Big-Data-Generated Traffic Flow Prediction Using Deep Learning and Dempster-Shafer Theory

Ridha Soua, Arief Koesdwiady, and Fakhri Karray
Centre for Pattern Analysis and Machine Intelligence (CPAMI)
Department of Electrical and Computer Engineering,
University of Waterloo
Waterloo ON, N2L 3G1, Canada
Email: rsoua, abkoesdw, karray@uwaterloo.ca

***Abstract*—This work addresses short-term traffic flow prediction by proposing a big-data-based framework. The proposed framework uses data fusion to deal with heterogeneous data generated from various sources. The data are categorized into two types: streams of data and event-based data. In this work, Deep Belief Networks (DBNs) are used to independently predict traffic flow using streams of data, i.e., historical traffic flow and weather data, and event-based data, i.e., tweets. Furthermore, Dempster's conditional rule for updating belief is used to fuse evidence coming from streams of data and event-based data modules to achieve enhanced prediction. The experimental results using real-world data show the merit of the proposed framework compared to the state-of-the-art ones.**

I. Introduction

One of the fundamental services of planned smart cities is smart mobility. With more than a billion cars on roads today that is expected to double to around 2.5 billion by 2050 [1], offering super-efficient navigation and safer travel journey for road and highway users are among the most difficult and universal challenges faced by traffic management authorities. These authorities are urged to transform the mobility service from a reactive service to a proactive one, which anticipates future gridlocks and proposes ideal departure time for drivers. To achieve these goals, obtaining accurate and reliable information about current and future traffic flow is crucial.

Traffic flow prediction is a challenging problem. During trips, drivers faces several inclement weather conditions, e.g., rain, storm and snow, and unforeseen events, e.g., accidents, concerts, and road closures. These factors impact significantly the travel time of each driver, which consequently affect the overall traffic flow of transportation networks. The Federal Highway Administration (FHWA) indicates that in United States alone, approximately 55% of all delays are caused by non-recurring congestion events, e.g., traffic incidents, work zones, bad weather and special events [2]. According to [3], these events can be classified into two categories: planned events, e.g., roadway construction and maintenance, and special event.

Over the past few years, sensing technologies have spurred the rise of big data enabling anyone to monitor and collect data from sensors, streets, lights and cars. Cisco estimates that 50 billion devices and objects will be connected to the Internet by 2020 [4]. Hence, connected cars evolve in a data-rich environment, where they consistently generate and receive a variety of data. Citizens also produce huge streams of data using social media. While driving, citizens announce their travel plans and share information regarding current traffic or road conditions using Twitter, Facebook, or Waze. This information can be used to enrich our experiences of how cities function and as a fuel that drives predictive related traffic applications such as traffic flow prediction and informed decision-making.

While the prediction of traffic using streams of data has received a lot of attention, the technical capacity of social media, as event-based data, to provide insights on traffic flow is only just now catching up. This paper proposes a Deep Belief Networks (DBNs) based approach to predict traffic flow using streams of data, i.e., historical traffic flow and weather data, and event-based data, i.e., tweets. Subsequently, an extension of Dempster-Shafer Evidence Theory (DSET), namely Dempster's conditional rules for updating belief, is used to fuse traffic prediction beliefs coming from streams of data and event-based data modules. The rest of the paper is organized as follows. Section II describes the current literature on traffic flow prediction. Then, the technical background of the proposed traffic flow prediction framework based on deep learning and conditional approach for evidence combination data is detailed in Section III. In section IV, the proposed big-data-based framework is described. Section V presents the experimental results. Finally, the paper is concluded in Section VI.

II. State of the Art

While using historical traffic data for predicting short-term traffic flow has spurred a lot of studies, only a few research have actually tackled the advantage of using external factors such as weather, planned and unplanned events posted on social media. In this section, we focus on studies that incorporated these influencing factors [5].

In [6], the authors proposed a neurowavelet based framework to forecast hourly traffic taking into account the effect of rain fall. Stationary Wavelet Transform (SWT) has been used to mitigate the problem of non stationary in traffic time series data sets. Predicted components by Auto-Correlation Neural

978-1-5090-0620-5/16/$31.00

Network models are then recombined using an inverse SWT. However, their data set is limited to only two traffic junctions in Dublin city. Moreover, only weekdays of the two first weeks of January were considered for traffic prediction.

