS	Recommendation generation time (secs)	91.17	29.394	23.712	15.828
	Number of messages	0	203	1049	2141

We varied the number of partitions in our simulations and observed the effects of such variation on parameters, such as recommendation generation time and inter-message exchange among the computational nodes. Both the aforementioned parameters are most significant as they determine the efficiency of the proposed framework. Table 4.4 presents the results of recommendation generation time and number of messages for both METIS and Quadtree. We considered the maximum of 16 partitions, as increasing the number of partitions results in some partitions becoming empty (carrying no nodes). An important observation related to partitioning was that the numbers of nodes were same in each of the sub-regions created by METIS, whereas they were different in the sub-regions resulted from Quadtree. Such observation had also an effect on the results. With lesser number of partitions and large sized sub-regions, both the average recommendation generation time and average number of messages passed were less in case of METIS as compared to Quadtree. However, as the number of partitions (each assigned to a processor) increases, the results of Quadtree and METIS became almost similar. Therefore, due to similarity in results for larger number of partitions, we preferred the Quadtree for further experimentation as Quadtree exhibits simplicity in terms of data handling and querying the partition from the map.

In Fig. 4.13 and Fig. 4.14, we evaluated the aforementioned metrics for our proposed scheme MacroServ by varying the number of vertices in the map. We utilized Quadtree for map partitioning.

Fig. 4.13 presents the effect of increasing the number of partitions (one partition per processor) as well as the size of map on the recommendation generation time. Results indicate that doubling the size of the region increases the recommendation generation time by an average of 26%. The increase in single processor results in decrease in the recommendation generation time by an average of 9%. Moreover, it can be observed from the results that by increasing the number of processors, the MacroServ framework can efficiently provide recommendations for the large-scale datasets with little effect on the recommendation generation time.

Fig. 4.14 shows the effect of increasing the number of partitions and map size on the average number of vehicles that travel from one zone to another (i.e., number of messages passed between processors). The results indicate that the increase in a single processor increases the average number of vehicles travelling from one zone to another by 39%. However, doubling the size of the map decreases the average number of vehicle crossing from one zone to another by 76%.

Based on the results presented in Table 4.4, Fig. 4.13, and Fig. 4.14, it can concluded that the MacroServ is a scalable service, as it can efficiently handle the large sized map by increasing the number of processors.

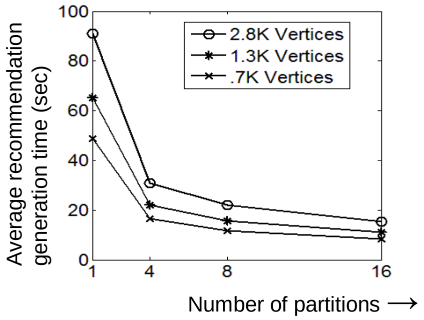


Fig.4.13. Effect of Increase in Number of Partitions on Average Recommendations Generation Time

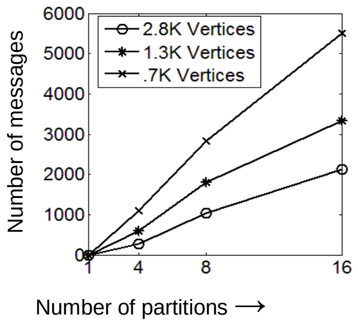


Fig.4.14. Effect of Increase in Number of Partitions on Number of Messages

4.6. Related Work

Numerous studies conducted in the past addressed various perspectives of emergency evacuation modeling, such as route finding [4.6], shelter site selection [4.5], evacuees' behavior [4.10], and traffic control strategies [4.24]. In recent years, there has been a growing interest in the multi-objective optimization techniques for evacuation route finding problem.

