Time Aware Hybrid Hidden Markov Models for Traffic Congestion Prediction

John FW Zaki, Amr M.T. Ali-Eldin, Sherif E. Hussein, Sabry F. Saraya, and Fayez F. Areed

Computer Engineering and Control Systems Department
Faculty of Engineering, Mansoura University, Mansoura, Egypt.

Abstract: Traffic Congestion is a socio-economic problem that swelled in the past few decades. Intelligent Transportation Systems (ITS) has become the cutting edge solution to most traffic problems. One of the important problems is the prediction of the incoming traffic pattern. There are a number of available approaches for traffic congestion prediction. One approach using NeuroFuzzy is discussed here. The approach is modified into a hybrid one using Hidden Markov Models (HMM). HMM is implemented to take into consideration time factor. It is used to select the right NeuroFuzzy network suitable for this particular time period for efficient congestion prediction. The novelty in this research is: 1) showing that the right choice of traffic pattern for training affects the quality of the prediction dramatically. 2) The results from the hybrid model showing 6% MAE rate which outperforms the standard standalone NeuroFuzzy approach of 15% error.

Keywords: Hidden Markov Models, NeuroFuzzy, Traffic Time Effect, Traffic Congestion Prediction, and Empirical Evaluation.

1. Introduction

The aim of this paper is to develop a time-aware prediction model for traffic congestion based on empirical data. Although, time has a dramatic effect on traffic congestion, most of the algorithms used in traffic prediction research account only for variables such as speed, flow, and density without taking into consideration the effect of time. For example, in the morning, there is a morning rush hour when people are going to work. The same for the afternoon, when people are returning back home. Additionally, one can argue that the different days of the week have different traffic congestion patterns. A Monday would have a different traffic pattern than a Wednesday. A weekend would neither have the same amount of traffic like a working day nor does it have traffic flowing to the same destinations at times similar to those of the working days. Most probably weekend traffic is flowing towards malls, shopping centers, and leisure destinations. The same concept applies for public holidays. It can also be claimed that time seasonality such as weeks, months, and seasons have different traffic congestion patterns. For instance, the model being developed in this research smartly accounts for some time dependency. Shankar et. al. [1] studied traffic congestion using three different fuzzy techniques to estimate traffic congestion. Each technique had speed and density inputs with three different input levels. Their best performing model is replicated and tested here using a different dataset. In this paper, some of the shortcomings of related work are mentioned and an enhancement is introduced. The proposed model considers how to better choose training data. Additionally, a novel hybrid model utilizing neurofuzzy and Hidden Markov Models (HMM) to predict traffic congestion is also developed. The model uses a neurofuzzy approach to detect traffic congestion with the assistance of HMM to automatically choose the most suitable neurofuzzy network for that particular time of the day or day of the week.

The organization of this paper is as follows: The next section provides a review of the existing techniques. The review is followed by the theory and the dataset used for empirical evaluation. Next, the contribution of this research and the analysis of the experiments are discussed. Finally, the paper draws to a close with the results and discussions, and closes with concluding remarks.

Received: September 10th, 2017. Accepted: September 24th, 2018
DOI: 10.15676/ijeei.2019.11.1.1

2. Review of Existing Techniques

Zhang and Colleagues [2] developed a Fuzzy Wavelet Neural Network algorithm and optimized it using a Quantum Particle Swarm Optimization (QPSO) algorithm. It provided good precision and stability. In another research, Li et. al. [3] used Feed-Forward Neural Network (FFNN) for traffic Prediction. Li [4] used dynamic fuzzy neural network (D-FNN) for traffic flow prediction. The algorithm automatically establishes the network structure. In an early research on Artificial Neural Networks (ANNs), Park et. al. [5] applied Radial Basis Function (RBF) and Back Propagation (BP) to short-term time series traffic volume prediction. While Kazemi and Abdollahzade [6] proposed local linear neurofuzzy model that is trained offline and adapted to online data using weighted least squares. Another approach by [7] implemented fuzzy-neural model (FNM) to predict the traffic flows in an urban network. Shankar et. al. [1] used traffic flow information captured from a traffic camera to evaluate congestion using three different fuzzy techniques. Abdulhai [8] pre-sented short-term traffic flow prediction based on a combination of both Artificial Neural Networks (ANNs) and Genetic Algorithms (GAs). In a different research, [9] developed a Hierarchical Fuzzy Rule-Based System (HFRBS) and optimized it by (GAs). The approach is accurate and robust for traffic congestion prediction.

Abu-Lebdeha [10] used ANNs to predict travel time on urban arterials for both congested and non-congested traffic conditions. Ishak and Alecsandru [11] developed a prediction query manager that decides between two prediction algorithms based on error decision algorithm. One based on ANNs architecture and the other is memory-based on the past commuter's travel experience. Boto-Giralda and Colleagues [12] developed a fuzzy ARTMAP ANN algorithm for short-term forecasting of traffic time series. They also implemented a wavelet denoising process. In his article, Celikoglu [13] introduced an ANNs for real-time mapping of traffic density in conjunction with a macroscopic traffic flow model. Another study by [14] used dynamic time-delay recurrent wavelet neural network model to predict traffic flow. Further reading available in papers such as [15-18].

From the previous review, it is clear that different approaches of traffic congestion prediction are available. However, time effect has not been studied in any of those approaches. Additionally, the use of HMM in traffic prediction is not broadly applied. Therefore, this paper proposes a model that uses HMM and NeuroFuzzy to study time effect on traffic congestion prediction.

3. Dataset

Table 1. Sample Traffic Data

LinkRef	Date	TimePeriod	AverageJT	AverageSpeed	DataQuality	LinkLength	Flow
AL1260	01/12/2014	0	114	82.11	4	2.6	10.75
AL1260	01/12/2014	1	113.71	82.31	2	2.6	10.25
AL1260	01/12/2014	2	127.19	73.59	2	2.6	8.5
AL1260	01/12/2014	3	110.75	84.51	2	2.6	6.5
AL1260	01/12/2014	4	124.7	75.06	2	2.6	5.5
AL1260	01/12/2014	5	124.31	75.3	2	2.6	5
AL1260	01/12/2014	6	123.76	75.63	4	2.6	5.5
AL1260	01/12/2014	7	121.26	77.19	4	2.6	5.5
AL1260	01/12/2014	8	115.38	81.12	4	2.6	5.75
AL1260	01/12/2014	9	120.23	77.85	4	2.6	6.5
AL1260	01/12/2014	10	132.81	70.48	2	2.6	7.5
AL1260	01/12/2014	11	115.03	81.37	2	2.6	6.5
AL1260	01/12/2014	12	122.49	76.41	2	2.6	7
AL1260	01/12/2014	13	114.28	81.9	2	2.6	11
AL1260	01/12/2014	14	120.45	77.71	2	2.6	14.5

This research is based on data collected from the Highways in England. The network is composed of 4400 miles of major motorways in England and accounts for only 2% of all England's roads [19]. England's Highway Agency made traffic data available for the public in monthly comma separated files (csv) files from 2009 to date. Each monthly file contain roughly

7 million records of traffic flow data. As shown in the sample Table 1, the data is averaged every 15 minutes for all the junctions resulting in 96 readings per junction per day (2976 readings per junction per month). In the experiments, 2499 junctions were used. Table 2 shows the explanation of the headers in Table 1.

