Integrating SUMO and Kalman Filter Models Towards a Social Network Based Approach of Public Transport Arrival Time Prediction

Ahmad Faisal Abidin, Mario Kolberg, Amir Hussain

Computing Science and Mathematics
University of Stirling
Stirling, Scotland
{faa, mko, ahu}@cs.stir.ac.uk

Abstract — Bus arrival time is a key service for improving public transport attractiveness by providing users with an accurate arrival time. In this research, a model of bus arrival time prediction, which aims to improve arrival time accuracy, is proposed. The arrival time will be predicted using a Kalman Filter (KF) model, by utilising information acquired from social networks. Social Networks feed road traffic information into the model, based on information provided by people who have witnessed events and then updated their social media accordingly. This research compares different KF models and identifies the best models to use for traffic prediction by employing traffic simulator, Simulation in Urban Mobility (SUMO). This paper discusses modelling a road journey using Kalman Filters and verifying the results with a corresponding SUMO simulation. Integrating the SUMO measures with the KF model can be seen as an initial step to verifying our premise that realtime data from social networks can eventually be used to improve the accuracy of the KF prediction. In order to acquire optimal estimation, verifying the trustworthiness of social network information is crucial. This paper discusses some ideas to establish a level of trust in social networks. This is important as KF model prediction will suffer if bogus information from social networks is used.

Keywords - Kalman Filter, SUMO, Social Networks.

I. INTRODUCTION

Literature reviews [1][2] have shown that most arrival time prediction models are based on historical arrival patterns and/or other explanatory variables correlated with the arrival time. The explanatory variables used in these previous studies include historical arrival time (or travel time), schedule adherence, weather condition, time-of-day, day-of-week, dwell time, number of stops, distance between stops and road–network condition [1][2]. The collection and transmission of such variables has been largely made possible using emerging technologies, including wireless communication, automatic vehicle location (e.g. Global Positioning System(GPS)), and other sensing technologies. The effect of congestion was treated differently in most models. For example, some have used traffic properties like volume and speed from simulation results [3], while others have clustered their data into different time periods [4]. Historical data based models are used in geographical areas with less traffic congestion, because the models assumed cyclical traffic patterns. Kalman Filter (KF) techniques and Artificial Neural Network (ANN) approaches were used mostly in urban areas [1]. KF models can be applied while a bus trip is in progress due to its simplicity of calculation. While other models are dependent on cyclical traffic data patterns or require independence between variables, ANNs do not require that variables be uncorrelated or have a cyclic pattern.

There are many algorithms based on mathematical theory and/or statistical models that have been proposed for bus travel time prediction. However, there is a gap amongst those algorithms. One particular issue is the question of how the algorithm will receive and incorporate live real-time traffic event information. Without receiving such information, those algorithms could not produce an accurate result [5,6,7]. One example of such information is detail on road incidents. This kind of information is vital to be passed to KF Model for further processing. Social network communication is a novel way to collect and include current road condition information, rather than using GPS or other (road) sensors to detect the numbers of cars on the road and speed of travel. In addition, this approach allows for the identification of unexpected traffic events, and the subsequent inclusion of this new, real-time information, as part of potential route calculations and updates. This provides updated information during journeys that may not have been available when travel was initially planned or started. In this situation, social networks can play a pivotal role as a source of information. Simulation in Urban Mobility (SUMO) can run realistic simulations, including traffic patterns and road conditions. In this paper, we model a particular journey with a Kalman model and then try and verify the prediction with the SUMO simulation. Further, we feed location information from SUMO into a KF model to improve its prediction accuracy.

This demonstrates: verification of KF prediction results, an integrated experimentation environment for arrival time predictions, and that the use of live data improves the KF prediction accuracy. This can be exploited by using road condition information from social networks together with KF prediction models.

The remainder of this paper is structured as follows. Section II discusses the KF Models used in this paper, as well as briefly reviewing other work in the literature. Section III then provides a description of SUMO and the modifications
made to successfully utilise it as part of this work. Experimental results are presented in section IV, with section V concluding the paper.

