

Article

Detecting Urban Transport Modes Using a Hybrid Knowledge Driven Framework from GPS Trajectory

Rahul Deb Das * and Stephan Winter

Department of Infrastructure Engineering, The University of Melbourne, Melbourne, VIC 3010, Australia;
winter@unimelb.edu.au

* Correspondence: rahuld@student.unimelb.edu.au

Academic Editors: Bin Jiang, Constantinos Antoniou and Wolfgang Kainz

Received: 16 September 2016; Accepted: 28 October 2016; Published: 9 November 2016

Abstract: Transport mode information is essential for understanding people’s movement behavior and travel demand estimation. Current approaches extract travel information once the travel is complete. Such approaches are limited in terms of generating just-in-time information for a number of mobility based applications, e.g., real time mode specific patronage estimation. In order to detect the transport modalities from GPS trajectories, various machine learning approaches have already been explored. However, the majority of them produce only a single conclusion from a given set of evidences, ignoring the uncertainty of any mode classification. Also, the existing machine learning approaches fall short in explaining their reasoning scheme. In contrast, a fuzzy expert system can explain its reasoning scheme in a human readable format along with a provision of inferring different outcome possibilities, but lacks the adaptivity and learning ability of machine learning. In this paper, a novel hybrid knowledge driven framework is developed by integrating a fuzzy logic and a neural network to complement each other’s limitations. Thus the aim of this paper is to automate the tuning process in order to generate an intelligent hybrid model that can perform effectively in near-real time mode detection using GPS trajectory. Tests demonstrate that a hybrid knowledge driven model works better than a purely knowledge driven model and at per the machine learning models in the context of transport mode detection.

Keywords: GPS; trajectory; fuzzy logic; neuro-fuzzy; transport mode; context; travel behaviour

1. Introduction

Understanding travel behaviour is core to transport planning and various context-aware mobility services. This paper focuses on detecting travel modes in urban mobility in near-real time, where the context relates to urban mobility based activities [1,2]. Traditionally, travel behaviour information is collected from memory by paper-based or telephone surveys. Since there is a time gap between the actual travel and the reporting time, such information is subject to mis-reporting or under-reporting [3]. Other, sensor-based methods, such as traffic counts [4] or smart card use, [5] are isolated, incomplete and inconclusive. However, with the emergence of positioning sensors and inertial measurement units (IMU) on-board of smartphones, recording a smartphone user’s mobility-based activities (such as transport modes) using those mobile sensors is currently being researched [6,7]. Transport mode information is a critical component of travel demand estimation and movement behaviour analysis at an individual and collective level [8]. In addition, detecting transport modes in real time can enable various services, such as mode-specific mobile notification and context-aware auto-answering. For example, while driving, it can be automatically activated the auto-answer on receiving a call in order to avoid any distraction on the road. A near-real time mode detection can also help in emergency situations, near-real time patronage estimation along a given route and ad hoc policy enforcement.

A raw trajectory, as captured by GPS, represents only the geometrical property of movement and cannot reveal underlying behaviour information. Hence, there is a semantic gap between a raw trajectory and an individual's movement behaviour. Research in moving object databases and trajectory analysis started looking at extracting different information by enriching a raw trajectory by domain knowledge and infrastructure information and thereby bridging the gap adequately [9,10].

The majority of transport mode detection research is about offline inference based on completed trajectories. In contrast, near-real-time transport mode detection is comparatively a new concept. Transport mode detection, being a classification problem, has been approached by artificial neural networks (ANN), support vector machines (SVM), decision trees (DT) and several other machine learning techniques so far [11–17], and less so by knowledge-driven models [18,19].

Having said that, the predictive ability and, thus, the performance of traditional machine learning methods primarily depend on the feature vectors (with input-output mapping) used during the training phase. In a classification problem, a machine learning approach generally predicts a given class for an input test feature vector without notion of uncertainty or confidence in its prediction, and thus, cannot provide alternative predictions. There are also cases when there is a scarcity of getting a sufficient amount of training data or an incomplete universe of discourse.

Capturing uncertainty in mode detection is also important, as the kinematic observations (speed profile, proximity behaviour) may often be vague and uncertain (given a fine-grained, dense GPS trajectory), with the possibilities of being different classes with various degree of truth (certainty), which a machine learning technique cannot address. Such uncertainty can be modelled through a knowledge-driven expert system approach. In this paper, we will model the uncertainty through a hybrid expert reasoning model. Knowledge-based reasoning incorporates expert knowledge (and thus, the model is flexible and provides an interaction with a number of domain experts) in terms of simple IF-THEN rules. The rules can be extended or reduced. Each rule can also incorporate any number of pieces of information through different *t*-norm operators, and thus, it is easy to develop a generic behavioral model for a given problem domain. Expert reasoning is flexible, more expressive and intuitive, whereas a machine learning model offers limited expressive power and needs to be trained every time with a new training sample when there is a need to upgrade the model. Since a knowledge-driven (expert system) model can explain its reasoning scheme, hence the model can also reflect any anomaly in user's movement pattern or his/her driving behaviour based on the set of rules that are fired for a given set of input features, which is not very prominent by a machine learning model.

Fuzzy expert systems are powerful expert system models as they can handle the uncertainty and vagueness in a human-understandable format. The success of a fuzzy expert system lies in its proper selection of membership functions and their parameters, which is generally done manually. Thus, fuzzy logic-based models provide limited capacity under varying conditions, especially when there are large numbers of fuzzy variables [20]. Applied to trajectory interpretation, fuzzy models may not be of consistent quality with their given rule sets along with their membership function(s) over a given travel due to GPS multipath and signal loss, especially on shorter segments in the context of near-real time mode detection. Thus, a fuzzy inference model needs an automation to select its membership function parameters automatically by learning from a given input-output mapping. This is accomplished by integrating machine learning with a fuzzy system: only a neuro-fuzzy system is capable of developing a fuzzy expert system automatically with the provision of making modifications by experts in later phases without having proper training data.

Therefore, this paper proposes an integrated multi-layered hybrid neuro-fuzzy framework for transport mode detection in the interest of public transport infrastructure. The framework combines an artificial neural network (ANN) with a Sugeno-type fuzzy logic (see Section 3.3) in order to enable the fuzzy inference system (FIS) to tune its parameters through an iterative learning process. At the same time, the model also makes the ANN more transparent by expressing the fuzzy knowledge base and reasoning scheme. The aim of this paper is to develop a novel multi-layered neuro-fuzzy-based

hybrid intelligent knowledge-driven model that can perform better than Mamdani-type fuzzy models (see Section 3.3) and work at par with some of the state-of-the-art machine learning models, but has the ability to explain the reasoning scheme for near-real time mode detection. Thus, this paper hypothesizes that *a hybrid neuro-fuzzy approach will bridge the gap of ability to represent knowledge and learning capacity in an uncertain condition and compensate the trade-off between a machine learning-based approach and a fuzzy logic expert system and can develop a more robust and transparent classification than that of its counterparts in transport mode detection in near-real time.*

The contributions of this paper are as follows: (a) To the best of the authors' knowledge, this is the first work in the field of transport mode detection where a hybrid intelligent model is developed using a machine learning approach and a fuzzy expert system. (b) This work investigates the performance of a hybrid knowledge-driven model compared to a purely knowledge-driven model and machine learning models. (c) This paper also presents a novel approach to deal with multi-class problems, using a multi-layered neuro-fuzzy model. Most of the hybrid models used in other areas of transportation research deal with regression problems such as travel time estimation, demand estimation or flow behaviour. However, this paper develops an adaptive and multi-layered hybrid intelligent model to address the transport mode classification problem.

In this paper, we have introduced the term "near-real time" for the first time in transport mode detection research. Detecting transport modes in near-real time resembles to activity recognition on a second-by-second basis in pervasive and mobile computing [21,22]. However, for practical reasons, activity recognition (in the context of body part movement and gesture recognition) at a finer granularity based on inertial sensor data can be performed comparatively within a shorter time window (typically in the order to 1–10 s) than the length required for transport mode detection using GPS (typically in the order of 60–120 s). This temporal difference is due to the temporal delay in GPS signal updates typical on commercial smartphones. Thus, instead of using the term "real time", we use "near-real time" to indicate the granularity of the query window in a qualitative way. The model presented in this paper is compared with a multiple input multiple output (MIMO) Mamdani fuzzy inference system (MFIS) and some machine learning models, and the results demonstrate the efficacy of the model in terms of its consistency in performance and reasoning ability.

The remainder of the paper is organized as follows. Section 2 contains the literature review followed by Section 3 containing a brief theory of the fuzzy and neuro-fuzzy model. In Section 4, a near-real time mode detection architecture is presented, followed by an implementation and evaluation in Section 5. A discussion is presented in Section 6. In Section 7, conclusions and possible future research directions are presented.

2. Literature Review

Travel is a central concept in human behaviour, transport geography, travel demand modelling, public health research, location-based services (LBS) and mobile computing. Each of these disciplines requires a variety of information related to travel, at varied granularities. In urban contexts, different types of travel-related information (e.g., departure times, arrival times, travel routes, durations, modality, company, and trip purposes) are of interest for managing traffic, understanding people's travel demand and preferences and also policy making. Such travel related information, including information on transport modality, is traditionally collected by paper-based or telephone surveys [23]. However, traditional surveys are memory dependent and, thus, suffer from data quality issues. Traditional travel surveys are also limited to low participation rates. In order to overcome such problems, more recently smartphone-based travel surveys have been carried out [8,24] where the participants do not need to memorize the details of their travels. Instead, their movements are recorded as raw trajectories by GPS and other sensors on-board their smartphones. An application accessing these sensors can automatically collect travel data in the background without user intervention. These raw data need to be semantically enriched by algorithms capable of interpreting such data. Only then, spatial database management systems can extract different travel related information

from the raw data in order to answer corresponding queries [10]. Detecting the mode of travel is an important task in this trajectory inference process.

The existing literature explores historical trajectories (i.e., solves the interpretation of the raw data once the trip is complete) in order to infer the modalities that had been used during the travel, based on different features. Such detection frameworks are called offline mode detection models. Since a travel can take place by more than one modality, there is a need to break the entire trajectory into trajectory segments travelled in the same mode. Assuming walking is necessary in between any two other modes, Zheng and colleagues proposed a walking-based segmentation and then inferred the mode information on the trajectory [17]. They tested four different machine learning models, decision tree (DT), Bayesian network (BN), conditional random Field (CRF) and support vector machines (SVM), obtaining the maximum accuracy of 75% using DT. They classified four modalities with five kinematic features using GPS signals. Liao and colleagues also used GPS information to develop a CRF-based model to infer the modes along trajectories [25]. Dodge and colleagues used GPS trajectories and developed an SVM-based model with 82% accuracy for four modes using three kinematic features [26]. Dodge and colleagues have introduced the concept of global features and local features computed from a trajectory. They have primarily used variation in sinuosity and the deviation of different kinematic features, such as velocity and acceleration, from the median line [26]. Since GPS comes with varying positional accuracy due to various environmental factors and signal shortages, recently, there is a trend to integrate different inertial navigation sensors, such as accelerometers, with GPS, which is explored in [14].