The authors in [4.25] studied demand-based strategies for aggregate-level routing with and without congestion. The authors proposed a network flow model that optimized an evacuation specific lexicographic objective function. The function computes the time dependent evacuation routes for each of the source. However, being a combinatorial optimization problem, the proposed approach is difficult to be solved for large realistic networks. Therefore, the authors utilized two heuristics to solve the problem, but with a tradeoff of solution quality. Lim *et al*. [4.6] modeled evacuation problem as network flow optimization problem. The static network is expanded over the planning horizon for each time interval. However, this makes the optimization problem extremely large to solve. Therefore, the authors proposed a heuristic based solution that utilized Dijkstra's algorithm to compute evacuation paths, and a greedy algorithm to find the maximum flow and exit schedule for each path at each time interval. In a similar study, the authors in [4.9] utilized mixed integer programming for a dynamic network flow optimization problem. The authors proposed a heuristic solution that was applied over the time expanded transportation network, where the time horizon was divided into intervals of equal length. However, time expansion of the network made optimization problem infeasible for large scale evacuation scenarios. Coutinho-Rodrigues *et al.* [4.5] proposed a multi-objective optimization problem to find evacuation paths and the location of shelters for urban evacuation planning. The authors considered many objectives for optimization, such as path lengths, path risks, evacuation times, lengths of backup paths, and number of shelters. The set of primary routes between disaster site and shelter locations were generated with a bi-objective shortest path approach by considering the path lengths and path risks. The model was tested on a smaller network with limited roads and intersections.

Stepanov *et al.* [4.7] proposed an integer programming formulation for route assignment that utilized M/G/c/c state dependent queues to address congestion and time delays on road links. The authors computed a set of evacuation routes with kth shortest path algorithm, and then utilized M/G/c/c model to evaluate the travel time along the shortest paths. A drawback in such approach is that the shortest paths may become congested during real evacuation scenarios due to the presence of numerous unforeseen random factors, such as traffic accidents and weather conditions. The authors in [4.10] developed a traffic simulation framework that assigns evacuees with the predefined routes at the beginning of evacuations. During the journey the evacuees were able to change the routes. The authors studied the effect of non-compliance of evacuation orders by evacuees during evacuations. However, the architectures and implementation details of the proposed framework were not discussed. El-Sergany *et al.* [4.31] proposed a framework for flood disaster management and a transport distribution model for evacuations. The authors utilized linear programming on a small scale scenario with trip distribution matrix among the affected sites and destination shelters.

Huang *et al.* [4.26] presented a centralized traffic control framework for emergency vehicles. The authors utilized A* algorithm to compute three types of paths: (a) primary path between source and destination, (b) secondary path that is disjoint of the primary path, and (c) a path that connects both the primary and the secondary paths. However, authors did not mention any details about the implementation and test data of their framework.

Abdelgawad *et al.* [4.34] presented a multi-objective optimization framework that combines the vehicular traffic and mass transit system for emergency evacuation. The paper investigated the three objectives: (a) minimum in-vehicle travel time, (b) minimum at-origin waiting time, and (c) minimum fleet cost in case of mass transit. However, the authors have not included the real-time changing parameter such as road conditions that affect the real-world evacuation.

It is noteworthy to mention that most of the aforementioned optimization-based evacuation models are unable to scale well for large-scale evacuation scenarios. Therefore, most of the techniques employ various heuristics to reduce the solution space, which results in sub-optimal route recommendations. In contrast to the optimization-based approaches, there also exist some commercial/non-commercial traffic simulation packages, such as INDY [4.27], PARAMICS [4.28], DynusT [4.29], and TransCAD [4.30]. Among the aforementioned, the PARAMICS [4.28] is commercial software and has been utilized mostly for micro-scale simulations. However, a common problem that most of such packages suffer from is the lack of scalability, especially when the network size is large and different from the network under normal conditions. Therefore, to address scalability, we utilized parallel computing in our proposed simulation framework.