II. KALMAN FILTER (KF) MODELS

Much research has been conducted into forecasting bus travel/arrival time using a variety of techniques and models. This includes techniques such as time series analysis and ANNs, and a summary is shown in table 1. However, only two models, Statistical [8] and Kalman Filter models [9], take dynamic information into consideration. KFs are a powerful tool when it comes to controlling noisy systems [10]. The basic idea of KFs is that they process (filter) incoming data, and produce an improved quality of output data.

<table>
<thead>
<tr>
<th>Technique</th>
<th>Remarks</th>
<th>Dynamic Info</th>
</tr>
</thead>
<tbody>
<tr>
<td>Historical [11]</td>
<td>Predict travel time at particular time as the average travel time for the same period historically</td>
<td>No</td>
</tr>
<tr>
<td>Time series analysis [12]</td>
<td>Predict assuming historical patterns will remain the same in the future</td>
<td>No</td>
</tr>
<tr>
<td>Artificial Neural Network [13]</td>
<td>Predict based on example data using large database for accurate prediction</td>
<td>No</td>
</tr>
<tr>
<td>Real-time approach [14]</td>
<td>Assume the future travel time to be the same as the present one</td>
<td>No</td>
</tr>
<tr>
<td>Statistical [8]</td>
<td>Predict travel time based on dependent variable based on a function with independent variables</td>
<td>Yes</td>
</tr>
<tr>
<td>Kalman Filters/KF [8,10]</td>
<td>Establish relationships between variables, and corroborates using field observation</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Kalman Filters are an estimation approach. That is they infer values of parameters from observations which may be noisy, inaccurate, and uncertain. Importantly, unlike many other approaches, Kalman Filters are recursive and hence can take new observations into account as they arrive. With this, Kalman Filters can be executed at runtime of the system under observation. The algorithm is referred to as a ‘filter’ as calculating the estimate from noisy input data essentially ‘filters’ out the noise [15]. Kalman filters estimate a process by estimating the process state at a given time and then obtaining feedback in the form of noisy measurements. Generally, the equations for the Kalman filter fall into two groups: time update equations and measurement update equations. Time update equations are responsible for predicting forward (in time), using the current state and error covariance estimates to obtain a priori estimates for the next time step. The measurement update equations are responsible for the feedback, i.e. for incorporating a new measurement into the a priori estimate to obtain an improved a posteriori estimate. The time update equations can also be thought of as predictor equations, while the measurement update equations can be thought of as corrector equations. Indeed, the final estimation algorithm resembles a predictor/corrector algorithm for numeric problems [15].

The addition of state constraints (physical information) to a KF can significantly improve the estimation accuracy of the filter [5,7]. In this sense, the addition of linear information may be useful for that goal. KF models theoretically deliver the best and most up to date results when they have continuous access to dynamic information. Many existing models therefore make use of dynamic information. However, the performance of these models can often suffer due to issues with taking account of scenarios that utilise rapidly updating real world information. Many road users utilise GPS to navigate and estimate the duration of their journeys. However, it is not always possible to handle some events solely by GPS information. For instance, if there is an accident and we know it will take 30 minutes or more to clear; this information has to be made available to and be included in the prediction algorithm. Social networks can distribute real time information on road traffic, which could potentially improve the accuracy of arrival time prediction.

III. SIMULATION OF URBAN MOBILITY (SUMO)

SUMO [16,17] is a road traffic simulation engine which is designed to handle large road networks. Every vehicle is modelled explicitly. Every vehicle has their own routes, and moves individually through the network. This means that for every vehicle in the simulated network its current location and speed are known. At every time step which has a duration of 1s, these values are updated. The simulation of the network is time discrete and space continuous. When simulating traffic, the street attributes, such as maximum velocity and right of way rules, are taken into account. Conceived as multi-modal environment, SUMO not only can simulate a model of car movements, but also a variety of public transportation models including trains and buses. Public transportation is described by a departure time and the route it takes, which again is made up of subroutes that describe a single traffic modality.

A. Road Traffic Simulation Theory in SUMO

Vehicle movement in SUMO is mostly simulated by employing queue approaches and single vehicles are moved between such queues. In SUMO every single vehicle is simulated by taking into account the driver’s behaviour and vehicle’s capability. SUMO offers space discrete and space continuous simulation. Each vehicle has a certain location described by a floating point number. Every street divides into cells and vehicles are moving on a street by moving from cell to cell.