Stenneth and colleagues used infrastructure information and speed information to distinguish five modalities using five features. They developed a decision tree-based mode detection model that yields 93.5% accuracy [15]. Ohashi and colleagues developed a vibration-based mode detection model using a Bayesian network with 80% accuracy with a focus on a fine distinction between a car and a motorbike, which is deemed to be a challenging problem, since both of them share the same route network and show almost a similar speed and acceleration profile. They have collected the vibration sensor signal on-board a smartphone to capture the vibration profile of different modalities. However, they do not attempt to segment the trajectory and did not address the issue of composite modes [27]. Gonzalez and colleagues developed a neural network-based mode detection model using GPS sensors. They distinguish three modalities using eight features with 91% accuracy [13]. Hemminki and colleagues used accelerometer to detect modality. They have used a discrete hidden Markov model (DHMM) and AdaBoost, with which they obtained 84.2% accuracy [28].

Compared to the rich offline mode detection research, there have been only a few near-real time mode detection attempts made so far. In one of them, Byon and colleagues developed a neural network-based mode detection model using three kinematic features in near-real time on four modalities. They claim the model works best on ten-minute query windows, for which they obtained 82% accuracy. However, ten-minute time windows may be too long for certain applications, such as emergency services or context-sensitive location-based services [11,12], and may lose out with more frequent change of modes.

With that in mind, Reddy and colleagues' mode detection framework that works on a second-by-second basis with 74% accuracy [14] is more relevant in real-time scenarios. Since GPS sensors on a commercial smartphone cannot sample at finer granularity due to hardware and software limitations, the literature shows that such small temporal query windows require additional sensors (at least an inertial navigation sensor) that can sample at higher frequencies. Reddy and colleagues utilized the accelerometer on-board a smartphone to calculate acceleration-based features. They also used the GPS sensors to get the speed value over one-second intervals.

However, machine learning-based mode detection models require a substantial training data to train the models and also lack explanatory power. On the other hand, fuzzy logic-based mode detection models do not require any training. Here, the model is developed based on expert knowledge. Fuzzy logic-based models express the knowledge base in simple IF-THEN rules. The models can

also handle uncertainty, vagueness and imprecision. Schussler and Axhausen developed a fuzzy logic-based mode detection model on five modalities using three speed-only features [29]. Xu and colleagues developed a fuzzy logic based model that can distinguish four modalities with 93.7% accuracy [30], and Biljecki and colleagues developed a Sugeno-type fuzzy logic-based mode detection framework that can classify ten modalities with 91.6% accuracy [18].

The success of any fuzzy logic-based model depends on the expert knowledge brought in and the consistency between a particular observation and the universe of discourse for each fuzzy linguistic label. Since traditional Mamdani- or Sugeno-type fuzzy logic-based models cannot tune their membership parameters, they suffer from low performance when there is a lack in expert knowledge, when they are applied on noisy observation data or when they are applied on observations from another spatial context. On the other hand, the neural network-based models developed by [12,13] and others can adjust well in a varied condition. Hence, in order to overcome the individual limitations of both the fuzzy logic and machine learning (neural network) models, this paper presents a novel and a more robust and expressive *hybrid* framework by integrating both fuzzy and neural network-based approaches into a *neuro-fuzzy system*. The hybrid approach has been successfully used in other contexts already, such as in traffic modelling, different transportation control systems and people's mode choice behaviour.

For example, Panella and colleagues developed a neuro-fuzzy model to address vehicular traffic flow in an urban environment. They developed a centralized system where the vehicular movement data are transmitted and based on the kinematics, a particular flow state is determined. They used a hyper-plane clustering technique in the training stage [31]. Neuro-fuzzy systems have been also used in traffic control in different types of intersections. Henry and colleagues show a neuro-fuzzy model working satisfactorily at intersections with simple and medium complexity. At more complex intersections, a neuro-fuzzy model needs integration with an optimal control [32]. Wannige and colleagues developed a neuro-fuzzy-based traffic control system in their simulated study. They used two four-way traffic junctions and a road connecting both the junctions. They have investigated how traffic behaviour changes on that particular road segment between the two junctions and how the traffic system adapts with varying conditions. Their study shows that a neuro-fuzzy logic-based traffic control system works better than a fixed-time signal control system. The model also minimizes the delay time significantly during red light phases at each junction. Wannige and colleagues also showed how traffic lights at both of the junctions synchronize adaptively when the volume of traffic increases significantly at one of the two junctions [33]. In a slightly different work, Dell'Orco and colleagues developed a neuro-fuzzy model to predict users' decisions in transport mode choice [34]. Their assumption is based on the uncertainty and imprecision in data in an urban environment. Using simple fuzzy rules, they have demonstrated how users' perception can be encoded in linguistic attributes. The model performs better in forecasting users' mode choice behaviour than that of a random utility-based model.

However, neuro-fuzzy models have not been used for travel mode detection yet. Due to the particularities of travel modes, the presented model will have some distinctive properties, which are discussed in the subsequent sections.

3. Theory

In this section, the basic definitions and concepts are presented related to trajectories and the model architecture. Trajectories are defined in this paper assuming the space-time points are captured using a positioning sensor.

3.1. Trajectories

A trajectory (T_r) is a sequence of time ordered spatio-temporal points (P_i) that represents an individual's movement behaviour over a given space and time in terms of coordinates (x_i, y_i, z_i) in a three-dimensional Euclidean space at a given time stamp (t_i). However, for the study of travel modes,

the z value can be ignored. Trajectories can be recorded by different sources, such as checkpoints or cordons [35], portable GPS data loggers [36], GPS sensors on-board of smartphones [8], CCTV [37], or smartcards [38], to name a few. A trajectory can be expressed mathematically as follows:

$$T_r = \{P_i\} : P_i = (x_i, y_i, z_i, t_i) | \forall i : t_{i-1} < t_i \quad (1)$$

A (raw) trajectory is transformed to a semantic trajectory by incorporating infrastructure information and domain knowledge at a given context.

3.2. Near-Real Time Mode Detection

In contrast to a real-time mode detection, this research deals with a near-real time mode detection. The difference between the two concepts lies in the delay in response time (Vis-à-Vis in inference strategy). For a real-time detection the location information is pinged on a second-by-second basis or at a very fine granularity. However, the commercial Android-based smartphones suffer from battery drainage on heavy usage of GPS. In addition to that, in an urban environment, the GPS trajectory involves frequent signal gaps and multipath effects, which make a fine-grained sample unreliable for detecting the modes. On the other hand, in contrast to body part movement in the context of activity recognition in pervasive and mobile computing [21], the transport mode does not change so frequently within a few seconds, and thus, a comparatively more coarser-grained time window containing more than one piece of GPS location information is deemed to be useful for detecting modes in the interest of close to real-time information retrieval for various mobility-based service provisions. Figure 1a shows a real-time mode detection concept where the smartphone continuously pings its location information to a central server on a second-by-second basis (or at an interval set by the sampling frequency), whereas Figure 1b shows for a near-real time scenario a shorter sequence of GPS points being sent to the central server over a given time window containing richer kinematic information for mode detection.

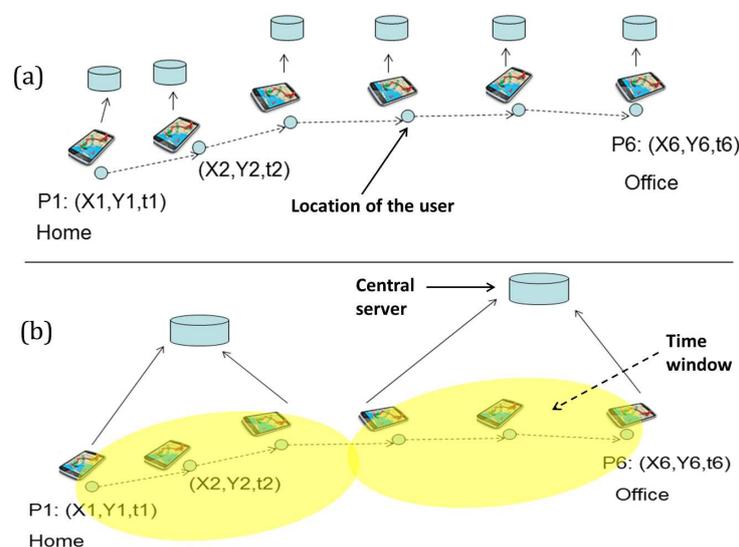


Figure 1. This figure illustrates how the location information is pinged at a real time scenario (a) and at a near-real-time scenario (b), while travelling from home to office.

3.3. Fuzzy Expert System

A fuzzy expert system is based on fuzzy set theory [39]. Unlike a crisp set theory where an element is either present or absent in a given set, fuzzy set theory assigns a membership value to an element and, thus, introduces the concept of a partial membership of that element in a number of different set(s). If A is a fuzzy set defined on a universe of discourse U , then the membership of an element

y in A can be defined by a membership function (MF) $\mu_A(y)$ within an interval of $[0, 1]$. This can be mathematically expressed as follows.

$$A = \{(y, \mu_A(y) | y \in U), \mu_A(y) : U \rightarrow [0, 1]\} \quad (2)$$

A fuzzy variable is expressed through a fuzzy set, which is attributed with a set of fuzzy values. Thus, a fuzzy variable A can be characterized by a set of fuzzy values, which is known as term set $\{T\}$, and a set of membership functions $\{M\}$, where:

$$T_A(y) = T_y^1, T_y^2, \dots, T_y^k \quad (3)$$

$$M_A(y) = \mu_y^1, \mu_y^2, \dots, \mu_y^k \quad (4)$$

There are two types of fuzzy models used. These are the Mamdani fuzzy model and the Takagi–Sugeno–Kang (TSK) fuzzy model, more popularly known as the Sugeno fuzzy model. In a Mamdani fuzzy model, both the antecedent (**IF**) and consequent (**THEN**) both are fuzzy. The **IF** part contains the fact, and the **THEN** part contains the conclusion. In the case of a Mamdani fuzzy approach, both the fact and the conclusion both are not certain, which occurs in most of the real-life situations due to limitations in system architecture, data acquisition, the varied level of perception of the user, the quality of data and the predicted outcome in a given context. A multiple input single output (MISO) Mamdani fuzzy rule can be represented as follows.

R_{m1} : **IF** avg_speed is high and avg_acceleration is uniform, **THEN** delay_time is low.

In contrast, a Sugeno fuzzy model involves a fuzzy antecedent and a crisp consequent part, which is generally expressed in terms of a polynomial function of order 'n'. A Sugeno fuzzy rule can be represented as follows.

R_{s1} : **IF** avg_speed is high and avg_acceleration is uniform, **THEN** delay_time is 10 sec.

While developing the rule base, each rule consists of a number of facts, which are combined using a t -norm or t -conorm or a negation operator. In this research the rule bases developed are all based on a t -norm operator. On firing a given rule, a MIN operator is used to select the minimum membership value from a number of fuzzy antecedent variables (A_i) to obtain the corresponding consequent (C_i) lamina in the consequent function. Once all of the rules are fired, all of the selected consequent lamina are aggregated; in order to generate the final output (crisp) value from the combined consequent lamina, which corresponds to the center of gravity (cg) of the combined consequent lamina. Figure 2 illustrates how each rule is fired, and the consequent lamina are combined once all of the rules are fired. In this figure (Figure 2), two rules are shown where the rules state,

R_1 : **IF** A_1 is T_1 and A_2 is T_2 , **THEN** C_1 is T_1 .