Since the interest here is predicting traffic condition using non-deterministic models, the quality of the data is utmost importance. Hence, data mining techniques were used to extract a suitable junction data for the purpose of this research.

Table 2. Meaning of Column Headers

Variable name	Variable description
LinkRef	A unique alphanumeric link id representing a junction to junction link.
Date	Date of travel.
TimePeriod	One of 96 15-minute intervals in the day (0-95 where 0 indicates 00:00 to 00:15).
AverageJT	The average journey time to travel across the LinkRef in seconds.
AverageSpeed	The average speed (km/h) of vehicles entering the link within a given 15-minute time period.
	Indicator showing the quality of the journey time data for the link and time period.
	1 indicates the highest quality data and 5 the lowest. See below for detailed description:
	1 = Observed or vertically1 in-filled data with a good spatial match2 to the link
DataQuality	2 = Observed or vertically in-filled data with a poor spatial match to the link.
	3 = Horizontally3 in-filled data with a good spatial match to the link.
	4 = Horizontally in-filled data with a poor spatial match to the link.
	5 = No observed data so data are in-filled using free-flow data.
LinkLength	The length of the link (km).
Flow	An average of the observed flow for the link, time period and day type.

For example, a profile of a certain junction does not have any sizable congestion pattern or another junction profile that only contain slow speeds which could bias the study towards urban traffic instead of highway. Therefore, smart routines were developed to qualify the junctions based on the data profile available to suit the study at hand. A junction called ”AL1260” representing the A453 between A50 and A42 is chosen for this research.

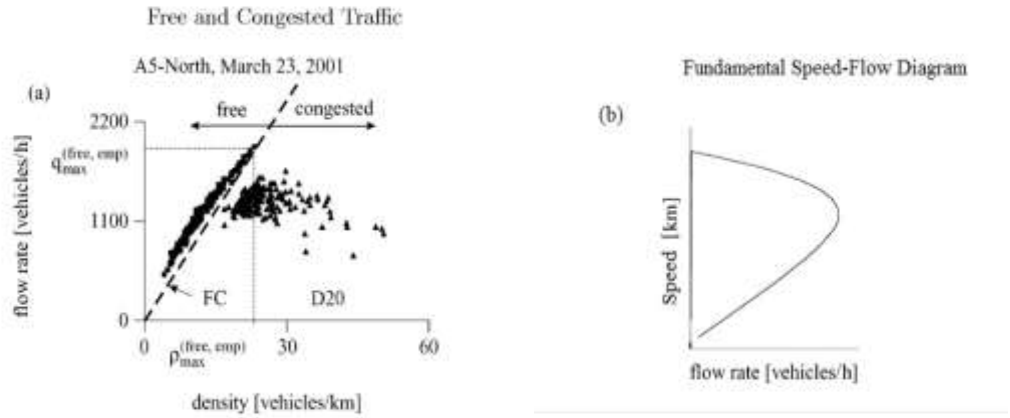

Figure 1. Fundamental Flow-Density and Speed-Flow Relationships

A standardization of the units is applied to allow for the calculation of additional variables and solid analysis. According to [20], three phase traffic theory, the fundamental flow-density relationship and the fundamental speed-flow relationship are shown in Figure. 1 (a) and (b) respectively. They represent the full profile of traffic speed, flow, and density relationships. That is, any traffic pattern may have a part or the whole of the profile shown in the fundamental relationships graphs.

This is clear when comparing the traffic profile shown in Figure. 2 of the chosen ”AL1260” to the fundamental diagrams in Figure. 1. The junction has a traffic profile containing data that is representable for all traffic conditions making it suitable for such a traffic study. After the data is selected, it is cleaned from outliers and other errors. The data point is considered an outlier if it is lower than one sixth of the sum of the two points around it [21]. If so, it is replaced by the average of those two points.

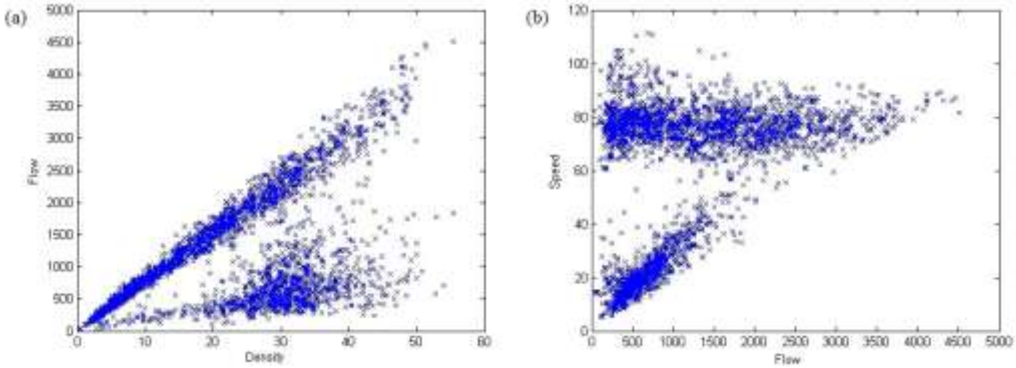

Figure 2. Collected data flow density and speed flow

From the other side, if the speed is above 140 Km/hr it is considered as outlier since the speed in miles on UK roads translate roughly to 110 Km/hr. Subsequent to the data cleaning, statistics such as mean and standard deviation are calculated. A rolling time window of 90 minutes is used to define the statistics for short-term traffic prediction. Each 90 minutes period has 6 measurements at 15 minutes intervals which forms a trajectory. This trajectory moves forward in time one step at each 15 minutes window forming rolling trajectories.

4. Methods

In this section, the adaptive neurofuzzy inference system (ANFIS) model developed by Shankar et. al. [1] is tested and modified. Furthermore, a new hybrid approach of ANFIS and HMM is developed. The new approach utilizes HMM to choose among different ANFIS models to suit a particular time of the day or day of the week. The HMM model predicts which ANFIS network is required for the next 90 minutes of traffic. It looks ahead; therefore it is able to decide which ANFIS suits the incoming traffic pattern.