The impact of snowy and icy conditions on traffic with data from Buffalo, New York was investigated in [7]. The authors modeled the impact of adverse weather on traffic speed at both macroscopic and microscopic levels. Nevertheless, the authors used a simple linear regression method, which is an inflexible parametric model. Since transportation networks are complex and very correlated, it is crucial to predict traffic flow from a network perspective.

A pertinent information related to traffic events shared in social networks can be used to enhance traffic prediction accuracy. This was the purpose of the study made by Jingrui et al. [8]. They used Twitter data as an external data source for improving long-term traffic prediction. However, their auto-regression model incorporating traffic and social activity intensity is not well adapted to the complexity of high dimension of features generated by transportation systems and social networks.

Authors of [9], improved their study proposed in [7] by including weather variables extracted from Tweets. They evaluated a linear regression model with Twitter-based weather variables. This study shares with other mentioned works the limitations of linear regression models.

Recently, a specific attention was given to deep learning [10], [11] to forecast traffic flow. This was motivated by the fact that deep learning could learn features with a less prior knowledge. In addition, It is able to handle large amount of data and high dimension of features that can be generated by transportation systems in a smart city. However, these recent studies did not take into account neither weather factors nor disrupting events included in social networks.

III. Technical Background

In this section DBNs and DSET are introduced. By implementing DBNs, an accurate short-traffic prediction can be achieved. The prediction uses two types of data: streams of data, i.e., traffic flow and weather data, and event-based data, i.e., tweets. DSET will be used to fuse decisions coming from the two data modules to achieve better prediction.

A. Deep Beliefs Networks (DBNs)

DBNs are a stack of Restricted Bolztmann Machines (RBMs), which are greedily trained using unsupervised algorithms [12]. An RBM is an energy-based undirected graphical model that has no connections within visible ($\mathbf{x}$) or hidden ($\mathbf{h}$) units, as shown in Figure 1.

The energy function of an RBM is defined as:

$$E(\mathbf{x},\mathbf{h}) = -\mathbf{b}^\top\mathbf{x} - \mathbf{c}^\top\mathbf{h} - \mathbf{h}^\top\mathbf{W}\mathbf{x} \qquad (1)$$

where $\mathbf{b}$ and $\mathbf{c}$ are bias vectors associated with the hidden and visible layers respectively, and $\mathbf{W}$ is the weight matrix connecting the visible and hidden layers. The set of parameters $\mathbf{b}, \mathbf{c},$ and $\mathbf{W}$ are denoted by θ. The distribution of observing

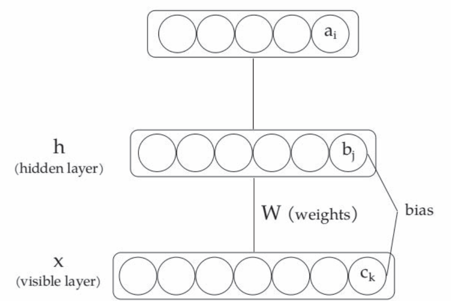

Fig. 1: Restricted Boltzmann Machines

a particular visible and hidden units configuration is defined as:

$$P(\mathbf{x},\mathbf{h}) = \frac{e^{-E(\mathbf{x},\mathbf{h})}}{Z} \qquad (2)$$

where Z is the marginal function. From here, the distribution of observing a set of visible units can be obtained using the following equation:

$$P(\mathbf{x}) = \sum_{\mathbf{H}} P(\mathbf{x},\mathbf{h}) \qquad (3)$$

To train an RBM, the negative log-likehood (NLL) function of all training points ($t = 1, ..., T$), which is given by equation 4, needs to be minimized.

$$NLL = \frac{1}{T}\sum_t l(f(\mathbf{x}^{(t)})) = \frac{1}{T}\sum_t -\log P(\mathbf{x}^{(t)}) \qquad (4)$$

Furthermore, the NNL function is optimized with respect to θ using stochastic gradient descent algorithm. Using this algorithm, the set of parameter θ of the RBMs is updated according to the following equation:

$$\theta^{(t+1)} = \theta^{(t)} - \alpha\left(\nabla_\theta(-\log P(\mathbf{x}^{(t)}))\right) \qquad (5)$$

where α is the learning rate, and for each parameter $\mathbf{W}$, $\mathbf{b}$, and $\mathbf{c}$, the update equations are given as the following:

$$\begin{aligned}
\mathbf{W}^{(t+1)} &= \mathbf{W}^{(t)} - \alpha\left(\mathbf{h}(\mathbf{x}^{(t)})\mathbf{x}^{(t)^\top} - \mathbf{h}(\tilde{\mathbf{x}})\tilde{\mathbf{x}}^\top)\right) \\
\mathbf{b}^{(t+1)} &= \mathbf{b}^{(t)} - \alpha\left(\mathbf{h}(\mathbf{x}^{(t)}) - \mathbf{h}(\tilde{\mathbf{x}})\right) \\
\mathbf{c}^{(t+1)} &= \mathbf{c}^{(t)} - \alpha\left(\mathbf{x}^{(t)} - \tilde{\mathbf{x}}\right) \qquad (6)
\end{aligned}$$

where $\mathbf{h}(\mathbf{x}) \triangleq \mathrm{sigmoid}(\mathbf{b}+\mathbf{W}\mathbf{x})$, and $\tilde{x}$ is the negative sample estimated using Gibbs sampling in Contrastive Divergence algorithm.

The parameters learned using the unsupervised algorithm then are used to initialize neural networks. In this work, the neural networks are trained using supervised back-propagation algorithm to learn several tasks at the same time, where each task consists of traffic flow prediction at each road. This type of learning is known as Multi Task Learning (MTL) [11]. This learning method allows each task to share feature representation constructed by the stacked RBMs. In addition, since the main task in of the DBNs is regression, the output layer of the DBNs uses linear activation functions. Intuitively, MTL is suitable for transportation networks, where several freeways are connected to each other. Figure 2 depicts a multi task regression stacked on top of the stacked RBMs.

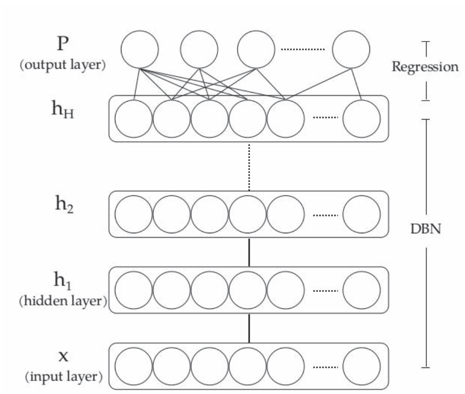

Fig. 2: DBNs architecture for multi-task regression

B. Dempster-Shafer Evidence Theory (DSET)

The traffic flow provided by DBNs using various sources of data will be fused to provide better decisions regarding traffic flow prediction accuracy. One of the methods for data fusion is Dempster-Shafer Evidence Theory (DSET). This method was first introduced in [13]. The crucial feature of DSET is the combination of evidence obtained from several sources by modeling the conflict between evidences. Unlike Bayesian networks, DSET uses masses instead of probabilities.

In DSET, the total set of mutually exclusive and exhaustive propositions of interest that a decision authority may discern, is called the Frame of Discernment (FoD). It is the set of possible conclusions to be drawn. Let $\theta = \{\theta_1 \dots . \theta_n\}$ be an FoD. A singleton proposition θ_i denotes the lowest level of discernible information. The power set of θ is the set of all possible subsets of θ. Given $|\theta| = n$, the power set denoted by 2^θ contains 2^n elements that are composed of all the subsets of θ. In the following, $\bar{A}$ denotes all singletons in θ that are not included in A. The Basic Belief Assignment (BBA) or mass assignment provides the support for proposition A.

Definition 1: The mapping m: $2^\theta \mapsto [0,1]$ is a BBA for the frame θ if

$$m(\emptyset) = 0 \text{ and } \sum_{A \subseteq \theta} = 1 \tag{7}$$

The set of propositions in a frame θ that possess non-zero BBAs or masses are called the focal elements of θ, and is denoted by $\mathcal{F} = \{A \subseteq \theta : m(A) > 0\}$. The triple variables $\{\theta, \mathcal{F}, m\}$ is called the Body of Evidence (BoE). For any event A, all testimonies in its favour can be collected and the *belief function* which corresponds to its can be determined. By taking into account all the focal elements related to A, its *plausibility function* can be determined. Consider the following definition.