R_2 : **IF** A_1 is T_1 and A_2 is T_3 , **THEN** C_2 is T_2 .

For given inputs when fuzzy variable $A_1 = y_1$ and $A_2 = y_2$, each rule is fired, and the corresponding fuzzy consequent is inferred. In order to defuzzify the consequent part and to obtain the final output (F_0) a "center of gravity" method is used as follows.

$$F_0 = \frac{\sum u_i \cdot y_i}{\sum u_i} \quad (5)$$

Unlike Mamdani fuzzy model, in the case of a first order Sugeno fuzzy model, on firing each, rule the consequent part takes on a crisp value in terms of a number of coefficients (p, q, r) based on a given function. For example, in the previous example, in Rule 1, when $A_1 = y_1$ and $A_2 = y_2$, the output $C_1 = f(y_1, y_2)$, where:

$$f(y_1, y_2) = py_1 + qy_2 + r \quad (6)$$

Each rule (R_i) weighs its output by a firing strength w_i . Once all of the rules are fired, a weighted average is used to generate F_0 for a given Sugeno model.

$$F_0 = \frac{\sum w_i \cdot C_i}{\sum w_i} \tag{7}$$

In the case of a zero order Sugeno model, p and q essentially become zero. Both the conventional Mamdani and Sugeno fuzzy models are dependent on proper rule base and membership functions. Often, it is difficult to choose a proper membership function along with its characteristic parameters for a given fuzzy set. Fuzzy expert systems also cannot learn in varying conditions and need a human expert intervention for modification. In order to select the membership function parameters automatically and in turn construct the rule base, a hybrid knowledge driven technique, such as an adaptive neuro-fuzzy inference system (ANFIS), is required.

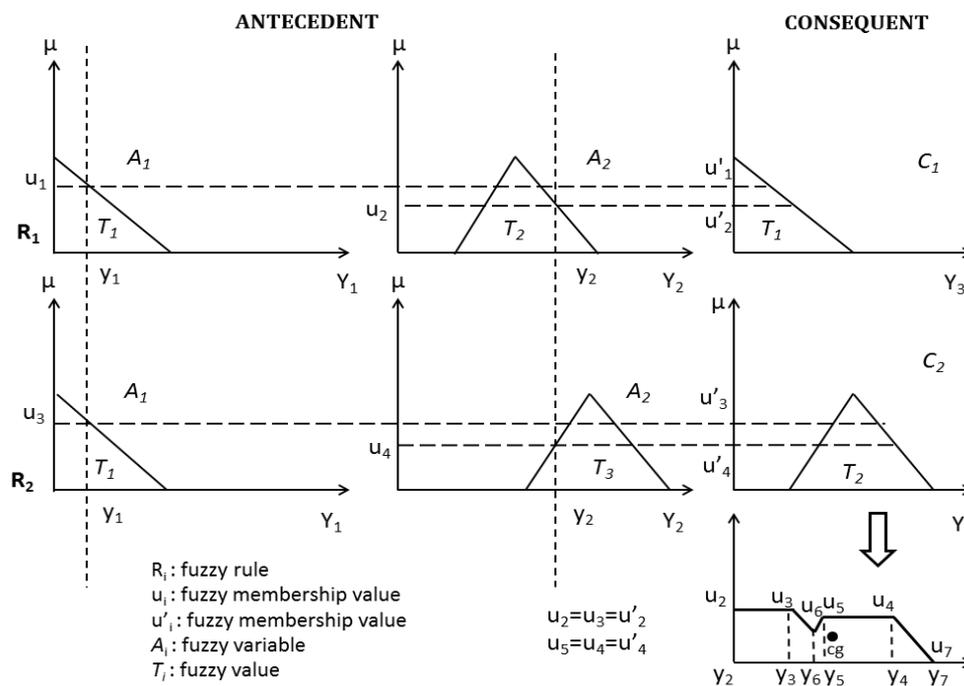


Figure 2. A Mamdani fuzzy inference system with two rules.

3.4. Adaptive Neuro-Fuzzy Inference System

An adaptive neuro-fuzzy inference system (ANFIS) is a neuro-fuzzy-based hybrid model that is equivalent to a Sugeno fuzzy model by its operation and reasoning process, whereas it is equivalent to a neural network (with a connectionist structure) by its architecture and learning ability [40]. ANFIS requires a training phase that initializes the knowledge base with a set of rules and membership functions with automatically-selected function parameters. The training takes place through a number of iterations. A standard ANFIS model follows a hybrid learning using a forward and backward pass [41]. An ANFIS model consists of five layers.

Layer 1 is a fuzzification layer. The inputs are fuzzified in this layer based on the respective membership functions. In Figure 3, the nodes A_i and B_i are linguistic values of input x and y , respectively. The parameters involved in the given membership function are called antecedent parameters. The nodes in Layer 1 are adaptive nodes in the sense that the nodes will keep on changing the antecedent parameters during the training stage to achieve minimum errors. Layer 2 contains the rule base with a t -norm operator which is generally considered as equivalent to a MIN or

a product operator [42]. The nodes in Layer 2 are all fixed nodes. Each node in Layer 2 emits a firing strength (w_{li}) of the corresponding rule, where (w_{li}) can be expressed as:

$$w_{li} = \prod_i^V \{\mu_A(x_i)\} \tag{8}$$

where $\mu_A(x_i)$ is the membership function of fuzzy set A for a linguistic variable i for a given rule r , assuming the total number of linguistic variables is V . The firing strengths are then normalized by the nodes in Layer 3 as follows:

$$s_r = \frac{w_{li}}{\sum_l^n w_{li}} = \frac{\prod_i^V \{\mu_A(x_i)\}}{\sum_l^n \prod_i^V \{\mu_A(x_i)\}} \tag{9}$$

where l is the layer number, and r is the node number in a given layer and n is the total number of nodes in Layer l .

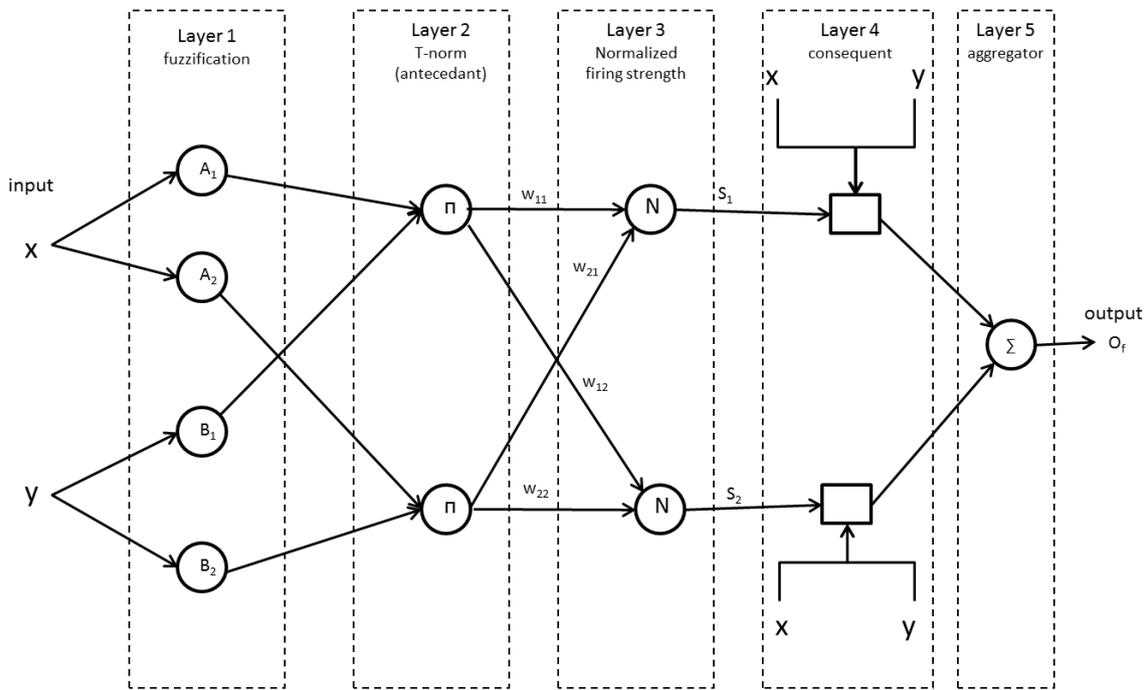


Figure 3. A basic adaptive neuro-fuzzy inference system (ANFIS) architecture.

Layer 4 computes the consequent part of each rule based on the firing strength. An ANFIS model is based on a Sugeno architecture for computing its consequent part. A first order Sugeno model computes a consequent part as follows:

$$O_r = s_r(a_r x + b_r y + p) \tag{10}$$

where O_r is an output in a consequent part for rule r , s_r is the normalized firing strength and a_r, b_r and p are consequent parameters. Layer 5 aggregates all of the individual consequent parts from of the respective rules and defuzzifies to generate the overall output (O_f):

$$O_f = \sum s_r f_r = \frac{\sum_r^n s_r (a_r x + b_r y + p)}{\sum_r^n s_r} \tag{11}$$

In the case of a zero-order Sugeno model, the consequent part simplifies into p . The consequent parameters are tuned in a forward pass using a least square estimation where the error term (E) can be expressed as:

$$E_k(a, b) = \sum_{t=1}^N (T_k - (a_k x + b_k y + p))^2 \quad (12)$$

where $E_k(a, b)$ is the error term for the k – th entry in the training data, T_k is the target output for the k – th entry, and N is the total number of iterations. Thus, the overall error is:

$$E = \sum E_k \quad (13)$$

The objective is to minimize $E_k(a, b)$, and hence, the objective functions can be mathematically expressed as

$$\frac{\partial(E_k(a, b))}{\partial a} = 0, \frac{\partial(E_k(a, b))}{\partial b} = 0 \quad (14)$$

In order to determine the antecedent and consequent parameters a hybrid back propagation technique is used. The consequent parameters, are determined through a least square estimation in a forward pass, whereas the antecedent parameters are determined using a gradient descent technique in backward pass. The rules can be generated in one of three ways: grid partitioning, subtractive clustering or fuzzy c -means clustering (FCM). In this paper, a grid partitioning technique has been used to search the entire input space and generate all of the possible rules. Hence, if V is the number of linguistic variables, and m the number of linguistic values for each variable then the total number of rules n is:

$$n = V^m \quad (15)$$

4. Knowledge Driven Frameworks for Near-Real Time Transport Mode Detection

In this section, two fuzzy logic-based knowledge-driven models are developed. In the first framework, a MIMO Mamdani fuzzy inference system (MFIS) is developed, which is based on a priori expert knowledge (without any training). In the second framework, a hybrid knowledge-driven model is developed using a neuro-fuzzy approach.