The simulation advances in steps with the length of the default timestep being one second. The position of each vehicle is determined by the lane it occupies. In order to construct a vehicle movement on the road network, a Car Following Model is implemented in SUMO. The Car Following Model describes how one vehicle follows another and how a driver reacts to the changes in the relative positions of the vehicle ahead in an uninterrupted flow. Two main characteristics were taken into account, vehicle loca-
tion and speed. The concept of Car Following Model is configured as follows: The lead vehicle is denoted by $n$ and the vehicle behind is denoted by $n+1$. Position and speed of the lead vehicle, at time instant $t$ are denoted by $x_n^t$ and $v_n^t$ respectively. The position and speed of the follower are also denoted similarly to lead vehicle, $x_{n+1}^t$ and $v_{n+1}^t$. Thus, the gap between the lead vehicle and the follower vehicle is $x_n^t - x_{n+1}^t$.

Figure 1. Space continuous (C) and space discrete (A & B) in SUMO

![Space continuos (C) and space discrete (A & B) in SUMO](image)

Figure 2. Car Following Model in SUMO

![Car Following Model in SUMO](image)

Vehicles in SUMO have their own internal route which is not shared with other vehicles. It is also possible to define two vehicles using the same route. In this case the route must be “externalized”, which means it needs to be defined prior to being referenced by the vehicles. Figure 3 and Table II, show the screenshot and brief explanations of the way to configure a vehicle in order to move on particular routes.

![Definition of vehicles and routes in SUMO](image)

**Table II. Descriptions of Setting Vehicle and Route**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type/Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time</td>
<td>(Simulation)</td>
<td>The time step described by the values within this timestep-element</td>
</tr>
<tr>
<td>Id</td>
<td>Id</td>
<td>The name of the vehicle</td>
</tr>
<tr>
<td>Type</td>
<td>String</td>
<td>The id of the vehicle type to use for this vehicle</td>
</tr>
<tr>
<td>Depart</td>
<td>Float</td>
<td>The time step at which the vehicle shall enter the network</td>
</tr>
<tr>
<td>Route edges</td>
<td>Id</td>
<td>The id of the route the vehicle shall drive along</td>
</tr>
<tr>
<td>Maxspeed</td>
<td>Float</td>
<td>The vehicle's maximum velocity (in m/s)</td>
</tr>
<tr>
<td>CarFollowing</td>
<td>carFollowing-Krauss</td>
<td>The Krauß-model with some modifications which is the default model utilised in SUMO</td>
</tr>
<tr>
<td>minGap</td>
<td>Float</td>
<td>Empty space after leader [m]</td>
</tr>
</tbody>
</table>

**B. Traffic Modelling in SUMO**

SUMO offers the possibility to model traffic demand [18]. A trip is a vehicle movement from one place to another defined by the starting edge (street), the destination edge, and the departure time. A route is an expanded trip, it not only contains the first and the last edge, but all edges the vehicle will pass. SUMO needs routes as input for vehicle movements. There are several ways to generate routes for SUMO. The choice depends on the available input data.

In this work, the vehicular demand modelling was derived from the project Travel and Activity Patterns Simulation (TAPAS) in Cologne City, Germany [19]. The TAPAS project generated trips of drivers by exploiting information on the population, demographic characteristics, the points of interests in the urban area, and places where work and free-time activities take place. It considered information on numbers of people living in an area, household distribution and socio-demographic characteristics. This information was then used to construct a daily activity plan which also takes into account age, gender, and employment status and vehicle availability. Within the context of the TAPAS project, data collected in the City of Cologne was recorded, including 30,700 daily activity reports from more than 7000 households. TAPAS published this open data and permits users to integrate it with other vehicular models.

The research presented in this paper leverages the openness of TAPAS by constructing a model of bus movement based on the TAPAS data.
Road accidents were constructed using TAPAS data on how the traffic flow reacts over particular traffic incidents. This resulted in a realistic traffic model for our experimentation. Figure 4 and 5 show two screenshots of SUMO. Figure 4 presents a large scale overview of the road network, whereas Figure 5 depicts a small part of the network with some vehicles present.