In Shankar’s ANFIS model, it uses one ANFIS to predict various traffic conditions along the full day/week. This system is called here “Single ANFIS model”. Such a system is composed of two inputs, one output and the building blocks of the neuro-fuzzy layers. The two inputs are speed and density while the output is the level of congestion (LOC). Each input has three different levels of membership functions namely (slow, medium, and fast) for the speed, and (low, medium, and high) for the density as shown in Table 3. The trapezoidal input membership functions are used. They are automatically tuned using hybrid backpropagation and least squares method [22]. The output membership function is a constant and ranges from (0 – 3) with nine output levels as shown in Table 4. These levels according to [1] are: 2 free flow, 2 slow moving, 1 mild congestion, 2 heavy congestion, and 2 serious jam conditions.

Table 3. Input Membership Function

Speed Levels	Slow	Medium	Fast
	0-18	15-35	> 30
Density Levels	Low	Medium	High
	0-10	7-22	> 18

Table 4. Output Membership Function

Output levels					
free flow 1	0	free flow 2	0.67	slow moving 1	1
slow moving 2	1.33	mild congestion	1.67	heavy congestion 2	2
heavy congestion 1	2.33	serious jam 2	2.67	serious jam 1	3

Having defined the model input and output membership functions, the building blocks of the neurofuzzy system are shown in Figure 3.

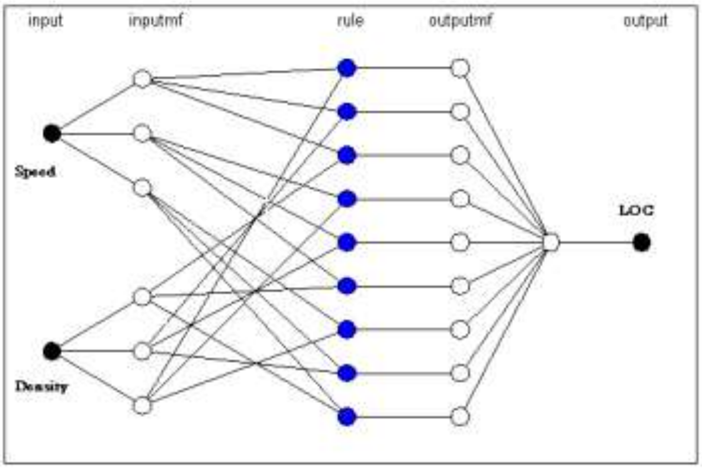

Figure 3. Neuro-Fuzzy Model

With the main elements of the system defined, the only remaining part is using the membership functions of the inputs and output to generate the fuzzy rules of the model. This is shown in Table 5 where the fuzzy inference system maps the crisp inputs into a crisp output through the four main stages of an inference system. Those are the fuzzifer, rules, inference engine, and defuzzifier [23]. The inputs are converted into fuzzy sets using membership functions through the fuzzifier stage. The next step is the inference; it is made based on a set of rules. Finally, the output is generated using output membership functions through the defuzzification stage.

Table 5. Fuzzy Rule Set

Speed Levels	Density Levels		
	Low	Medium	High
Slow	heavy congestion 1	serious jam 2	serious jam 1
Medium	slow moving 2	mild congestion	heavy congestion 2
Fast	free flow 1	free flow 2	slow moving 1

Up to this point, the model is a replication of the single ANFIS model developed in [1]. Nevertheless, the model is missing important pieces. Firstly, the data used in the model has not been verified if it is representable of all traffic states. The sample used may cover a part of the traffic full profile shown in Figure. 1. Therefore, the error of the model will definitely increase if it is tested with a traffic profile dramatically different than that used for the training. Secondly, the model was not clear on the percentage of data used for training and testing. Thirdly, it is difficult to define traffic state based on point by point measurement. For example, assuming a vehicle is traveling at a speed of 40 Km/hr. This speed can be observed at a free-flow traffic condition, medium flow condition, and a breakdown or recovery from traffic congestion. Also, the change in the speed does not show the traffic condition based on point by point. Hence, statistics are required to define a trend over a period of time that can assist in determining the state of traffic. In the following few paragraphs, an attempt to resolve the above issues is made. The model is modified and a new model is introduced.

The single neuro-fuzzy model predicts the state of congestion ranging from free flow traffic to serious congestion. To validate the assumption of full traffic profile is needed for the training; the 2499 junctions are split into two groups. The first group contains speed and density profiles that when reconstructed, they were found to cover a partial range of either free flow or congested traffic data according to Kerner's attempt to reconstruct traffic pattern [20]. The second group contains profiles of speeds and densities that cover the full range of free flow and congested traffic.

The experiment is implemented using a model that selects an arbitrary training junction from the first group which represents one month of data for that particular junction. The trained model is tested using the remaining 2498 junctions. The error is calculated using mean absolute error (MAE). The experiment is repeated with a training junction from the second group that covers the full range of speeds and densities. The trained network is tested with the remaining 2498 junctions.

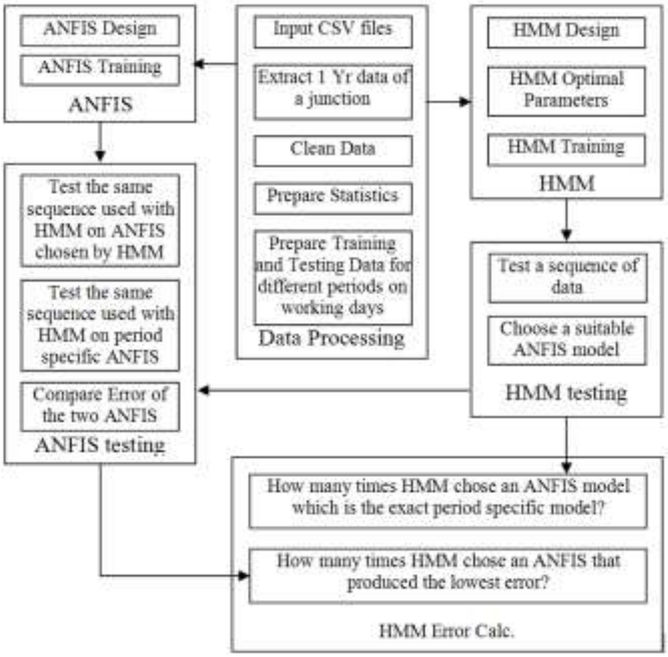

Figure 4. Proposed Hybrid Model Structure

Since no one solution fits all, it is important to use different neurofuzzy networks for different junctions. Even more, for the same junction, it is important to consider the time factor. That is, within the same day there are morning and evening rush hours. There are times of average traffic in the afternoon and late evening, and finally very low late night/early morning traffic. Additionally, within the same week, traffic differs on various weekdays. A novel model is developed here that depends on using multiple neurofuzzy networks “An Array of ANFIS models” to predict traffic on different times of the day and different days of the week. The model is smart enough to automatically switch among different networks at different times using HMM. The algorithm [24] uses expectation maximization (EM) to find the log-likelihood of the best

ANFIS network to implement for the next 90 minutes. It updates its decision every 15 minutes with a horizon of 90 minutes. Hidden Markov model is used to determine the most suitable neurofuzzy network for the incoming traffic profile as shown in Figure. 4.