4.1. Framework 1: Multiple Input Multiple Output Mamdani Fuzzy Model

The MIMO MFIS presented in this paper consists of a fuzzy inference engine consisting of 76 fuzzy rule sets (rule base). The antecedent part contains five fuzzy variables with three fuzzy values for each of the variables (Table 1). The consequent part consists of four alternative solutions (*bus*, *train*, *tram*, *walk*) with their corresponding certainty factors (CF) ranging from 0 to 100. The rules are developed in such a way that they can handle different quality (inaccuracy level) in positional information and different kinematic behaviour shown by a given transport mode. In order to combine different facts in the antecedent and consequent part, a t -norm operator (AND) is used. The fuzzy variables in the consequent part are independent of each other; however, their certainty value (CF) depends on the rule firing and a given input feature vector. In order to defuzzify the consequent outputs, a center of gravity method is implemented. The membership functions are all selected manually. Figure 4 shows a MIMO MFIS model developed in this paper. Some of the fuzzy rules (out of 76) are as follows.

R1: IF avgSpeed is low AND maxSpeed is low AND avgBusProx is far AND avgTrainProx is far AND avgTramProx is moderate, THEN CF for walk is high AND CF for bus is low AND CF for train is low AND CF for tram is low.

R2: IF avgSpeed is moderate AND maxSpeed is moderate AND avgBusProx is near AND avgTrainProx is far AND avgTramProx is far, THEN CF for walk is low AND CF for bus is high AND CF for train is low AND CF for tram is low.

R3: IF avgSpeed is moderate AND maxSpeed is moderate AND avgBusProx is moderate AND avgTrainProx is far AND avgTramProx is moderate, THEN CF for walk is low AND CF for bus is moderate AND CF for train is low AND CF for tram is high.

R4: IF avgSpeed is high AND maxSpeed is high AND avgBusProx is far AND avgTrainProx is near AND avgTramProx is far, THEN CF for walk is low AND CF for bus is moderate AND CF for train is high AND CF for tram is moderate.

R5: IF avgSpeed is moderate AND maxSpeed is high AND avgBusProx is far AND avgTrainProx is far AND avgTramProx is far, THEN CF for walk is low AND CF for bus is high AND CF for train is low AND CF for tram is low.

Table 1. Fuzzy variables and their fuzzy values for MIMO Mamdani fuzzy inference system (MFIS).

Fuzzy Variable	Fuzzy Value
Antecedent	
Average speed (avgSpeed)	low, moderate, high
Maximum speed (maxSpeed)	
Average proximity to the bus network (avgBusProx)	near, moderate, far
Average proximity to the train network (avgTrainProx)	
Average proximity to the tram network (avgTramProx)	
Consequent	
CF for bus	low, moderate, high
CF for train	
CF for tram	
CF for walk	

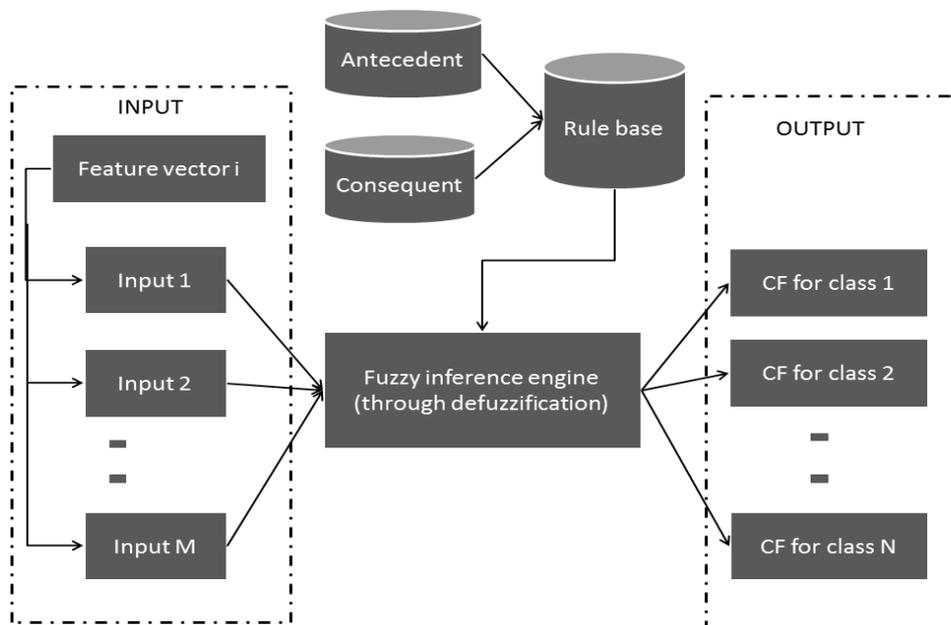


Figure 4. A MIMO MFIS model with M number of input and N number of classes with their varied certainty values. In this research, $M = 5$ and $N = 4$.

4.2. Framework 2: Multi-Layered Adaptive Neuro-Fuzzy Model (MLANFIS)

In existing transportation research and traffic control systems, ANFIS models deal with regression-type problems. In contrast, in this paper, a multi-class problem has been posed, requiring to developing a multi-layered ANFIS (MLANFIS) model in order to provide a near-real time transport mode detection framework (Figure 5). The core of the framework is a processing layer that contains a number of ANFIS modal blocks in parallel connection, where each ANFIS modal block corresponds to a given class. If there are K numbers of classes, then there will be K numbers of ANFIS modal blocks. Hence, the cardinality of the framework is K . Since each ANFIS modal block is trained in parallel without any direct connection in between them, each ANFIS modal block contains its own rule base.

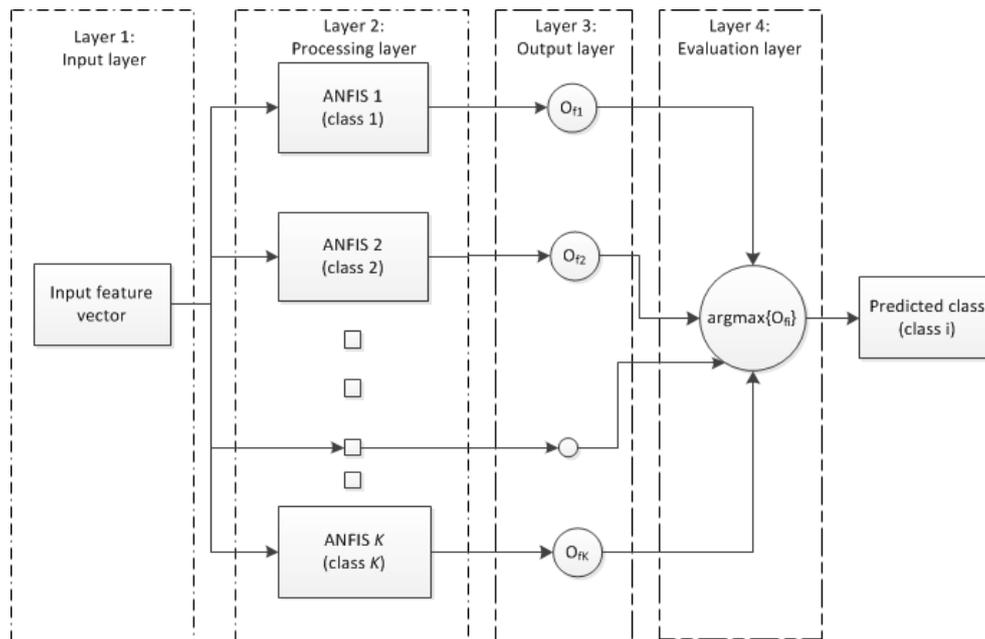


Figure 5. Multi-layered ANFIS framework for mode detection (MLANFIS).

In this paper, transport modes are categorical in a classification problem, which is not possible to deal with in a standard neuro-fuzzy approach due to its very nature of generating continuous real values. Hence, the classification problem is converted to a regression problem first, where each ANFIS modal block deals with a binary evaluation of a given modal class. An ANFIS modal block is attributed by a specific modal class (categorical value) it deals with and a level of certainty (real value) of being a given modal class. For each modal class, a separate set of training samples (training instances) and an ANFIS model are developed. In each training set, each feature vector is of a certainty factor (CF) of either zero or one, which quantifies the belongingness of that given feature vector to a given class. Hence if there are K numbers of modal classes, then there are K numbers of training sets, where each set of training samples contains the same set of feature vectors, but different output patterns. For example, the modal class *bus* contains samples in the given feature vector that are segments of a trajectory representing a bus ride, then the output is quantified in terms of a CF of one (see Section 3.3). If the feature vector is not of a bus ride, then the output CF is quantified as zero. This process is iterated for all of the instances in each training set for K modal classes. The logic behind such certainty quantification is that each ANFIS block (corresponding to a given modal class) will be trained in such a way that if it ($ANFIS^T$) is fed with a test sample ($fv_t : test\ feature\ vector$), it will assign some CF as an output through its reasoning process depending on the input feature vector. If the sample represents a given modality, it will get the maximum CF corresponding to that ANFIS modal block.

The framework consists of four layers (Figure 5). Layer 1 is the input layer, which contains the input feature vector. Layer 2 is the processing layer, which consists of trained ANFIS modal blocks

(ANFIS^T), one for each class. Layer 3 is the output layer for each ANFIS modal block. Layer 4 is the evaluation layer where all of the *CF* outputs are aggregated and evaluated using an *argmax* operator to select the maximum value. The predicted class for that given input feature vector is then determined based on the maximum *CF* generated by the respective trained ANFIS modal block. Thus, in near-real time each query is assessed in parallel in different ANFIS modal blocks, and a modal class is predicted based on the maximum *CF* value.

5. Evaluation

5.1. Data Set

In order to evaluate the hypothesis and test the model, a GPS dataset had been collected in Greater Melbourne, Australia, for 85 h, collected over three months using an application on an Android-based smartphone. The dataset contains 106 trajectories of, in total, 612,375 GPS points. The dataset covers four modalities, *bus*, *train*, *tram* and *walk*, which are four common public modalities in an urban environment (Figure 6). Unlike prompted recall surveys [43,44], the ground truth was recorded on the fly, and hence, the ground truth information is consistent and highly accurate.



Figure 6. GPS trajectory distribution in Greater Melbourne.

The data set covers modalities of similar features on different routes, as well as different modalities on overlapping routes (a portion of the bus network overlaps with the tram network). Since in this paper, a near-real time mode detection is performed, i.e., no prior segmentation can be produced, there is a possibility that within any given time window, two modalities may exist together. In this

case, it is assumed that always one of them is walking, as only a walk connects between two different non-walking modalities. For this to hold always true, the extent of the time window must be chosen smaller than any individual walking segment. That co-existing modes over a given time window is termed as a *composite* mode. However, from observation within a shorter temporal window (say 60 s to 120 s) there could be a maximum of two co-existing modes, one of which must be *walk*. Hence, all of the *composite* modes in this research are labelled as *walk*.

5.2. Pre-Processing and Feature Preparation

Before generating the feature vectors, each trajectory is pre-processed. A pre-processing stage involves filtering a trajectory based on positional accuracy, where any GPS point with positional accuracy <40 m (i.e., the major axis of the confidence ellipse is >40 m) is considered as noise and eliminated from the trajectory. The raw GPS data were collected in WGS84 coordinates. In order to perform spatial analysis, the dataset was projected onto the GDA94 coordinate system followed by feature computation.