SUMO Output

SUMO allows for the generation and output of a large number of different measures to files or socket connections. Outputs such as raw vehicle positions, trip information, and vehicle route information are triggered using the command line. For this research, we employ all of the outputs above. Raw vehicle positions contain edges (streets) and lanes along with the vehicles driving on them for every time step. Vehicles are described by their name, position and speed. This output has been utilised as input to our Kalman Filter model to update the predictions based on real time trip information, as will be discussed in the following section. The trip information contains the vehicle departure time, the journey start time, and the arrival time; stored as XML data (Fig. 6 and Fig. 7).

**C. SUMO Output**

SUMO allows for the generation and output of a large number of different measures to files or socket connections. Outputs such as raw vehicle positions, trip information, and vehicle route information are triggered using the command line. For this research, we employ all of the outputs above. Raw vehicle positions contain edges (streets) and lanes along with the vehicles driving on them for every time step. Vehicles are described by their name, position and speed. This output has been utilised as input to our Kalman Filter model to update the predictions based on real time trip information, as will be discussed in the following section. The trip information contains the vehicle departure time, the journey start time, and the arrival time; stored as XML data (Fig. 6 and Fig. 7).

**Values in Figure 6 are defined in Table II:**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time</td>
<td>(Simulation)seconds</td>
<td>The time step described by the values within this timestep-element</td>
</tr>
<tr>
<td>Id</td>
<td>Id</td>
<td>The id of the edge/lane/vehicle</td>
</tr>
<tr>
<td>Pos</td>
<td>Metres</td>
<td>The position of the vehicle at the lane within the described time step</td>
</tr>
<tr>
<td>Speed</td>
<td>Metres/seconds</td>
<td>The speed of the vehicle within the described time step</td>
</tr>
</tbody>
</table>

From Figure 6, it can be seen that at time step 22630 the bus is at position 506.26 metres of edge #2345698 and lane #2345698_0. As can be seen from the static road network data in Figure 7, the total length of lane 2345698_0 is 671.38 metres. Thus the total distance travelled by the bus up to this point can be calculated by the lengths of the edges and lanes travelled so far plus the current position in the current edge/lane. This data is then used together with the Kalman model.

**D. Linking SUMO and Kalman Filter Model**

In order to link SUMO and the KF model, we integrate the components discussed in the previous sections in Matlab. The outcome from the visualisation can then be plotted to produce the end result. A bus journey which has been modelled in SUMO is utilised to feed data into and also verify the results from the KF Model.
Journey information from SUMO will be used with the Kalman Filter. Our KF Model has been designed to correspond to the journey programmed in SUMO.

The Kalman model is designed to produce an estimate of position based on the vehicle speed. Bus position and speed are the data items from the SUMO simulation. We configure the Kalman Model as follows:

\[ x = \begin{bmatrix} \text{Position} \\ \text{Speed} \end{bmatrix} \]  \hspace{0.5cm} (1)

We then set the system model as follows:

\[ x_{k+1} = \alpha x_k + w_k \]  \hspace{0.5cm} (2)

\[ y_k = \mu x_k + z_k \]  \hspace{0.5cm} (3)

\[ = \begin{bmatrix} 1 \\ \Delta t \end{bmatrix}, \mu = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \]  \hspace{0.5cm} (4)

\( x_k \) represents the state variable, \( y_k \) the measurement variable, \( \alpha \) the state transition matrix, \( \mu \) the state measurement matrix, and \( z_k \) is the measurement noise.

\( \alpha \) describes how the system changes over time. \( \Delta t \) is the interval of the time measuring position. Equation 2 predicts the state at the next timestep for system model \( \alpha \). Subscript \( k \) clarifies KF model is executed in a recursive manner. The state measurement matrix \( \mu \) is used in Equation 3 predicting the position based on measured velocity.