4.7. Conclusions

In this paper, we presented a service architecture MacroServ that performs the real-time route recommendations for the evacuees at the time of a disaster. The proposed service utilizes the live information extracted from the ITS and the road side sensors to calculate preferred evacuation paths that have maximum traffic flow capacity, least congestion, and travel cost. Unlike our approach, most of the existing work on disaster route recommendations is based on optimization techniques that compute a set of optimal routes under a specific set of parameters. However, optimization techniques are unable to precisely capture the effect of numerous stochastic and time-varying factors, which have significant influence on evacuations. Moreover, incorporating the stochastic factors in optimization models significantly increases the problem space and computation times. Therefore, to test the performance of the MacroServ service, we developed a scalable traffic simulation model that can be configured to simulate evacuations under different conditions and parameters. To achieve the desired level of scalability and speed we utilized parallel computing on a computer cluster that runs parallel instances of the real-time route computation algorithm. The evacuation simulations were performed on a real map of City of Fargo, USA consisting of 2,800 intersections, 7,370 roads, and a population size of 108,000. The simulation results indicated that by not routing the traffic towards the least congested routes during an emergency, the evacuations can suffer from massive traffic jams, which increases the evacuation times and waiting times. Moreover, the results provided best case estimates for the evacuation times under a given set of parameters and stochastic factors. The evacuation simulations can allow the disaster management bodies to plan and optimize the traffic operations during a possible evacuation. Moreover, it provides a way to better analyze the critical network elements, the effect of evacuees' behavior, and managerial factors on evacuations.

In future, we intend to extend our model by incorporating more number of parameters to address the uncertain factors during emergency scenarios. For instance, a driver's behavior may vary due to stress and fear. Moreover, evacuees' compliance to the recommended routes and time of disaster also plays an important role in the road congestions. All such real-life parameters are having significance and must be considered in the design of emergency evacuation models.

4.8. References

[4.1] J. Li, Q. Li, C. Liu, S. U. Khan, and N. Ghani, "Community-based collaborative information system for emergency management," *Computers & Operations Research*, vol. 42, pp.116–124, 2014.

[4.2] G. Galindo and R. Batta, "Review of recent developments in OR/MS research in disaster operations management," *European Journal of Operational Research*, vol. 230, pp. 201–211, 2013.

[4.3] O. Khalid, M. U. S. Khan, S. U. Khan, and A. Y. Zomaya, "OmniSuggest: A Ubiquitous Cloud based Context Aware Recommendation System for Mobile Social Networks," *IEEE Transactions on Services Computing*, vol. 7, no. 3, pp. 401-414, 2014.

[4.4] Z. Fang, Q. Li, Q. Li, D. Han, and S. Shaw, "A space–time efficiency model for optimizing intra-intersection vehicle–pedestrian evacuation movements," *Transportation*

Research Part C: Emerging Technologies, vol. 31, pp. 112–130, 2013

[4.5] J. Coutinho-Rodrigues, L. Tralhão, L. Alçada-Almeida, "Solving a location-routing problem with a multi-objective approach: the design of urban evacuation plans," *Journal of Transport Geography*, vol. 22, pp. 206–218, 2012

[4.6] G. J. Lim, S. Zangeneh, M. R. Baharnemati, and T. Assavapokee, "A capacitated network flow optimization approach for short notice evacuation planning," *European Journal of Operational Research*, vol. 223, pp. 234–245, 2012.

[4.7] A. Stepanov and J. M. Smith, "Multi-objective evacuation routing in transportation networks," *European Journal of Operational Research*, vol. 198, pp. 435–446, 2009.

[4.8] M. Saadatseresht, A. Mansourian, and M. Taleai, "Evacuation planning using multiobjective evolutionary optimization approach," *European Journal of Operational Research*, vol. 198, no. 1, pp.305-314, 2009.

[4.9] S. Bretschneider and A. Kimms, "Pattern-based evacuation planning for urban areas," *European Journal of Operational Research*, vol. 216, pp.57–69, 2012.