It does not only switch among the right networks during the same day but also switches the right network among different days. For example, if the traffic during the rush hour of a Wednesday matches the trained profile of the Wednesday rush hour neurofuzzy network, HMM assigns the Wednesday network to predict that traffic. However, if at an instance, Wednesday traffic pattern closely matches a pattern of a Monday rush hour, the Monday neurofuzzy network is assigned to predict that traffic pattern instead of the normal Wednesday network. The HMM looks 90 minutes in advance using the Viterbi algorithm. Its decision is updated every 15 minutes.

HMM is suitable for traffic prediction since HMM is a stochastic process and traffic is stochastic in nature. In a first order observable Markov process [25], the future state probability is independent of all the past states given the current state [26, 27]. Assuming P denotes the probability function, denotes the state S at time *t*.
Then,

S_t

$$P(S_t \mid S_{t-1}, S_{t-2}, S_{t-3}, \ldots, S_1) = P(S_t \mid S_{t-1}) \tag{1}$$

Where the $P(S_t|S_{t-1})$ is called the state transition probability a_{ij} from state $S_{t-1} = S_i$ to state $S_t = S_j$ such that S_i, S_j are two distinct states.

$$a_{ij} = P(S_t = s_j \mid S_{t-1} = s_i), \qquad 1 \le i, j \le N \tag{2}$$

Where *N* is the number of states which obeys the following standard probability constraints:

$$a_{ij} \ge 0 \tag{3}$$

The initial state transition probability matrix denoted as π and defined as:

$$\sum_{j=1}^{N} a_{ij} = 1 \tag{4}$$

$$\pi = P(St) = S1 \tag{5}$$

For a Hidden Markov Model (HMM), the system state is unknown (hidden) and is only observed through a probabilistic function of an event or observation O connected to that unknown state. In turn, this introduces a new HMM probability term called the emission matrix where *M* is the number of observations and b_{jk} is:

$$B_{N\times M} \qquad b_{jk} = P(O_k\ at\ \ t \mid S_t = s_j) \tag{6}$$

Where $1 \le J \le N,\ 1 \le K \le M, t = 1,2,3, \ldots$

It represents the probability of a particular observation given a particular hidden state. For convenience, the notion is used to refer to the HMM model parameters.

Rabinar [25] answered the three main $\lambda = \{A, B, \pi\}$ problems facing the implementation of any HMM algorithm. Those are:

What are the initial model parameters that maximizes the $\lambda = \{A, B, \pi\}$ probability $P(O \mid \lambda)$ denoted as $\lambda^* = \{A^*, B^*, \pi^*\}$. Given an observation sequence $O = \{O_1, O_2, O_3, \ldots O_t\}$.

What is the $P(O|\lambda)$? Given initial model parameters $\lambda=\{A,B,\pi\}$ and an observation sequence $O=\{O_1,O_2,O_3,\ldots O_t\}$.

What is the optimal state transition sequence? Given a set of observation sequence $O=\{O_1,O_2,O_3,\ldots O_t\}$ and the HMM parameters $\lambda=\{A,B,\pi\}$.

For space limitation, summarized answers for the two most important questions are provided herein. Dempster [24] used Expectation Maximization (EM) to find the maximum likelihood of the model initial parameters. Baum-Welsh iterative training algorithm [28] is used to optimize the model parameters.

The focus now is shifted to finding the optimal state sequence associated with a given observation sequence. That is predicting the optimal state sequence. The Viterbi algorithm [29] was implemented to find the optimal state sequence $S=\{S_1,S_2,S_3,\ldots,S_T\}$ for a given observation sequence $O=\{O_1,O_2,O_3,\ldots O_t\}$. A new variable $\delta_t(i)$ is defined as follows:

$$\delta_t(i)=\max_{s_1,s_2,s3,\ldots s_{t-1}} P(S_1S_2S_3\cdots S_t=s_i,O_1O_2\cdots O_t\,|\,\lambda) \tag{7}$$

Where $\delta_t(i)$ is the highest probability along a single path which accounts for the first t observations and ends at state s_i. By induction [25]

$$\delta_{t+1}(j)=\max_i \ [\delta_t(j)a_{ij}].b_j(O_{t+1}) \tag{8}$$

To obtain the state sequence, a four steps process of initialization, recursion, termination, and path backtracking is implemented. Since the path backtracking is a backward state retrieval process, the model tracks back one step at a time to find the previous state that maximizes the probability to reach the current state.

The unknown traffic conditions are represented by the hidden states of the HMM. As stated earlier to find the states, a trend is required. That is, statistics such as the average μ, and standard deviation σ are needed for both speed and density inputs to find the unknown traffic conditions. Since the average measures central tendency of the data and the standard deviation measures the variation of the data, combining both provide better confidence in the classification and prediction [30].

An assumption regarding the transition of the states is required for HMM. That is, the transition of the states is said to be stationary over time to enable HMM application to traffic. Therefore, the daily traffic is divided into 5 periods symboled by “P#” as shown in Table 6 where the “#” is a number corresponding to the order of the period:

- 2 peak periods (morning from 5:00 a.m. to 10:00 a.m.) and afternoon from 1:30 p.m. to 6:30 p.m.)
- 2 off-peak periods (from 10:00 a.m. to 1:30 p.m.) and (from 6:30 p.m. to 12:00 a.m.) with average traffic.
- 1 off-peak period (from 12:00 a.m. to 5:00 a.m.) with low traffic.

Table 6. Daily and Weekly Periods

	Mon	Tue	Wed	Thur	Fri
00:00-5:00	P1	P6	P11	P16	P21
5:00-10:00	P2	P7	P12	P17	P22
10:00-13:30	P3	P8	P13	P18	P23
13:30-18:30	P4	P9	P14	P19	P24
18:30-00:00	P5	P10	P15	P20	P25

The above daily split is applied to the 5 working days of the week. This will result in 25 different neurofuzzy networks. Each network will be trained with traffic pattern from its corresponding period. Hence, each ANFIS model is defined and referred to later in the text as period specific or model corresponding to particular time period. The same for HMM construction and training, there must be a number of HMMs equal to the number of neurofuzzy networks; one for each period. HMMs will choose the most suitable neurofuzzy network for prediction according to the incoming traffic pattern. It is cruital for the integrity of the study to assume stationary traffic over each of the periods defined above. Figure. 5, 6, 7, 8, and 9 show sample of traffic data for different periods of the day. They show periods of free flow traffic, medium flow and congestion. It is evident from Figure. 10 which represents one month of data that there are periods of congestion on daily basis where speed drops dramatically.