In this framework, five features are computed: *average speed*, *maximum speed* (which is actually 95th percentile of *maximum speed*), *average proximity to bus network*, *average proximity to tram network*, and *average proximity to train network*. Since walking can take place anywhere (say close to a bus route or a street or a train network during transfer) the nearness to the street network is not utilized in this research. Proximity values are computed using a spatial buffer of 40 m (assuming standard GPS positional accuracy in this research) of each GPS point to its nearest bus network, train network and tram network. In case there is a network absent within a 40 m radius, the proximity value to that network from a given GPS point is assigned as 100 m to avoid a null value, or zero proximity. The data set is split up into training, checking and testing data sets. The trajectories selected as the training data set are of higher travel time duration than the trajectories used to generate checking and testing data set, and hence the number of features for training is always higher than the checking (and testing data) in all the experimental setups (Table 2). After training the four ANFIS modal blocks, each of them generates 243 distinct fuzzy rules.

5.3. Experiment

Five sets of experimental setups are designed based on growing time window size starting from 30 s, 40 s, 50 s, 60 s and 120 s. In order to compare the performance of the proposed framework (MLANFIS) a number of machine learning models are also developed based on a multi-layered perceptron neural network (MLP), a radial basis function-based neural network (RBF), a decision tree (DT), K-nearest neighbor (KNN), and a naive Bayes (NB). The result shows at a 60-s and a 120-s time window that MLANFIS yields significant accuracy for detecting different transport modes in near-real time.

In order to evaluate, the same training and testing data have been used for the MLANFIS model and all of the machine learning models. Since MFIS does not require to be trained, hence an MFIS model evaluated using only a testing dataset, which has been used to test the predictive ability of MLANFIS and the machine learning models. A checking dataset is used while building the MLANFIS model in order to make sure the model does not get over-fitted. Table 2 shows the number of features used as training, checking and testing datasets for different models. Figure 7 shows how checking error and training error vary with the number of iterations (epochs). A total of 200 iterations are performed for each MLANFIS modal block building. A training error shows a gradual decrease in magnitude over 200 iterations. On the other hand, the checking error shows a gradual decrease in magnitude up to a certain epoch followed by a sudden increase in magnitude. That critical epoch point indicates the moment when the model starts getting over fitted. The membership function parameters are selected at that particular given epoch before the checking error gets increased.

In order to measure the accuracy of the models, precision accuracy, and recall accuracy are used, which are based on true positives (tp), false positives (fp), true negatives (tn), and false negatives (fn). The formula for precision and recall accuracy are provided as follows:

$$precision = \frac{tp}{tp + fp} \quad (16)$$

$$recall = \frac{tp}{tp + fn} \quad (17)$$

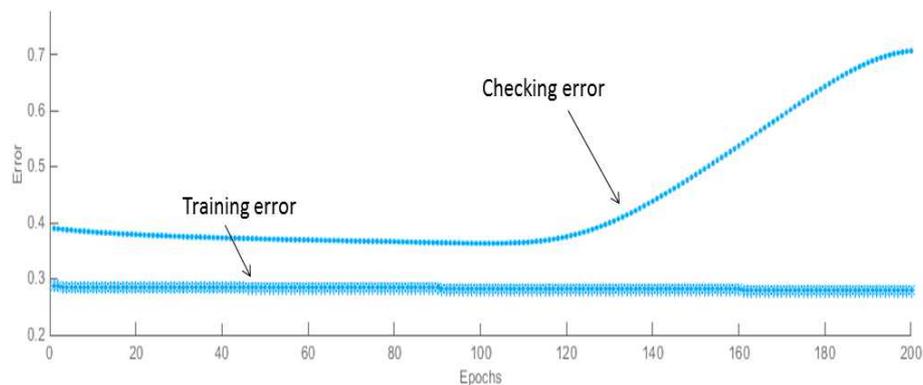


Figure 7. An over-fitting in *walk* modal block in MLANFIS.

Table 2. Number of features used for training, checking and testing.

Time Window (s)	Features Type	MLANFIS	Machine Learning	MFIS
30	Training	21,279	21,279	-
	Checking	13,459	-	-
	Testing	10,099	10,099	10,099
40	Training	15,665	15,665	-
	Checking	9894	-	-
	Testing	7433	7433	7433
50	Training	12,390	12,390	-
	Checking	7814	-	-
	Testing	5876	5876	5876
60	Training	10,243	10,243	-
	Checking	6456	-	-
	Testing	4859	4859	4859
120	Training	5011	5011	-
	Checking	3132	-	-
	Testing	2371	2371	2371

Tables 3 and 4 show recall and precision accuracy of seven different predictive models, including an MLANFIS and MFIS at 60-s time window. In terms of recall accuracy, MLANFIS outperforms the MFIS model and performs on par with the machine learning models for *walk*, *train*, *tram* mode. However, MLANFIS works poor in terms of recall accuracy for *bus* when compared to the machine learning models. On the other hand, the MFIS model performs better than MLANFIS and other machine learning models in terms of precision accuracy, particularly for *train* (96.86%) and *tram* (87.91%). MLANFIS works best and very close to an RBF model in terms of precision accuracy for *bus* (92.19%). This suggests the rules generated for bus ANFIS block in MLANFIS model are properly tuned and thus giving rise to less Type I error for *bus* when evaluated by a MLANFIS. However the

rules in the *bus* ANFIS block are not sufficient enough to capture all of the kinematic behaviour and signal quality during a bus ride, and hence, although MLANFIS generates less Type I error, but higher Type II error for *bus*, that led to low recall accuracy for *bus* mode, when compared with the machine learning models. Since different predictive models perform differently for different modes in terms of precision and recall, hence in order evaluate the overall performance of the models, an F1-score (F) is considered, which combines the precision and recall together.

$$F = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \quad (18)$$

In terms of F1-score, MLANFIS performs similarly as MLP and DT for walk mode detection and outperforms a MFIS and all other machine learning models (Figure 8). MLANFIS outperforms all other models for train mode detection. For train mode detection, MLANFIS yields 0.91 F1-score followed by 0.88 by MLP, which is the highest F1-score generated by any machine learning model. For tram mode, MLANFIS yields 0.82, which is very close to MLP, which yields 0.84, and a DT model, which generates a 0.81 F1-score. However, for bus mode detection, MLANFIS generates 0.76 F1-score, which is less than the machine learning models, but higher than the MFIS model (Figure 8).

Table 3. Recall accuracy (%) at a 60-s time window.

Mode	MLANFIS	MFIS	RBF	MLP	NB	KNN	DT
Walk	92.58	93.47	85.60	91.10	83.80	88.60	89.30
Bus	65.21	61.20	69.60	77.60	74.90	77.30	74.70
Train	93.33	61.77	93.80	93.30	91.30	88.60	95.30
Tram	84.16	60.06	79.30	85.10	89.80	79.50	82.60

Table 4. Precision accuracy (%) at a 60-s time window.

Mode	MLANFIS	MFIS	RBF	MLP	NB	KNN	DT
Walk	88.94	80.33	88.40	91.90	92.80	90.10	91.30
Bus	92.19	65.12	92.00	85.00	87.50	80.10	85.60
Train	89.74	96.86	73.10	80.90	85.80	82.20	77.40
Tram	80.78	87.91	70.30	84.10	64.80	77.20	79.40

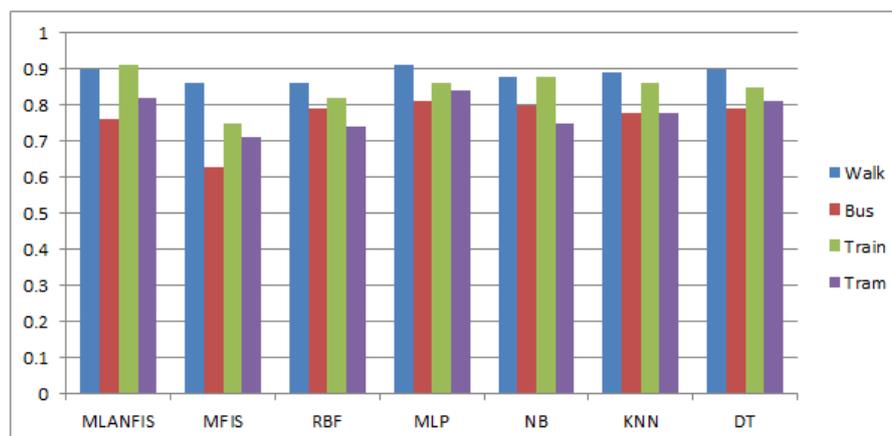


Figure 8. F1-score at a 60-s time window.

When evaluated within a 120-s time window, MLANFIS shows the same pattern in terms of recall and precision accuracy, as well as the F1-score. MLANFIS yields the highest recall accuracy

for *walk* mode, which is 92.87%, seconded by MFIS and DT, which are approximately 91.4%. For *train* mode detection, RBF yields the highest recall accuracy, which is 99.10%, whereas an MLANFIS generates 94.31% accuracy. However, an MFIS generates 74.40% accuracy for *train* mode detection showing worse performance than MLANFIS and the machine learning models. MFIS also performs poor compared to MLANFIS and the machine learning models in terms of recall accuracy for *bus* and *tram* mode detection. In terms of precision accuracy for *train*, MFIS works best, generating 94.57% accuracy, followed by MLANFIS, which generates 89.23% accuracy, whereas the highest precision accuracy was generated by the machine learning model (NB in this case), which is 87.70% (Table 5). However, in terms of F1-score, MLANFIS outperforms all of the predictive models for *train* mode detection, whereas it works on par with the machine learning models (and outperforming a MFIS) for *walk* mode, detection (Figure 9). For *tram* mode MLANFIS yields 0.84, which is very close to MLP (0.86) and DT (0.83) and outperforms MFIS (0.74), RBF (0.78), NB (0.76) and KNN (0.80).

Table 5. Precision accuracy (%) at a 120-s time window.

Mode	MLANFIS	MFIS	RBF	MLP	NB	KNN	DT
Walk	92.05	86.12	94.10	96.10	95.80	92.30	93.50
Bus	85.93	66.14	88.10	85.60	86.80	80.40	88.10
Train	89.23	94.57	73.90	76.50	87.70	77.70	83.90
Tram	84.08	83.13	75.10	84.00	66.50	79.80	82.00

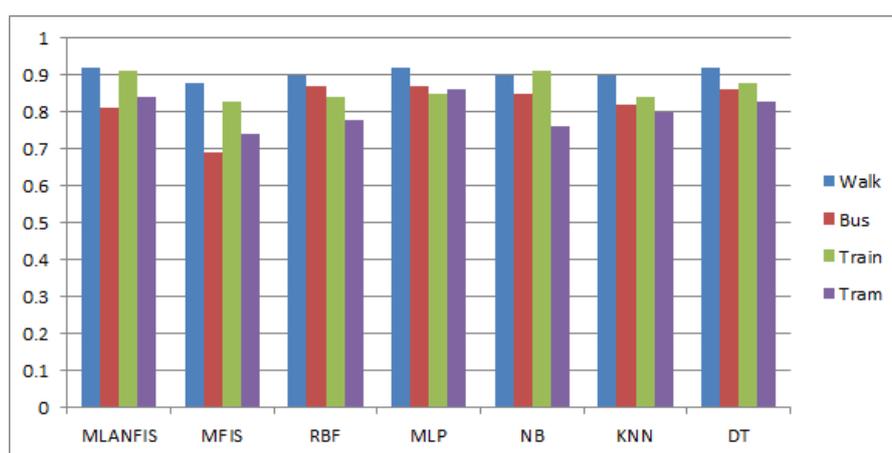


Figure 9. F1-score at a 120-s time window.