[4.10] A. Pel, M. Bliemer, and S. Hoogendoorn, "Modelling Traveller Behaviour under Emergency Evacuation Conditions," *European Journal of Transport and Infrastructure Research*, vol. 11, no. 2, pp. 166-193, 2011.

[4.11] Intelligent Transportation System, http://www.its. dot. gov/, accessed on March 3, 2014.

[4.12] S. M. Ross, "Introduction to probability models, 9th Edition." ISBN-13: 978-0-12-598062-3, *Academic Press*, 2006

[4.13] Red River Information, http://www.ndsu.edu/fargo_geology/whyflood.htm, accessed on March 3, 2014

[4.14] City of Fargo website: https://www.cityoffargo.com /Maps/, accessed on March 3, 2014

[4.15] North Dakota Department of Transportation (NDDOT), https://www.dot.nd.gov/, accessed on March 3, 2014

[4.16] K. Chen and J. Miles, "PIARC Handbook on Intelligent Transport System," *Artech House, London and Boston*, 1999, ISBN 1-50852-103-2, http://road-network-operations.piarc.org/index.php?option=com_content&task=view&id=38&Itemid=71&lang=en, Assessed on May 21 2014.

[4.17] L. A. Klein, M. K. Mills, and D. R.P. Gibson, "Traffic Detector Handbook: Third Edition—Volume I," FHWA Report, No. FHWA-HRT-06-108, Federal Highway Administration, McLean, VA, 2006, http://www.fhwa.dot.gov/publications /research/operations/its/06108/06108.pdf, Assessed on May 21 2014.

[4.18] Roadside weather mast, A48 east of St Nicholas, http://www.geograph.org.uk/photo/2187674

[4.19] V. W. Inman and G. W. Davis, "The Effects of In-Vehicle and Infrastructure-Based Collision Warnings at Signalized Intersections," FHWA Report, No. FHWA-HRT-09-049, Federal Highway Administration, McLean, VA, 2009, assessed on May 21 2014.

[4.20] Open Street Map API, http://www.openstreetmap.org/, accessed on March 3, 2014

[4.21] OpenMPI API, http://www.open-mpi.org/, accessed on March 3, 2014

[4.22] http://www.ccast.ndsu.edu/, accessed on March 3, 2014.

[4.23] https://www.cityoffargo.com/attachments/e83e7b08-c38a-4be2-ad1c - ad320241dfb1/Fargo%20Growth %20Plan%20 2007_Appendices.pdf

[4.24] N. T. N. Anh, Y. Chevaleyre, and Jean Daniel Zucker, "Optimizing the Placement of Evacuation Signs on Road Network with Time and Casualties in Case of a Tsunami," *IEEE 21st International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises*

(WETICE), pp. 394-396, 2012.

[4.25] D. R. Bish and H. D. Sherali, “Aggregate-level demand management in evacuation planning,” *European Journal of Operational Research*, vol. 224, pp. 79–92, 2013.

[4.26] C. Huang, C. Kung, C. Yang, C. Tseng, C.-H.A. Chou, “A Centralized Traffic Control Mechanism for Evacuation of Emergency Vehicles Using the DSRC Protocol,” *4th International Symposium on Wireless Pervasive Computing*, 2009.

[4.27] http://www.floodsite.net/html/cd_task17-19/indy.html, accessed on March 3, 2014.

[4.28] PARAMICS, http://www.paramics-online.com/, accessed on March 3, 2014.

[4.29] DynusT, http://dynust.net/, accessed on March 3, 2014.

[4.30] TransCAD, http://www.caliper.com/tcovu.htm, accessed on March 3, 2014.

[4.31] A. T. El-Sergany and S. Alam, “Trip Distribution Model for Flood Disaster Evacuation Operation,” *ITE Journal*, vol. 82, pp. 43-37, 2012.

[4.32] M. Berg, O. Cheong, M. Kreveld, and M. Overmars, “Computational Geometry: Algorithms and Applications,” *Springer Science & Business Media*, 2008.