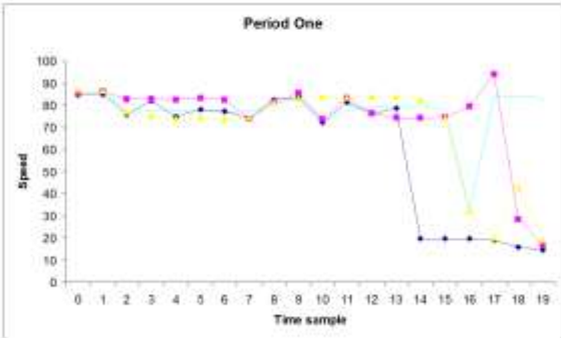

Figure 5. Sample of Period One

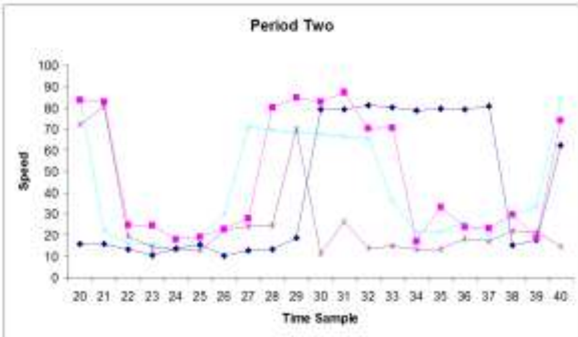

Figure 6. Sample of Period Two

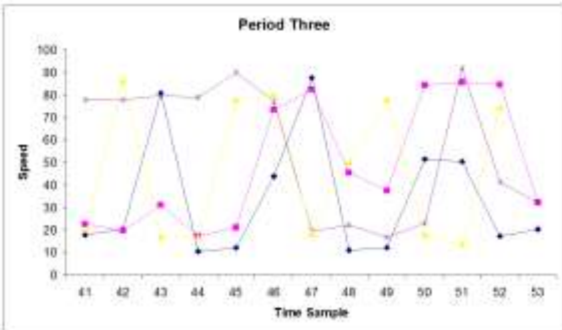

Figure 7. Sample of Period Three

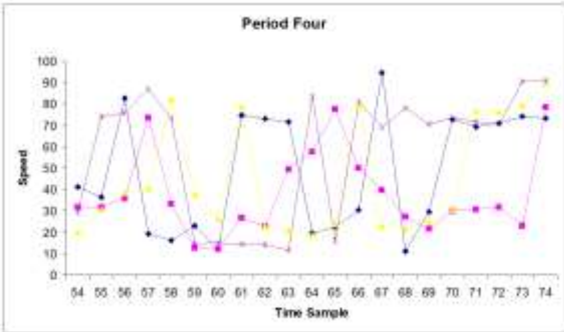

Figure 8. Sample of Period Four

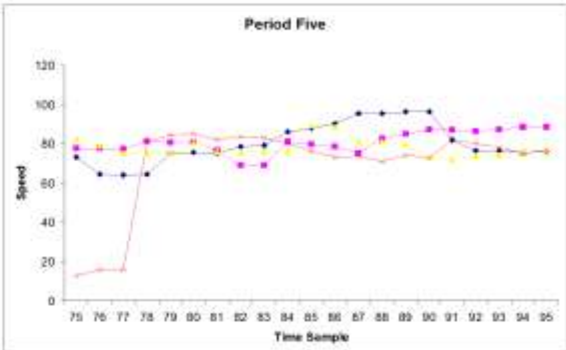

Figure 9. Sample of Period Five

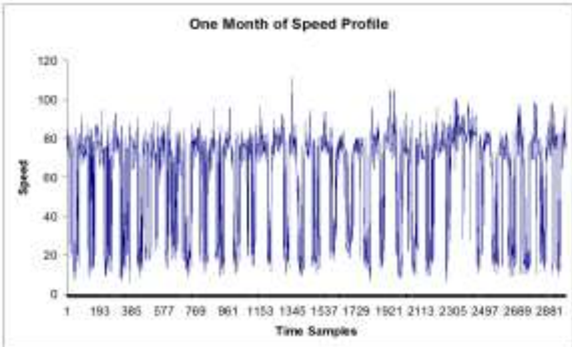

Figure 10. Sample of one month Data

The application of the above method is applied over the previously chosen junction in the dataset section (AL1260). A one year data for the chosen junction is used. A 70% of the junction data is used for training and 30% for testing. Both the 70% and 30% data are selected from different seasons in the year to reduce the effect of traffic seasonality within the year. After the data is selected, it is cleaned from outliers and other errors. Subsequent to the data cleaning, statistics for both speed and density are calculated. A rolling time window of 90 minutes is used to define the statistics for short-term traffic prediction in each period. Each 90 minutes period has 6 measurements at 15 minutes intervals which forms a trajectory. This trajectory moves forward in time one step at each 15 minutes window.

An experiment to verify the model is conducted. The different neurofuzzy networks are used statically in their time periods. For instance, the network trained using Monday morning rush hour is used to predict traffic on Monday morning rush hour. The results of that are compared to the results of using one neurofuzzy network to cover the full week traffic prediction. Following this, the different neurofuzzy networks are dynamically selected using HMM to predict traffic during different times of the day and days of the week.

5. Results and Discussions

The single neuro-fuzzy model replicated here is trained and tested on a different dataset. It predicts the state of congestion ranging from free flow traffic to serious congestion. The model is trained using speed and density inputs and produces LOC as output. This output is compared to human decision of LOC (actual LOC).

Figure 11 shows the results of testing a neurofuzzy network from the first group (partial traffic profile). The MAE is 15%. While Figure. 12 shows the results of testing a neurofuzzy network from the second group. That is, a network trained with full traffic profile. The MAE is 11%. It is clear from the results that using the incorrect training profile leads to higher error rates.