Definition 2: Given a BoE $\{\theta, \mathcal{F}, m\}$ and $A \subseteq \theta$, Belief and Plausibility functions can be defined as:

$$\begin{aligned} \text{Bel}(A) &= \sum_{X \subseteq A} m(X) \\ \text{Pl}(A) &= \sum_{X \cap A \neq \emptyset} m(X) \end{aligned} \tag{8}$$

While $m(A)$ measures the support assigned to proposition A only, the belief assigned to A measures the supports for all proper subsets of A. The plausibility function can also be derived from the belief function as follows

$$\text{Pl}(A) = 1 - \text{Bel}(\bar{A}) \tag{9}$$

1) Dempster's Rule of Combination: In DSET, the Dempster's Rule of Combination (DRC) is used to find a new BBA that aggregates the evidence generated by several BoEs spanning the same FoD. It can be seen as several experts indicating their opinion via the attribution of masses over θ. DRC highlights combined masses of belief in the following way:

Definition 3: Given $BoE_i = \{\theta_i, \mathcal{F}_i, m_i\}$, where $i = 1, 2$ and $\theta = \theta_1 = \theta_2$, the DRC generates the following new BBA defined as:

$$BBA_{12} \equiv BoE_1 \bigoplus BoE_2 = \{\theta_{12}, \mathcal{F}_{12}, m_{12}(\cdot)\} \tag{10}$$

$$m_{12}(A) = \frac{\sum_{B,C:B\cap C=A} m_1(B) \cdot m_2(C)}{(1 - \sum_{B,C:B\cap C=\emptyset} m_1(B)m_2(C))}, \ \forall B, C \subseteq \theta \tag{11}$$

It is worth noting that $\sum_{B,C:B\cap C=\emptyset} m_1(B)m_2(C) \in [0,1]$. The denominator is a coefficient of normalization. In particular, if it is null, it means that there is a total conflict between the sources, and aggregation is then impossible. Thus, the use of this rule is valid only when the sources are sufficiently in agreement. Moreover, this rule is suitable when two distinct beliefs are always certain and treated equivalently. When these are not the case, another method to incorporate this notion has to be investigated. One way is to use Dempster's rule of conditioning [13].

2) Conditioning Rules for Updating Belief: The conditional approach for updating evidence allows to condition/update the already available partial evidence bases. Subsequently, only that portion of incoming evidence that is relevant is used for updating the existing evidence base. The well-known conditioning rule in the theory is Dempster's rule of conditioning [13], which is derived as a special case of DRC. When the BBAs in the DRC is given as $m_1(\cdot) = m(\cdot)$ and $m_2(C) = 1$, the DRC is transformed into the following

$$m(A|C) = \frac{\sum_{A=B\cap C} m(B)}{1 - \sum_{B\cap C=\emptyset} m(B)} \tag{12}$$

In [14], the authors generalize the rule of conditioning by differentiating two cases: conditioning rule with unreliable condition and conditioning rule with uncertain condition. The basic conditioning rule assumes the following:

The given condition is exactly true information given temporally after the prior belief. It puts a strict restriction on focal elements of the posterior belief [14].

First, the index λ ($0 \leq \lambda \leq 1$) is introduced to represent reliability of the condition. When $\lambda < 1$, the condition is not completely true. Based on [14], we implement the Dempster's conditioning rules for updating belief using the following algorithm:

Algorithm 1 Dempster's conditioning rules

Input: decisions coming from streams of data and event-based data processing units
Output: $m(B|A)$
for all roads **do**
  for all each data point at t **do**
    Define the reliability parameter λ based on the estimated features.
    if $B \supset A \neq \emptyset$ **then**
      $m(B|A) = \sum_{A=X\cap B} \lambda m(X) + (1-\lambda)m(A)$
    else if $A = B$ **then**
      $m(B|A) = \sum_{A=B\subseteq X} \lambda m(X) + (1-\lambda)m(B) + \sum_{X\cap B=\emptyset} \lambda m(X)$
    else if $A \not\subseteq B, A \neq B, A \neq \emptyset$ **then**
      $(m(B|A) = 1-\lambda)m(A)$
    else
      $m(B|A) = 0$
    end if
  end for
end for

In this work, tweets are considered as the new information that conditions the evidence coming from the streams of data module. In relation with algorithm 1, these data are denoted by A and B respectively.

IV. Big-Data-Based Traffic Flow Prediction

In the proposed big-data-based traffic prediction framework, heterogeneous sources of data are categorized into two types: streams of data and event-based data. Data that are categorized as streams of data are data that are continuously generated by sensors, i.e., traffic flow and weather data. Event-based data refer to data that are generated based on occurring events, i.e., social media data.

Figure 3 shows the overall steps of our road traffic flow prediction approach. First, DBNs are used to predict traffic flow from streams of data and event-based data separately. The predicted traffic flows then are clustered using the K-Means clustering method and fused using DSET.