When a comparison is made only between two different types of knowledge driven models (e.g., MLANFIS and MFIS), the results suggests MLANFIS performs better than MFIS (Figures 8 and 9). For a 60-s time window MFIS generates high Type II error for *bus*, *train* and *tram* mode compared to a MLANFIS. Thus a MFIS shows a drop in recall accuracy for different public transport modes except *walk* (Table 3). However a MFIS model yields higher precision accuracy for *train* and *tram* mode (Table 4) than that of the MLANFIS model, whereas MFIS performs worse compared to MLANFIS in terms of *bus* and *walk* mode detection. This can be justified as due to the particularities in rule base to capture the different kinematic behaviour in the MFIS model typically at a low speed condition, and *near to moderate* proximity to the tram network or train network, some portion of the actual tram or train trip is detected as *walk*. However, most of the retrieved *tram* and *train* instances are correctly detected owing to high precision accuracy in *train* and *tram* mode detection. The MFIS rule also does not work well when there is an overlap between tram network and a bus network. A MLANFIS can typically work better than the MFIS model in such ambiguous situations and shows an overall better

performance than that of the MFIS model (Figure 8). Some of the fuzzy rules (out of 243) generated by the MLANFIS bus modal block are as follows:

R1: **IF** avgSpeed is low AND maxSpeed is low AND avgBusProx is low AND avgTrainProx is low AND avgTramProx is low, **Then** CF for Bus is out1mf1;

R2: **IF** avgSpeed is low AND maxSpeed is low AND avgBusProx is low AND avgTrainProx is low AND avgTramProx is moderate, **THEN** CF for Bus is out1mf2;

Where outimfj is the CF value for the i^{th} consequent part for j^{th} fuzzy rule.

Table 6 shows a confusion matrix for MLANFIS at a 60-s time window. The confusion matrix illustrates that most of the Type II error for non-walk modes are misclassified as *walk*, and that happened during signal loss or typically at a low speed condition. This suggests a more rigorous rule formation by incorporating more sensor information, such as an accelerometer.

Table 6. Confusion matrix for MLANFIS at a 60-s time window.

Actual	Predicted			
	Walk	Bus	Train	Tram
Walk	2711	19	37	161
Bus	192	390	1	15
Train	25	4	420	1
Tram	120	10	10	744

The MLANFIS framework developed in this paper can also produce alternate solutions with varied degrees of confidence. For a given feature vector where the average speed is 64.6 km/h, the maximum speed is 73.9 km/h, the average proximity to bus network is 88.4 m, the average proximity to train network is 7.15 m, and average proximity to the tram network is 88.4 m, MLANFIS produced a certainty factor for being a *train* as 0.782 (Figure 10a) and for being a *bus* as 0.106 (Figure 10b). Due to the space limitations, Figure 10 shows only 29 rules out of 243 rules for each *train* and *bus* ANFIS modal block. This also explains the explanatory power and multiple output possibility from the proposed MLANFIS framework, which is missing in machine learning models.

Since choosing the appropriate membership function is important while developing a knowledge driven model, hence two different fuzzy membership functions such as a Trapezoidal function and a Gaussian function are tested while developing MLANFIS and MFIS models. However due to crisp geometrical nature of Trapezoidal function, there are cases when an input feature may fall outside a given range of fuzzy membership function and thus may bear a zero membership value owing to low performance in its predictive process. On the other hand since a Gaussian function is asymptotic in nature, it guarantees to generate a certain membership value μ always in the range of $[m, 1]$ where $\lim_{m \rightarrow 0} \mu = 0$. A trapezoidal membership function is characterized by four characteristic points (upper left, upper right, lower left and lower right), whereas a Gaussian membership function is characterized by only two characteristic parameters such as the center (c) and the width (σ). Table 7 shows different parameters for MLANFIS which are selected automatically based on a hybrid learning involving a gradient descent and least square estimation whereas the parameters for MFIS chosen manually resulting higher ambiguity and low performance in near-real time scenario. Figures 11 and 12 show two sets of three different Gaussian membership functions for average proximity to the train network in MLANFIS and MFIS respectively. Figure 13 shows how the certainty factor changes with two different fuzzy variables. The figure shows a prominent contrast between change in CF for a *bus* and *train* when considering the same fuzzy variables such as average proximity to the bus network and average speed (Figure 13a,b). Since walking can take place anywhere hence in this research nearness to the street network is not used as the streets in Melbourne show a significant overlap with the tram and bus network. Thus in order to detect the walking mainly a low speed behavior is considered (Figure 13d).

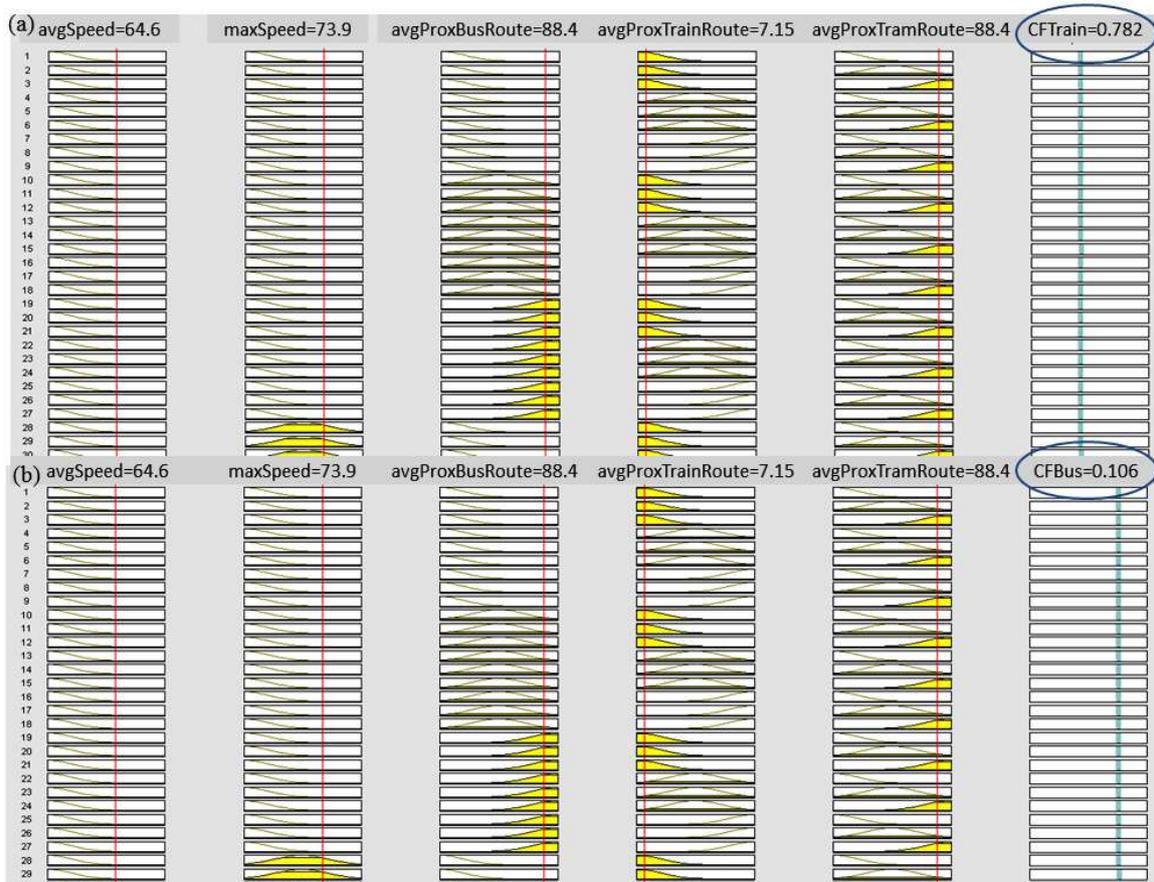


Figure 10. Certainty factors for train (a) and bus (b) for a given feature vector.

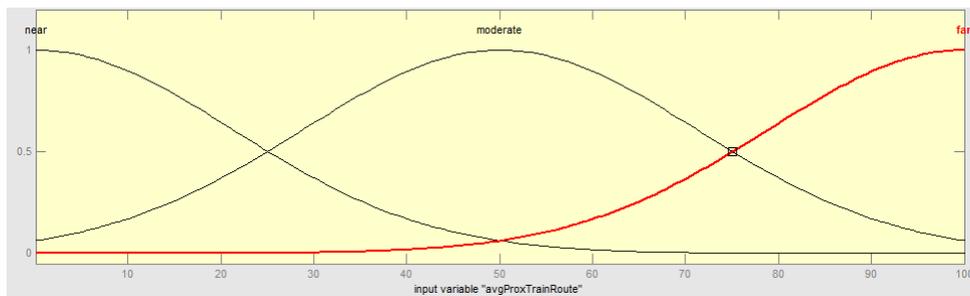


Figure 11. Fuzzy membership functions for average proximity to the train network in MLANFIS.

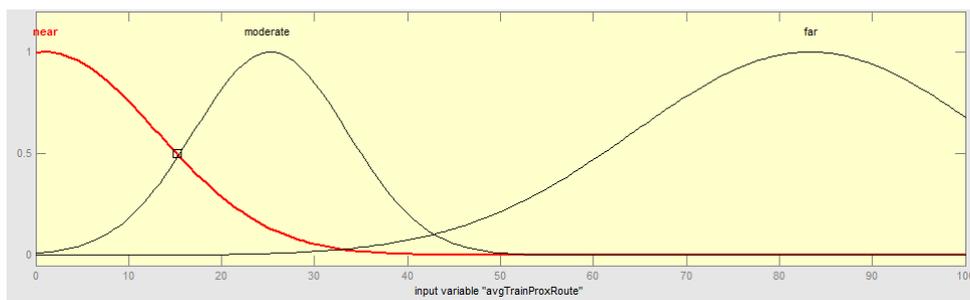


Figure 12. Fuzzy membership functions for average proximity to the train network in MFIS.