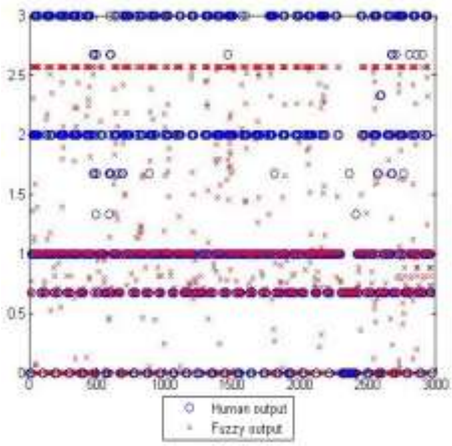

Figure 11. Testing Arbitrary Junction

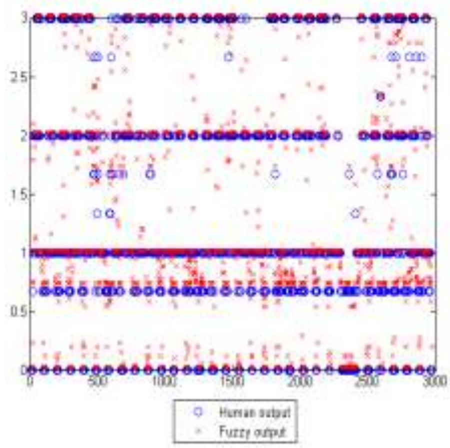

Figure 12. Testing a specific network

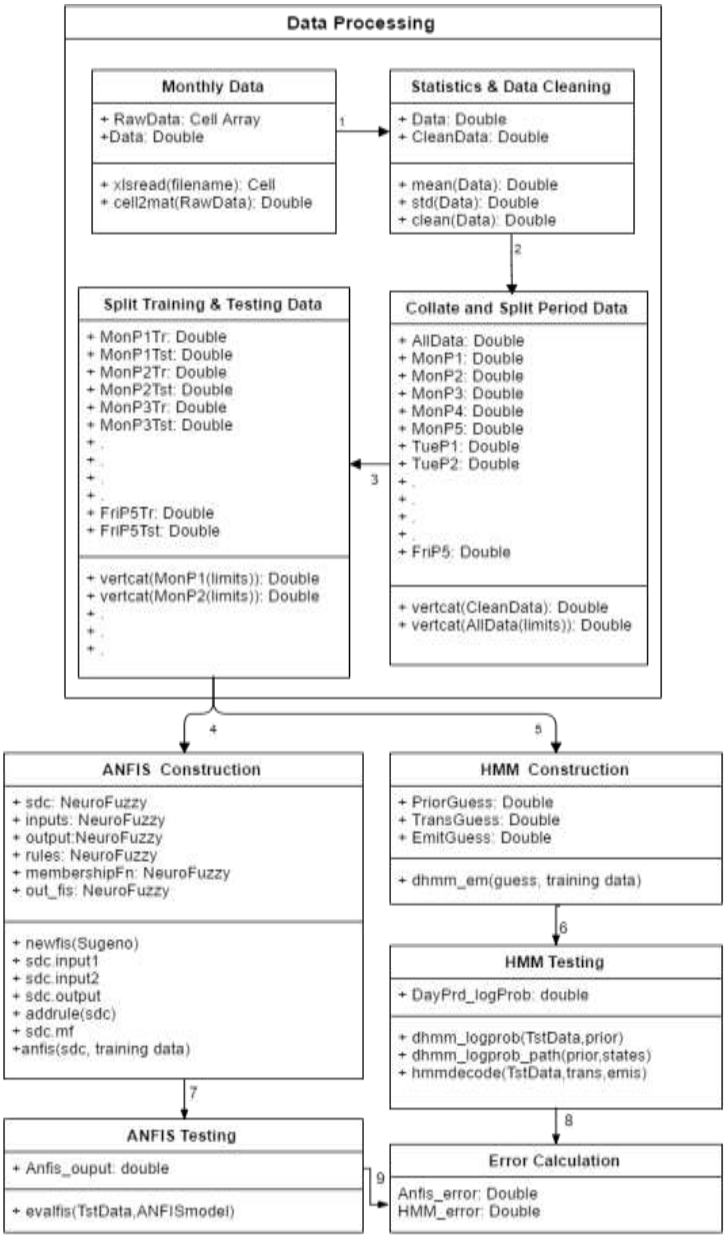

Figure 13. High-level design of the hybrid algorithm
For MATLAB implementation

The model using different neurofuzzy networks during different times of the day and days of the week to predict traffic states is tested. The networks are fixed statically corresponding to different time periods. The result of this experiment is compared to single neuro-fuzzy model approach to predict traffic during the whole week. The error is 9% which outperforms [1] model. Additionally, the novel HMM-neurofuzzy hybrid model where HMM chooses the most suitable ANFIS to predict traffic is experimented and its performance is evaluated.Herein, Figure. 13 shows UML design for the hybrid model implementation in MATLAB.

The data comes in monthly files, each file contain all the junctions traffic of the UK motorways. The model extracts one year data of a particular junction from each monthly file. The model then cleans the data and calculates statistics of mean and standard deviation to be used with HMM. It then prepares training and testing data from different seasons. This is obtained by selecting two months for training and one month for testing from each of the four seasons. Data manipulation techniques are applied to group similar daily data together and split the data into the give different groups as in Table 6. Each HMM and ANFIS pair is trained with data from its corresponding time period. ANFIS is trained with actual data as shown in Figure. 14 while HMM is trained using statistics.

In this process, the optimal HMM parameters are obtained. Once training is completed, a test vector is fed to the 25 HMMs where the best ANFIS is chosen based on the log-likelihood of the HMMs. The chosen ANFIS is tested with the same test vector and its error is recorded. The same test vector is applied to the ANFIS corresponding to the time period. The error is also recorded and compared to the error of the HMM chosen ANFIS. HMM is considered to have an error based on the number of hit/miss choices. That is, it chooses the ANFIS producing the smallest error. The approach uses HMM to lookahead in order to select a suitable neurofuzzy network to predict the incoming traffic state. Each day is split into 5 periods and the week is split into 5 working days excluding weekend. This results in a total of 25 different time periods which require 25 different HMMs and their corresponding 25 neurofuzzy networks. As a starting point, care is excerised while dealing with big data to ensure its integrity across the 25 different periods while training both HMMs and neurofuzzy networks. The HMMs are trained using statistics calculated from the incoming traffic profiles of corresponding time periods. It looks forward 90 minutes using path backtracking of Viterbi Algorithm and its decision is updated every 15 minutes.