In order to check system model representing the system above, we expand the expression for the equation that describing the state variable by applying the matrix \( \alpha \) in the system model.

\[ x_{k+1} = \alpha x_k + w_k \]  \hspace{0.5cm} (4)

\[ = \begin{bmatrix} 1 \\ \Delta t \end{bmatrix} x_k + w_k \]  \hspace{0.5cm} (5)

If the definitions of the state variables are applied, the meaning becomes clear.

\[ \begin{bmatrix} \text{Position} \\ \text{Speed} \end{bmatrix}_{k+1} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} x_k \begin{bmatrix} \text{Position} \\ \text{Speed} \end{bmatrix} + \begin{bmatrix} 0 \\ \Delta t \end{bmatrix} w_k \]  \hspace{0.5cm} (6)

\[ = \begin{bmatrix} \text{Position} + \text{Speed} \cdot \Delta t \\ \text{Speed} + w \end{bmatrix} \]  \hspace{0.5cm} (7)

Below are the explanations relating to the position:

\[ Position_{k+1} = Position_k + Speed_k \cdot \Delta t \]  \hspace{0.5cm} (8)

The expression above is a mathematical description of a physical principle stating that ‘current position = previous position + displacement’. This is why the system noise is not included in the expression. Below are the explanations about the speed:

\[ Speed_{k+1} = Speed_k + w_k \]  \hspace{0.5cm} (9)

This means that the speed of the vehicle is affected by the system noise \( (w_k) \). Let, looks into the measurement equation of the system model. Apply matrix \( \mu \) to the measurement equation.

\[ y_k = \mu x_k + z_k \]  \hspace{0.5cm} (10)

\[ = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} \text{Position} \\ \text{Speed} \end{bmatrix} + z_k \]  \hspace{0.5cm} (11)

\[ = position_k + z_k \]  \hspace{0.5cm} (12)

The expression above explains that we are measuring the speed (to measure position just change the value in the matrix \((1 0)\) and in this measurement noise is included. These facts match exactly with the description given in the example. The explanation of how the system mode is related to the real system has been given above. As explained, the matrices \( \alpha \) and \( \mu \) are not chosen arbitrarily, however set as a result of the modelling of physical relationship in the system. That is, the system model is a mathematical description of the physical relationship between speed and position, and the factors affecting speed.

IV. KALMAN FILTER RESULTS AND DISCUSSION

In this section we present the results from a comparison of estimation accuracy of Linear KF, Extended KF (EKF), and Unscented KF (UKF). Linear and non-linear KF models each have their own advantages [20]. No general statement can be made as to which is best, because it depends on the particular situation at hand. Comparison of different types of KF models, including conventional linear, and various types of non-linear KFs which are essentially variants of extended KFs linearizing nonlinear assumptions, is vital to identify the optimal model for estimation of arrival time.

This section presents our initial experimentation with our integrated KF-SUMO approaches to estimate the distance travelled of a bus. We implemented the bus journey in SUMO, which then provided both the final time estimates for comparison, and also the step by step journey data that is used with our KF approaches.

There are 2 sets of figures presented in this section, with each corresponding to a programmed journey. Fig. 10 to 13 depicts data on a journey which completes as planned. No unforeseen events such as an accident occur. Fig. 14-17 correspond to the same journey, but with an incident occurring in the 10th minute. The length of the journey is 3075 metres. Under normal conditions, the journey takes around 13 minutes, with the bus travelling on average 22 metres/minute. Process noise during the journey (delays caused by traffic lights, roadwork etc.) is set at 10 metres/minute.
A. Incident Free Journey

Fig. 10 - 13 show results from the incident free journey. Solid lines represent the estimation of the distance travelled using the left Y axis. The dashed line indicate the vehicle speed using the right Y axis.

In Fig 10, we can observe considerable differences between the predictions by all the KF algorithms and the SUMO measures. At the end of the SUMO measures, the closest KF estimate the distance travelled at just over 2500m. Clearly, this needs improvement.

Fig. 11 and 12 show results when SUMO measures are used to update the KF predictions at regular intervals during the journey. Fig. 11 shows results with the KF models loading SUMO journey data every 6 minutes. Although there is still considerable difference between the KF models results and the Sumo measurements (e.g. min 4 to min 9), by the end of the journey the KF and SUMO results are much closer than in Fig. 11. Results shown in Fig 12 with data exchanges every 3 min, indicate a further improvement of the estimation results of all KF models.

Finally, results in Fig. 13 show that data input to the KF models every minute, the KF results are very similar to the actual SUMO measurement. The Kalman predicted velocity is quite close to the curve of SUMO velocity. Thus, additional data input into KF improve predictions considerably. More frequent information feeds will produce better results.