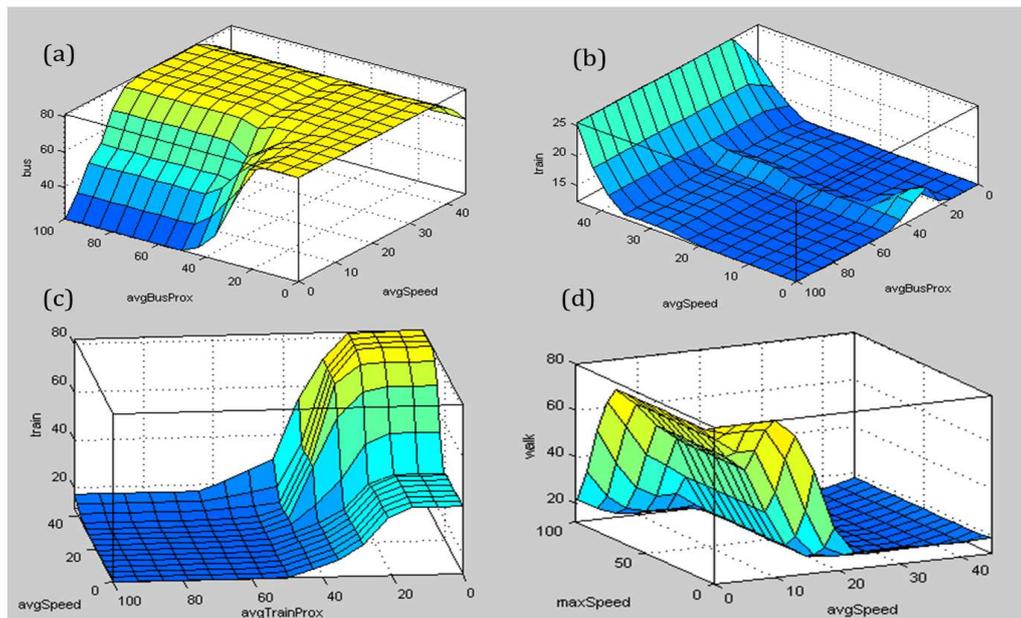


Figure 13. illustrates how CF for a given class changes with any two different fuzzy variables. z-axis indicates CF value whereas xy plane indicates fuzzy variable space. The figure shows how CF for bus changes with change in average proximity to the bus network and average speed (a); The figure also shows how CF changes for a train mode when considering the same fuzzy variables that is average proximity to the bus network and average speed (b); followed by the CF for train with average speed and average proximity to the train network (c); A change in CF for walk is shown with change in average speed and maximum speed (d).

Table 7. Different parameters for MLANFIS and MFIS for a Gaussian function at a 60-s time window.

Fuzzy Variable	Fuzzy Value	MLANFIS			MFIS		
		σ	c	Range	σ	c	Range
avgSpeed (km/h)	low	23.32	−0.21		5.30	0.21	
	moderate	23.68	55.29	0–110.9	7.81	21.9	0–45
	high	24.30	110.6		5.54	44.00	
maxSpeed (km/h)	low	23.29	−0.48		13.19	2.65	
	moderate	24.26	54.17	0–110.9	14.74	45.90	0–100
	high	24.26	109.5		13.20	97.93	
avgProxBusRoute (m)	near	20.73	−0.3		10.59	1.02	
	moderate	20.96	49.92	0.03–100	8.28	25.25	0–100
	far	21.36	99.91		20.33	83.80	
avgProxTrainRoute (m)	near	21.17	0.19		12.04	1.02	
	moderate	21.3	50.03	0.06–100	8.28	25.25	0–100
	far	21.20	100		18.99	83.30	
avgProxTramRoute (m)	near	21.14	0		11.23	1.02	
	moderate	21.02	49.95	0.03–100	7.36	25.30	0–100
	far	21.29	99.93		25.15	83.30	

For trapezoidal membership function, the recall accuracy at the 60-s time window for MLANFIS and MFIS drops significantly. For MLANFIS, for *walk*, recall accuracy drops from 92.58% down to 89.31%, for *bus* accuracy, drops from 65.21% down to 57.52%; for *train*, from 93.33% down to 88%; for *tram* accuracy; down from 88.94% down to 85.42%. For MFIS, the drop is more prominent. For MFIS, recall accuracy for *bus* drops from 61.20% down to 51.67%; for *train*, it drops from 61.77% down to

40.22%, and for *tram* the accuracy drops from 60.06% down to 35.74%. Thus, the result suggests that a Gaussian function is better than a trapezoidal membership function for near-real time mode detection using fuzzy logic-based knowledge-driven models. The results also suggest a hybrid neuro-fuzzy (MLANFIS) works better than a purely knowledge-driven fuzzy logic-based MFIS model and performs on par with some of the state of the art machine learning models, and even sometimes outperforms them for many places (Figure 8).

6. Discussion

Transport mode classification is an emerging research problem approached by different research communities. In this paper we have introduced the concept of a near-real time transport mode detection. We have developed a multi-layered neuro-fuzzy based model (MLANFIS). In order to choose the optimal temporal window in near-real time, five sets of experiments were performed. Based on the results a 60-s time window is selected as an optimal window which can generate satisfactory accuracy. However deciding an optimal temporal window is subjective and may vary from one service domain to another. For example, a traffic management organization may accept a longer temporal window (>120 s) if the main objective is to understand mode preference and patronage over a given route type (say, train route) assuming the downside that, there may be some quick transfers with in 2 min which may be missed by the proposed model when evaluate over a longer time window.

On the other hand for an emergency service provider or location-based e-marketing organization a shorter time window (≤ 120 s) is required since the main focus is to communicate with the user in awareness of their current travel mode (say, a gas station wants to advertise some discounted gas coupons to all private cars within 1 km). The shorter temporal window is necessary for all context-aware systems that relate to the current travel modality (say, auto-answering an incoming phone call while the called person is driving). Compared to the ANN model by Byon and colleagues who used longer time windows (in the order of 5 min and 10 min) [12], this paper is an improvement allowing shorter time windows of 1 min or 2 min using GPS only samples and infrastructure information.

However, by using GPS only samples it is not feasible to get a shorter temporal window than that of indoor activity recognition due to hardware and software limitations of the sensing system (and also to preserve the battery). Table 6 shows the accuracy of MLANFIS drops mainly due to the fact that all the non-walk modes are most of the times misclassified as *walk* mode during signal loss or at a low speed condition, which can be resolved in the future by integrating different inertial sensors, which can sense at significantly higher sampling rates than a GPS sensor on board of smartphones.

MLANFIS shows a performance improvement for some of the modes on increasing the time window in particular for *walking*, and *tram*. The model also demonstrates different accuracy while choosing different membership functions. This research also shows how knowledge driven (MFIS) and hybrid knowledge driven model (MLANFIS) can explain their reasoning scheme unlike conventional machine learning models.

However the success of MFIS depends on the number of fuzzy rules and their relevance. The success of MFIS also lies in proper membership functions and their shape, which can be automatically handled by a MLANFIS. However, the MLANFIS model developed in this paper is based on grid partitioning, which exhaustively searches the entire input space. Thus increasing the number of features (fuzzy variables) along with their term set will also increase the number of rules, which touches on the “curse of dimensionality” [41]. Although in this paper complexity associated with the models is not addressed, in general a grid partitioning suffers from higher temporal complexity and memory usage. This issue can be addressed in more complex hybrid models by adopting a subtractive clustering or fuzzy *c*-means clustering (FCM) approach.

7. Conclusions

In this paper, we have addressed the challenges of detecting transport modes in near-real time particularly for real time travel demand estimation in the interest of public transport authorities and different context-aware service provisions. This paper presents a neuro-fuzzy based hybrid knowledge driven framework for an inference system in the context of urban mobility. Since this research is focused on a near-real time approach, there is no need to segment the trajectories like the existing practice in transport mode detection on historical trajectories; and thus this approach will reduce the computational overhead and response time. To the best of the authors' knowledge this is the first work where a hybrid, multi-layered ANFIS (MLANFIS) model is developed to address the classification problem of transport mode detection. In this paper an optimal time window is also suggested for querying in near-real time. We have also drawn a comparison in performance between a number of knowledge driven models and a number of machine learning models.

The result shows in some cases some of the machine learning models perform well but they act like a black box and lack the capacity to explain their reasoning process. A DT based model can explain the reasoning process in a more deterministic way based on some threshold at each level which however varies in different conditions and cannot represent a generic kinematic behavior in a linguistic way for human understanding. On the other hand, MFIS is based on predefined generic rule sets which is understandable by a machine and a human, but since the process involves expert knowledge in constructing the rule base and the membership functions, a MFIS model fails in the situation, which is not explained to the model by the expert or in a situation where the expert knowledge is outdated. This problem is mitigated by the suggested multi-layered neuro-fuzzy based model with its capacity to encode knowledge through n -ary relationships through different t -norm operators and expressed in a human readable format. Thus a neuro-fuzzy model is more robust and effective than that of a fuzzy model. The results demonstrate that a neuro-fuzzy model can perform at par with machine learning algorithms for most of the modalities while outperforming a traditional fuzzy logic model (Figure 8). The hybrid model presented in this paper is capable of generating alternate possibilities with different certainty factors. The reasoning scheme can also explain the driving behavior of a person and deviation from regular behavior based on the type of rules fired, which can then trigger various mode specific context-aware service provisions. The result also demonstrates a knowledge driven approach (fuzzy and neuro-fuzzy) can also achieve a higher accuracy with a transparent reasoning scheme (Tables 3 and 4). Table 4 shows MLANFIS outperforms all the machine learning models in terms of precision accuracy for *bus* at 60 s time window. In the same line, a MFIS also outperforms the machine learning models in terms of precision accuracy for *train* and *tram*.

At 60-s time window, MLANFIS yields 83% average accuracy which is at par with a RBF, DT, and a NB model and outperforms a purely knowledge driven fuzzy model, which generates only 69% average accuracy. However an MLP based neural network model generates 87% average accuracy, which is higher than the neuro-fuzzy model developed in this paper. But at the same time the neuro-fuzzy framework developed in this paper can explain its reasoning process, which is missing in an MLP or RBF or even in a DT based model. In addition to that, a conventional fuzzy model cannot learn adaptively and thus is not robust to noise. In contrast, the presented neuro-fuzzy model can tolerate noise and adapt to varying conditions. The neuro-fuzzy model developed in this paper shows more consistent performance than that of a fuzzy logic based model in near-real time scenario. The neuro-fuzzy model is also tested against some other machine learning models (e.g., SVM) where the model shows better performance than those machine learning approaches.

The framework shows that a MLANFIS model can learn and explain its reasoning scheme, which overcomes limitations of a conventional MFIS type fuzzy expert systems developed by [18,30] as well as machine-learning models (e.g., neural network) [12,13]. In this paper four urban transport modes are used for testing the MLANFIS model, where the *train*, *tram* and the *walk* modes are detected with high accuracy, followed by *bus* mode. However the model can easily be extended for more modalities along with more input features. This may increase the ambiguity especially when two

modalities show similar movement patterns and share the same network (say, a car and a bus are moving on an express way with the same high speed). In such situations more features are required such as stop rate, heading change rate, vibration and ambient sound profile: All of these can easily be incorporated in the model.

Future research will investigate how the model behaves on integrating different sensor signals such as accelerometer, gyroscope and GPS. This integration also leads to new challenges as how to fuse sensors with their different data quality and ability to sample at different frequency. Future research will also look into how a Sugeno-based rule sets can be converted to a MIMO Mamdani fuzzy rule set where the consequent part may consist of multiple outputs expressed in natural language for a thorough knowledge representation. In the same line, future research could investigate the top-*k* most relevant rule sets for each modal block in an MLANFIS model in the context of travel mode detection.