The HMMs decides for the incoming traffic pattern which is the most suitable neurofuzzy network to best predict the incoming traffic state. This is obtained by finding the Log-Likelihood of the incoming traffic pattern using HMMs. The lowest Log-Likelihood value corresponds to the HMM best representing the incoming traffic pattern. HMM then selects its corresponding neurofuzzy network to predict the incoming traffic. The decision of HMM is verified by feeding the test vector to all the 25 neurofuzzy networks and measuring the resulting error. Intuitively, one of two results is expected. The first result is that the neurofuzzy network corresponding to this particular period of the day should produce the smallest testing error. The second result is that the neurofuzzy network chosen by the HMM should produce the smallest testing error. That is, in cases where the neurofuzzy network chosen by HMM differs from the neurofuzzy network corresponding to that particular time period, its error is smaller than the error produced by the neurofuzzy network corresponding to that particular time period. The most important outcome is that in almost 75% of the cases, the HMM chosen network is the exact neurofuzzy network corresponding to that particular time period. Only 25% of the cases had HMM decision different from the actual network corresponding to the particular period under consideration as shown in Table 7.

This proves the importance of the assumption regarding the relation between time and the nature of traffic. Traffic patterns are different during different times of the day and days of the week. Nevertheless, traffic is similar at similar times of same days on different weeks. That is, Wednesday rush hour traffic this week is similar to Wednesday rush hour traffic next week and the same applies of other days and other periods of the day. This fact is extremely important in helping the decision makers to plan various activities such as road works, closures, events etc.

Table 7. Testing HMM Decision

Total No. of test vectors	2840
HMM decision produces nuerofuzzy network corresponding to time period	2130
HMM decision produces nuerofuzzy network different to time period	710

Additionally, testing the traffic sequence vectors using the neurofuzzy networks suggested by HMM, the error produced by the hybrid model outperforms other tested models. It produced an MAE error of almost 6% which is much smaller in comparison to 15% of [1] model. Table 8 summarizes the ANFIS error of all models used.

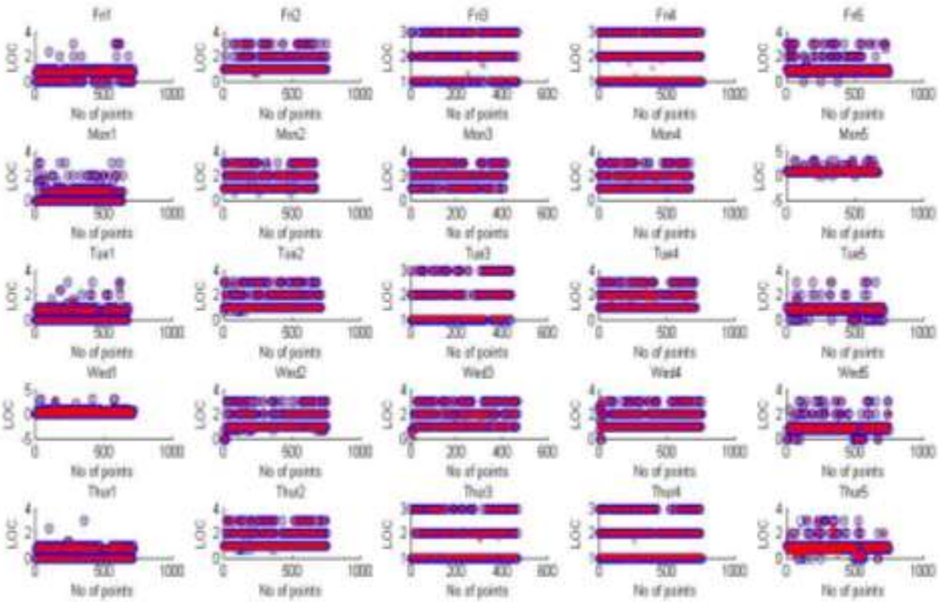

Figure 14. Training of the 25 ANFIS

Table 8. Summary of Different ANFIS errors

Method	Error
Single ANFIS without carefully chosen training data	15%
Single ANFIS with carefully chosen training data	11%
Array of ANFIS (static)	9%
Array of ANFIS with HMM	6%

6. Limitations and Challenges

In [1] approach, one ANFIS network was used to predict traffic congestion. In this paper, the hybrid approach utilizes one ANFIS network for each time period where there are 5 periods per working day of the week. This results in 25 different ANFIS networks per working week. Those 25 networks need to be evaluated each time a test vector is introduced. However, practically speaking, only two ANFIS needs to be evaluated. Those are the ANFIS chosen by HMM and the ANFIS corresponding to the specific time period. Therefore, theoretically speaking, the model uses twice as much time to take a decision in comparison to single ANFIS approach if it is run sequentially.

The code is written in MATLAB and the time used to execute the code was measured in each case. Single ANFIS approach uses 17 seconds while the new approach uses 40 seconds for testing two ANFIS models. A complete run of the 25 ANFIS and 25 HMMs consume 550 seconds. In addition, HMM adds more complexity to the system as it evaluates the test vector first. It is therefore important to note that for practical implementation, the ANFIS and HMM must be trained once in order to operate online. For the system update, it can be calibrated or re-trained offline as new data becomes available due to the size of the data and structure of the model.

From analytical complexity point of view, it is extremely difficult to measure the asymptotic behavior of individual MATLAB functions. Therefore, the focus here is on the complexity of the actual code written. Both models, the standard single ANFIS and the hybrid HMM-ANFIS have asymptotic behavior of O(n2).

7. Conclusion

In this paper, a time aware hybrid HMM-ANFIS model is introduced. The model takes into consideration that traffic differs according to different times of the day and various days of the week. In the model, HMM is employed to choose an ANFIS model from the pool of available ANFIS models to predict the incoming traffic. The decision lookahead horizon is 90 minutes and is updated every 15 minutes using Viterbi Algorithm. Testing the proposed model, it is found that 75% of the time the chosen ANFIS model by HMM is the one preallocated for this particular time period while 25% of the time the decision allocates a different ANFIS model than the one preallocated for the period. At first thought, this result was unexpected, however it makes perfect sense. The reason being, the ANFIS is trained with data profile specific to particular time period of the day or day of the week.

Therefore, it is more likely to be chosen by HMM as the best network to classify an unknown test sequence from its corresponding time period. There are instances where an unknown test vector would be best represented by a network different than its corresponding time period. The result of such a change in the network to use has reduced the prediction error of the ANFIS model to 6%. This outperforms the standard stand alone neurofuzzy approach. Further, the paper discusses the proper choice of training and testing data through an empirical evaluation. Future work includes testing the effect of seasonality on traffic congestion pattern.

8. References

[1]. Shankar, H., P.L.N. Raju, and K.R.M. Rao, Multi Model Criteria for the Estimation of Road Traffic Congestion from Traffic Flow Information Based on Fuzzy Logic. *Journal of Transportation Technologies*, 2012. 2: p. 50-62.