B. Journey with Traffic Incident

Similarly to the previous section, Fig. 14 - 17 show the prediction results from the KF models together with the measures from SUMO. However, now a traffic incident occurs. The vehicle comes to a complete stop for a few minutes causing a delay in the arrival time. The distance travelled overall is identical to the experiments in the previous section. The incident represents a challenge to arrival time prediction using KF models. The drop in speed and lack of movement can be seen in Fig. 14-17, starting at min 11, with the incident cleared in min 17.

In Fig. 14, where the KF models alone is being used with no updated journey information, there is a large discrepancy between the Kalman predictions and SUMO results. At the end of the journey, the Kalman Filter models estimates that the bus travels 2049 metres, which is significantly lower than the actual SUMO results.

Fig. 15 then shows the KFs results compared to SUMO measures with data fed into the KF models every 6 minutes. These results show that although the discrepancy between KFs and SUMO is smaller, the velocity estimation is still rather noisy. These results are improved in Figure 16, where the data exchange interval is increased to 3 minutes. Now the velocity prediction in the Kalman models increases during minutes 1 to 6, and then reduces from just under 30 metres/minutes to close to zero due to the incident. The distance prediction is much more accurate due to receiving information from SUMO during the journey.

Finally, in Fig. 17 the KF models receive information every minute. The KF velocity prediction is very similar in terms of behaviour to the SUMO results. The KF predictions of distance travelled closely follows the SUMO measurements throughout the journey. Clearly the more frequent the information can be made available to the KF model, the
more accurate are its predictions. This is especially important when unforeseen events occur during a journey.

Figure 14. Results without data exchange between KF and SUMO

Figure 15. Data exchanges between KF and Sumo every 6 min

Figure 16. Data exchanges between KF and Sumo every 3 min

From all the results presented in this paper, no particular Kalman Filter algorithm stands out in terms of its accuracy. Perhaps the linear and EKF estimations are closest for the cases where no SUMO measures assist the Kalman prediction. The EKF model is quite similar to the linear KF model. However, EKF models use nonlinear system model equations $f(x_k)$ and $h(x_k)$ in place for $a_k$ and $b_k$ in linear model [20]. Secondly, the matrices $a$ and $b$ are Jacobians of the nonlinear model [20]. While the UKF deviates slightly more than the linear and EKF models, all three Kalman models can be utilized to estimate bus arrival time.

This research focus on improving the accuracy of arrival time prediction approaches for vehicles. The section below will discuss the inclusion of road traffic information from social networks such as Twitter and combining this with a KF prediction algorithm.

V. UTILISING INFORMATION ON SOCIAL NETWORKS

Social networks such as Facebook or Twitter allow users to exchange messages. Especially short messages posted on Twitter reflect events as they happen. Hence, such content is particularly useful for real-time identification of events. Twitter provides hashtags, which allows users to relate their messages to particular events or topics. Specifically, road conditions and incidents are frequently discussed.

Filtering useful information on Twitter in real time is a challenging problem, due to the heterogeneity and immense scale of the data [21]. Twitter users post messages with a variety of content types, including personal updates and fragments of related information [22]. Furthermore, Twitter messages, by design, contain little textual information, and it can be difficult to validate them. However, a number of tools exist, which extract real time information from Twitter messages [23]. Some third party tools such as Tweetbinder and Topsy offer semantics analysis of the retrieved Twitter data. Data can be filtered according to originating from a single user, a group of users, or all public users. Further filters can be set so to capture only tweets originating from within a geographic area and also according to tweets containing particular key words. An example of this real time information is shown in Figure 18.
While travel time data can be obtained through various sources, such as loop detectors, microwave detectors, radar, etc., it is unrealistic to expect the whole roadway network being completely covered by such data collection devices. However, information from Twitter messages such as “M8 Delays of 15min Near J25/J24 westbound” can be used. The information about a 15 minutes delay can be input linearly into Kalman Filter models. However, a number of criteria to establish if the content of the message can be trusted need to be considered before making use of that information. For instance:

- Time of the message (messages sent very soon after the particular event is more valuable, very old messages are out-of-date and counterproductive).
- Location and extend of event (involving carriage way information).
- Location of sender of the message (a sender close by the incident is more reliable)
- The Sender of the message (messages from a trusted sender are more valuable, messages from authorities typically can be utilised).

Thus, besides extracting the actual piece of information from social network messages, a major concern in this context is trust.