Acknowledgments: This research has been supported by the Australian Research Council (LP120200130).

Author Contributions: Rahul Deb Das and Stephan Winter conceptualized the research idea together which is presented in this paper; Rahul Deb Das has implemented the conceptual framework, and took the lead in writing the paper; and Stephan Winter has guided throughout the process from model conceptualization to evaluation, and also contributed in writing and proof-reading.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Das, R.D.; Winter, S. A context-sensitive conceptual framework for activity modelling. *J. Spat. Inf. Sci.* **2016**, *12*, 45–85.
2. Abowd, D.; Dey, A.K.; Orr, R.; Brotherton, J. Context-awareness in wearable and ubiquitous computing. *Virtual Real.* **1998**, *3*, 200–211.
3. Wolf, J.; Schonfelder, S.; Samaga, U.; Oliveira, M.; Axhausen, K. Eighty weeks of GPS traces: Approaches to enriching trip information. *Transp. Res. Rec. J. Transp. Res. Board* **2004**, *1870*, 46–54.
4. Lowrie, P.R. *SCATS, Sydney Co-Ordinated Adaptive Traffic System: A Traffic Responsive Method of Controlling Urban Traffic*; Roads and Traffic Authority: Sydney, NSW, Australia, 1990.
5. Pelletier, M.P.; Trepanier, M.; Morency, C. Smart card data use in public transit: A literature review. *Transp. Res. Part C Emerg. Technol.* **2011**, *19*, 557–568.
6. Choudhury, T.; Consolvo, S.; Harrison, B.; Hightower, J.; LaMarca, A.; Legrand, L.; Rahimi, A.; Rea, A.; Bordello, G.; Hemingway, B.; et al. The mobile sensing platform: An embedded activity recognition system. *IEEE Pervasive Comput.* **2008**, *7*, 32–41.
7. Sohn, T.; Varshavsky, A.; LaMarca, A.; Chen, M.Y.; Choudhury, T.; Smith, I.; Consolvo, S.; Hightower, J.; Griswold, W.G.; de Lara, E. Mobility detection using everyday GSM traces. In Proceedings of the 8th International Conference on Ubiquitous Computing, Orange County, CA, USA, 17–21 September 2006; Springer: Heidelberg, Germany, 2006; pp. 212–224.
8. Cottrill, C.D.; Pereira, F.C.; Zhao, F.; Dias, I.F.; Lim, H.B.; Ben-Akiva, M.E.; Zegras, P.C. Future mobility survey: Experience in developing a smartphone-based travel survey in Singapore. *Transp. Res. Rec.* **2013**, *2354*, 59–67.
9. Spaccapietra, S.; Parent, C. Adding meaning to your steps. In *Conceptual Modeling—ER 2011, Brussels, Belgium*; Jeusfeld, M., Delcambre, L., Ling, T.W., Eds.; Springer: Heidelberg, Germany, 2011; pp. 13–31.
10. Spaccapietra, S.; Parent, C.; Damiani, M.L.; de Macedo, J.A.; Porto, F.; Vangenot, C. A conceptual view on trajectories. *Data Knowl. Eng.* **2008**, *65*, 126–146.
11. Byon, Y.; Abdulhai, B. Impact of sampling rate of GPS enabled cell phones on mode detection and GIS map matching performance. In Proceedings of the Transportation Research Board Annual Meeting, Washington, DC, USA, 21–25 January 2007; Volume 7.
12. Byon, Y.; Abdulhai, B.; Shalaby, A. Real-time transportation mode detection via tracking Global Positioning System mobile devices. *J. Intell. Transp. Syst. Technol. Plan. Oper.* **2009**, *13*, 161–170.

13. Gonzalez, P.; Weinstein, J.; Barbeau, S.; Labrador, M.; Winters, P.; Georggi, N.; Perez, R. Automating mode detection for travel behaviour analysis by using global positioning systems enabled mobile phones and neural networks. *IET Intell. Transp. Syst.* **2010**, *4*, 37–49.
14. Reddy, S.; Mun, M.; Burke, J.; Estrin, D.; Hansen, M.; Srivastava, M. Using mobile phones to determine transportation modes. *ACM Trans. Sens. Netw.* **2010**, *6*, 1–27.
15. Stenneth, L.; Wolfson, O.; Yu, P.S.; Xu, B. Transportation mode detection using mobile phones and GIS information. In Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, Chicago, IL, USA, 1–4 November 2011.
16. Zheng, Y.; Chen, Y.; Li, Q.; Xie, X.; Ma, W.Y. Understanding transportation modes based on GPS data for web applications. *ACM Trans. Web* **2010**, *4*, 1.
17. Zheng, Y.; Liu, L.; Wang, L.; Xie, X. Learning transportation mode from raw GPS data for geographic applications on the web. In Proceedings of the 17th International Conference on World Wide Web, Beijing, China, 21–25 April 2008; ACM: New York, NY, USA, 2008.
18. Biljecki, F.; Ledoux, H.; Oosterom, P.V. Transportation mode-based segmentation and classification of movement trajectories. *Int. J. Geogr. Inf. Sci.* **2013**, *27*, 385–407.
19. Tsui, S.; Shalaby, A.S. Enhanced system for link and mode identification for personal travel surveys based on global positioning systems. *Transp. Res. Rec.* **2006**, *1972*, 38–45.
20. Siler, W.; Buckley, J.J. *Fuzzy Expert Systems and Fuzzy Reasoning*; Wiley Interscience: Hoboken, NJ, USA 2004.
21. Bao, L.; Intille, S. Activity recognition from user-annotated acceleration data. In *Pervasive Computing; Lecture Notes in Computer Science*; Ferscha, A., Mattern, F., Eds.; Springer: Berlin/Heidelberg, Germany, 2004; Volume 3001, Book Section 1, pp. 1–17.
22. Bulling, A.; Blanke, U.; Schiele, B. A tutorial on human activity recognition using body-worn inertial sensors. *ACM Comput. Surv.* **2014**, *46*, 1–33.
23. Stopher, P. GPS, Location, and household travel. In *Handbook of Transport Geography and Spatial Systems*; Hensher, D., Ed.; Elsevier Ltd.: Amsterdam, The Netherlands, 2004; pp. 432–449.
24. Safi, H.; Mesbah, M.; Ferreira, L. ATLAS Project- developing a mobile-based travel survey. In Proceedings of the Australian Transportation Research Forum, Brisbane, QLD, Australia, 2–4 October 2013.
25. Liao, L.; Fox, D.; Kautz, H. Location-based activity recognition using relational Markov networks. In Proceedings of the 19th International Joint Conference on Artificial Intelligence, Edinburgh, UK, 30 July–5 August 2005; Morgan Kaufmann Publishers Inc.: San Francisco, CA, USA, 2005; pp. 773–778.
26. Dodge, S.; Weibel, R.; Forootan, E. Revealing the physics of movement: Comparing the similarity of movement characteristics of different types of moving objects. *Comput. Environ. Urban Syst.* **2009**, *33*, 419–434.
27. Ohashi, H.; Akiyama, T.; Yamamoto, M.; Sato, A. Modality classification method based on the model of vibration generation while vehicles are running. In Proceedings of the Sixth ACM SIGSPATIAL International Workshop on Computational Transportation Science, Orlando, FL, USA, 5–8 November 2013; ACM: New York, NY, USA, 2013; pp. 37–42.
28. Hemminki, S.; Nurmi, P.; Tarkoma, S. Accelerometer-based transportation mode detection on smartphones. In Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems, Roma, Italy, 11–15 November 2013; ACM: New York, NY, USA, 2013; pp. 1–14.
29. Schuessler, N.; Axhausen, K. Processing GPS raw data without additional information. In proceedings of the 88th Annual Meeting of the Transportation Research Board, Washington, DC, USA, 11–15 January 2009.
30. Xu, C.; Ji, M.; Chen, W.; Zhang, Z. Travel mode from GPS trajectories through fuzzy pattern recognition. In Proceedings of the International Conference on Fuzzy Systems and Knowledge Discovery (FSKD), Yantai, China, 10–12 August 2010.
31. Panella, M.; Rizzi, A.; Mascioli, F.; Massimo, F.M.F.; Martinelli, G. A Neuro-fuzzy system for the prediction of the vehicle traffic flow. In Proceedings of the Fuzzy Logic and Applications: 5th International Workshop, WILF 2003, Naples, Italy, 9–11 October 2003; Revised Selected Papers; Gesu, V., Masulli, F., Petrosino, A., Eds.; Springer: Berlin/Heidelberg, Germany, 2006.
32. Henry, J.J.; Farges, J.L.; Gallego, J.L. Neuro-fuzzy techniques for traffic control. *Control Eng. Pract.* **1998**, *6*, 755–761.

33. Wannige, C.T.; Sonnadara, D.U.J. Adaptive neuro-fuzzy traffic signal control for multiple junctions. In Proceedings of the International Conference on Industrial and Information Systems (ICIIS), Peradeniya, Sri Lanka, 28–31 December 2009.
34. Dell’Orco, M.; Circella, G.; Sassanelli, D. A hybrid approach to combine fuzziness and randomness in travel choice prediction. *Eur. J. Oper. Res.* **2008**, *185*, 648–658.
35. Duckham, M. *Decentralized Spatial Computing*; Springer: Heidelberg, Germany, 2013.
36. Wolf, J. Using GPS Data Loggers to Replace Travel Diaries in the Collection of Travel Data. Ph.D. Thesis, Georgia Institute of Technology, Atlanta, GA, USA, July 2000.
37. Aggarwal, J.; Ryoo, M. Human activity analysis: A review. *ACM Comput. Surv.* **2011**, *43*, 16:1–16:43.
38. Terrovitis, M.; Nikos, M. Privacy preservation in the publication of trajectories. In Proceedings of the 9th International Conference on Mobile Data Management, Beijing, China, 27–30 April 2008; pp. 65–72.
39. Zadeh, L.A. Fuzzy sets. *Inf. Control* **1965**, *8*, 338–353.
40. Negnevitsky, M. *Artificial Intelligence: A Guide to Intelligent Systems*; Addison-Wesley: Boston, MA, USA, 2002.
41. Jang, J.S.R. ANFIS: Adaptive-network-based fuzzy inference systems. *IEEE Trans. Syst. Man Cybern.* **1993**, *23*, 665–685.
42. Bobillo, F.; Straccia, U. Fuzzy ontology representation using OWL 2. *Int. J. Approx. Reason.* **2011**, *52*, 1073–1094.
43. Doherty, S.T.; Papinski, D.; Lee-Gosselin, M. Internet-based prompted recall diary with automated GPS activity-trip detection: System design. In Proceedings of the Transportation Research Board 85th Annual Meeting, Washington, DC, USA, 22–26 January 2006.
44. Stopher, P.; Bullock, P.J.; Horst, F.N.H. Conducting a GPS survey with a time-use diary. In Proceedings of the 83rd Annual Conference of the Transportation Research Board, Washington, DC, USA, 11–15 January 2003.



© 2016 by the authors; licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC-BY) license (<http://creativecommons.org/licenses/by/4.0/>).