[2]. Zhang, K., et al., A Network Traffic Prediction Model Based on Quantum-Behaved Particle Swarm Optimization Algorithm and Fuzzy Wavelet Neural Network. *Discrete Dynamics in Nature and Society*, 2016. March: p. 1-11.

[3]. Li, L., et al., Trend Modeling for Traffic Time Series Analysis: An Integrated Study. *IEEE Transactions on Intelligent Transportation Systems*, 2015. 16(6): p. 3430-3439.

[4]. Li, H., Research on prediction of traffic flow based on dynamic fuzzy neural networks. Springer, *Neural Computing and Applications*, 2015(1-12).

[5]. Park, B., C. Messer, and T. Urbanik, Short-Term Freeway Traffic Volume Forecasting Using Radial Basis Function Neural Network. Transportation Res Rec, *Journal of the Transportation Research Board*, 1998. 1651(1): p. 39-47.

[6]. Kazemi, R. and M. Abdollahzade, An adaptive framework to enhance microscopic traffic modelling: an online neuro-fuzzy approach. *Proceedings of The Institution of Mechanical Engineers Part D, Journal of Automobile Engineering*., 2016.

[7]. Yin, H., et al., Urban Traffic Flow Prediction Using a Fuzzy-neural Approach. Transportation Research Part C Emerging Technologies, 2002. 10: p. 85-98.

[8]. Abdulhai, B., H. Porwal, and W. Recker, Short-Term Traffic Flow Prediction Using Neuro-Gentic Algorithms. ITS Journal, Copyright Taylor & Francis, 2002. 7(3–41).

[9]. Zhang, X., et al., Hierarchical fuzzy rule-based system optimized with genetic algorithms for short term traffic congestion prediction. *Transportation Research Part C Emerging Technologies*, 2014. 43: p. 127-142.

[10]. Abu-Lebdeha, G. and A.K. Singhb, Modeling Arterial Travel Time with Limited Traffic Variables using Conditional Independence Graphs & State-Space Neural Networks. *Procedia - Social and Behavioral Science, El-Sevier*, 2011.

[11]. Ishak, S. and C. Alecsandru, Hybrid Model-Based and Memory-Based Traffic Prediction System. *Transportation Research Record Journal of the Transportation Research Board*, 2004. 1879(1): p. 59-70.
[12]. Boto-Giralda, D., et al., Wavelet-Based Denoising for Traffic Volume Time Series Forecasting with Self-Organizing Neural Networks. *Computer-Aided Civil and Infrastructure Engineering*, 2010. 25: p. 530–545.
[13]. Celikoglu, H.B., An Approach to Dynamic Classification of Traffic Flow Patterns. *Computer-Aided Civil and Infrastructure Engineering*, 2013. 28: p. 273–288.
[14]. Jiang, X. and H. Adeli, Dynamic Wavelet Neural Network Model for Traffic Flow Forecasting. *Journal of Transportation Engineering*, 2005. 131(10): p. 771-779.
[15]. Oh, C. and S. Park, Investigating the effects of daily travel time patterns on short-term prediction. *KSCE Journal of Civil Engineering*, 2011. 15(7): p. 1263-1272.
[16]. Quek, C., M. Pasquier, and B.B.S. Lim, a novel fuzzy neural approach to road traffic analysis and prediction. *IEEE Transactions on Intelligent Transportation Systems*, 2006. 7(2): p. 133–146.
[17]. Srinivasan, D., C.W. Chan, and P.G. Balaji, Computational intelligence-based congestion prediction for a dynamic urban street network. *Neurocomputing*, 2009. 72: p. 2710-2716.
[18]. Zhang, Y. and Z. Ye, Short-Term Traffic Flow Forecasting Using Fuzzy Logic System Methods. *Journal of Intelligent Transportation Systems*, 2008. 12: p. 102-112.
[19]. UK-Government. http://data.gov.uk/dataset/dft-eng-srn-routes-journey-times.
[20]. Kerner, B.S., The Physics of Traffic. 2004: Springer.
[21]. Qi, Y. and S. Ishak, Probabilistic Models for Short Term Traffic Conditions Prediction. 2014, *Louisiana State University The Department of Civil and Environment Engineering*.
[22]. Mitra, P., ANFIS based automatic Voltage Regulator with Hybrid Learning Algorithm. International Journal of advances in soft computing and applications, 2010. 2.
[23]. Mendel, J., Fuzzy logic systems for engineering: a tutorial. *Proceedings of the IEEE*, 1995. 83(3): p. 345-377.
[24]. Dempster, A.P., N.M. Laird, and R.B. Rubin, Maximum Likelihood from Incomplete Data via the EM algorithm. *Journal of Royal Statistics Society*, 1977. Vol 39(1): p. 1-38.
[25]. Rabiner, L.R., A tutorial on Hidden Markov Models and Selected AApplication in Speech Recognition. *IEEE*, 1989. 77(2): p. 257-287.
[26]. Hoel, P.G., S.C. Port, and C.J. Stone, Introduction to Stochastic Processes. Vol. (ISBN 0-88133-267-4). 1987: Waveland Press, Inc.
[27]. Levin, D.A., Y. Peres, and E.L. Wilmer, Markov Chains and Mixing Times. 2008: *American Mathematical Society*.
[28]. Baum, L.L.E. and G.R. Sell, Growth Functions for Transformations on Manifolds. *Pacific Journal of Mathematics*, 1968. 27(2): p. 211-227.
[29]. Viterbi, A.J., The Viterbi Algorithm. *Proceedings of IEEE*, 1973. 61: p. 268-278.
[30]. Ishak, S. and C. Alecsandru, Analysis of freeway traffic incident conditions by using second-order spatiotemporal traffic performance measures. Transportation Res Rec, *Journal of the Transportation Research Board*, 2005. 1925: p. 20 – 28.

John F.W. Zaki is a lecturer at Faculty of Engineering, Mansoura University His research interests are Intelligent Transportation Systems (ITS), Internet of Things and (IOT) Robotics.

Amr M.T. Ali-Eldin is an associate professor of computer engineering at Faculty of Engineering, Mansoura University. His research interests are Privacy, Security, Software engineering and Data Analytics.

Sherif E. Hussein is an associate professor of computer engineering at Faculty of Engineering, Mansoura University. His area of interests is Cloud Computing, Mobile applications, Multi-agent Systems, System Dynamics and Image Processing.

Sabry F. Saraya is an associate professor emeritus of control engineering at Faculty of Engineering, Mansoura University. His area of interests is adaptive, optimal and intelligent control systems.

Fayez F. Areed is a professor emeritus of control engineering at Faculty of Engineering, Mansoura University. His area of interests is adaptive, optimal and intelligent control systems.