VI. PRAGMATIC TRUST IN GENERIC SOCIAL NETWORK

![Diagram of Pragmatic Trust in Generic Social Network]

Figure 19. Trust based reputation built up in Social Networks

Trusting people who send a tweet message relating to road traffic is really important. Clearly, Kalman Filter estimation will deteriorate if false information is used. Thus, in order to tackle this problem, the reputation of the sender needs to be considered. We propose to use the user reputation built up in online social networks to verify the credibility of senders and their messages.

Concepts such as six degrees of separation, social trust, popularity and engagement are utilised and incorporated to establish the trustworthiness of a sender. The concept of six degrees of separation states that any two people in this world are linked with each other by no more than six intermediate friends. Thus, trust can be determined by the degree of relationship between two people, a closer relationship implies a more trustworthy relationship [21]. Personal information and experience from previous interactions could be seen as another trust metric [22]. Furthermore, trust can be found in roles of users. For example, information from a government agency or the police may be seen as highly trustworthy. In terms of the degree of relationship, the number of mutual friends, their behaviour, and relationship history can also be used to define a level of trust.

Building trust in social networks, based on popularity and engagement [22], is another way to identify trusted information. Popularity refers to the acceptance and approval of a member by others in the community, while engagement captures the involvement of someone in the community. Popularity trust can be seen to reflect the trustworthiness of a member in the community, while engagement trust reflects how much a member trusts other members in the community. Nodes with the same level of interactions, but with highly popular friends should have higher popularity trust than those with less popular friends.

This analysis to establish a level of trust can be improved by joining information from other social networks. For instance a Twitter user may have an extensive Facebook or Linkedin network, which allows to build up a more comprehensive picture. Figure 19 highlights trust measures offered by a number of example social network and ecommerce systems.

Reputation in Amazon is based on a ranking system of sellers based on customer feedback and the number of products sold by the seller. Amazon’s seller rating is based on an equation, seller rating = total points/total orders. eBay establishes trust by utilising a reputation system that collects information on the past behaviour of sellers and buyers. Negative feedback can only be left by buyers. Buyers can leave detailed feedback for sellers in 4 specific categories [23]. LinkedIn offers a higher degree of linking of people in certain settings. For instance, LinkedIn profiles usually have a larger number of work colleagues as part of a person’s profile than other social media networks. As users often know people in their network personally, this can be used to indicate a higher level of trust. Facebook provides a social network platform for users to link with each other and to exchange messages and pictures. Facebook also offers tools to check the degree of separation between two users. On average, the distance between any two members is 4 degrees [24]. A lower degree of separation indicates a closer acquaintance and thus a higher level of trust. Twitter provides a platform to be connected to the other twitter users. Twitter facilitates social networking communication and engagements by creating a platform like following-follower and retweeting. These platforms could be leveraged to gain trust.
VII. CONCLUSION

In this work, we integrated journey information obtained from a SUMO simulation into a Kalman Filter models in order to improve the accuracy of bus journey arrival time predictions. We have successfully demonstrated the integration of the two techniques. This provided us with an experimental platform which allowed us to check the Kalman predictions against SUMO measures. Furthermore, we have shown that inserting additional journey data obtained from SUMO in the Kalman prediction can greatly increase its accuracy. Regular information updates can further improve the accuracy.

For the predictions we have employed different types of Kalman Filter models: Linear, Extended, and Unscented Kalman Filters. No single approach offers significantly better predictions than others with the Linear and EKF models having slight advantages of the UKF. As linear models are the most straightforward approaches, for scenarios as described in this paper, these models offer good performance.

The strength of KF models is their ability to predict or estimate the state of a dynamic system from a series of noisy measurements as well as their ability to be executed during a journey. Based on their strength, additional credible and trusted information from social networks can be used to increase the prediction accuracy of KF models. Crucially, as malicious messages can easily be inserted in social networks, only trusted messages from a trusted source must be used. Integrating the information from social networks with the Kalman model, defining a robust approach to trust and ensuring the users’ privacy are key aspects of this work.

Future work is to focus on Big Data Semantics in order to retrieve relevant information relating to road incidents. This research involves filtering the large quantity of initial data in order to return a lower quantity of relevant information that can be used for further processing.

REFERENCES