[4.33] G. Karypis, and V. Kumar, “http://glaros.dtc.umn.edu /gkhome/metis/metis/overview,” accessed on March 2015.

[4.34] H. Abdelgawad, B. Abdulhai, and M. Wahba, “Multiobjective optimization for multimodal evacuation,” *Transportation Research Record: Journal of the Transportation Research Board* 2196, no. 1, pp.21-33, 2010.

5. CONCLUSIONS

In this chapter, we discuss the conclusion of the research we have performed during Ph.D. We carried out our research on the monitoring and analysis of dynamically changing factors on different networks for finding efficient solutions. In our research studies, we took the cases of two different networks: body area networks and road networks. In our first case study, we analyzed the body area network and monitored the effect of location of sensor on the activity recognition. Based on our study, we recommended the best possible combination of sensors' locations for different activities recognition. In the second phase of our study, we took the scenario of large scale evacuation. We monitored and analyzed the congestion being created due to the dynamic road conditions such as road densities and provide efficient route recommendation service architecture.

We presented a cloud based framework that integrates the cloud computing services with the BANs for human activity recognition. We also investigated the effects of on-body sensor locations on accuracy of activity recognition. Due to the varying energy responses that change with the position and rotation, we employed a methodology called Local Energy Based Shape Histogram (LESH) for feature extraction. The purpose of feature extraction was to represent the local regions in the sensor data matrix more comprehensively that ultimately results in the improved recognition accuracy. The LESH approach preserves maximum information about local energy by segmenting the sensors data in various regions. Three classifiers namely, the SLR, Naïve Bayes, and the SMO were used to study the relationship between the sensor location and human activity recognition. A detailed analysis shows that the task of activity recognition is highly dependent on the location of on-body sensors. Particularly, for the activities that are more dynamic and transient, using more than one sensor on different body locations simultaneously yields high recognition accuracy. The presented framework effectively integrates the cloud computing services with the BANs and has potential to be adopted practically for remote patient monitoring and therapies.

In this thesis, we also presented service architecture MacroServ that performs the real-time route recommendations for the evacuees at the time of a disaster. The proposed service utilizes the live information extracted from the ITS and the road side sensors to calculate preferred evacuation paths that have maximum traffic flow capacity, least congestion, and travel cost. Unlike our approach, most of the existing work on disaster route recommendations is based on optimization techniques that compute a set of optimal routes under a specific set of parameters. However, optimization techniques are unable to precisely capture the effect of numerous stochastic and time-varying factors, which have significant influence on evacuations. Moreover, incorporating the stochastic factors in optimization models significantly increases the problem space and computation times. Therefore, to test the performance of the MacroServ service, we developed a scalable traffic simulation model that can be configured to simulate evacuations under different conditions and parameters. To achieve the desired level of scalability and speed we utilized parallel computing on a computer cluster that runs parallel instances of the real-time route computation algorithm. The evacuation simulations were performed on a real map of City of Fargo, USA consisting of 2,800 intersections, 7,370 roads, and a population size of 108,000. The simulation results indicated that by not routing the traffic towards the least congested routes during an emergency, the evacuations can suffer from massive traffic jams, which increases the evacuation times and waiting times. Moreover, the results provided best case estimates for the evacuation times under a given set of parameters and stochastic factors. The evacuation simulations can allow the disaster management bodies to plan and optimize the traffic operations during a possible evacuation. Moreover, it provides a way to better analyze the critical network elements, the effect of evacuees' behavior, and managerial factors on evacuations.

In future, we intend to extend our model by incorporating more number of parameters to address the uncertain factors during emergency scenarios. For instance, a driver's behavior may vary due to stress and fear. Moreover, evacuees' compliance to the recommended routes and time of disaster also plays an important role in the road congestions. All such real-life parameters are having significance and must be considered in the design of emergency evacuation models.