A. Traffic Flow Prediction Using DBNs

This section presents the development of traffic prediction modeling using DBNs. In the data streams module as shown in Figure 3, traffic flow and weather data are established as the main sources of information. The traffic flow and weather data is normalized to [0,1]. One component of the collected weather

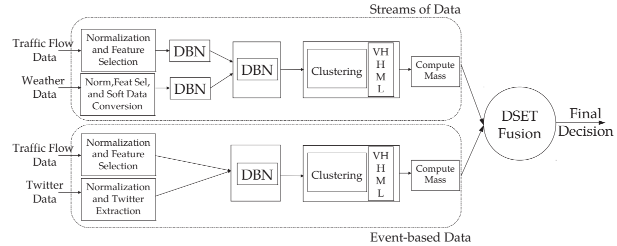

Fig. 3: Architecture of the big data-driven traffic flow prediction framework.

data is the weather condition, which is considered as soft data generated by operators. Although the weather condition is considered as soft data, it will be treated as streams of data since it is available continuously at every sampling time. Weather condition consists of 24 unique values, which are represented by:

$$\text{weath\_cond} = \{clear, cloudy, foggy, rain, snow, overcast, smoke, ...\} \tag{13}$$

In order to incorporate the weather conditions in the prediction model, the soft data representation is re-defined as an integer number in the range of $[1, 10]$, for instance $clear = 1$, $cloudy = 2$.

Feature selection is done by auto- and cross-correlation methods. The auto-correlation analysis determines the number of previous steps of traffic flow that are incorporated in the input of the prediction models. The original traffic flow values exhibit significant correlation even up to five hours in the past. Besides, to properly select the most weather factors that shape traffic flow behavior, cross-correlation coefficient between each pair of weather variables and traffic flow is calculated. Weather variables that have small correlation coefficient with respect to traffic flow or large correlation coefficients with respect to the highest cross-correlation coefficient are discarded. The selected features then are fed to the predictors.

The predictor used in our big data-driven traffic prediction is DBNs [12]. DBNs can be used as a predictor or a data fusion method. In the multi-task part of the DBNs, namely the last layer of the DBNs, a linear activation function is used instead of some non-linear activation ones. In the proposed framework, the traffic flow and weather data are predicted separately using DBNs. Hence, the k_1-step in the past traffic flow data are trained using k_2-step in the future traffic flow data as the output. Similarly for the weather data, the current weather data are trained using k_2-step in the future traffic flow data as the output. The result of each the predictors is then merged using data fusion. In the data fusion step, the decisions coming from both predictors are trained using DBNs with the actual traffic flow at k_2-step in the future as the output.

In the event-based data module as depicted in Figure 3, tweets are established as the source of information. However, based on our experiments, Twitter data can not stand alone without the help from traffic flow data. Therefore, the traffic flow data are included in the first step. Subsequently, the Twitter data are pre-processed from soft data to hard data by means of annotation and extraction. Since the scope of this work does not include the social media data processing, tools of annotation and extraction provided in [15] are used.

B. Traffic Flow Clustering and Mass Assignment

The aim of this step is to cluster the traffic flow based on the level of intensity. The traffic flow data are clustered into four clusters: Low (**L**), Medium (**M**), High (**H**), and Very High (**VH**) traffic. The method used to cluster the data is the well-known K-Means clustering. This latter aims to partition the data into K clusters in which each observation belongs to the cluster with the nearest mean or center. In the training, K-Means algorithm assigns each data point to a cluster and adjusted the centers of the clusters until they converge. Next, the output of clustering are used to compute the masses of each sample.

The following assumption is used to compute the masses:

Assumption 1: The prediction results of the predictors are assumed to be accurate enough, such that the misclustering can occur only in the neighboring clusters.

This assumption is important since it emphasizes that the confusion of prediction occurs only in the boundary between clusters. This statement is true if we have acceptable prediction accuracy results, which are the case in our work. Following this assumption, after the distances of a sample to the centers are calculated, the two lowest distances are selected. The selected two distances then are converted into some confidence values defined as:

$$C(d) = \exp\left(-\alpha d\right) \tag{14}$$

where $C(d)$ is the confidence value, $d (0 \leq d \leq 1)$ is the distance of a sample to a centre, and $\alpha (0 \leq \alpha \leq 1)$ is used to control the weight of the exponential function. This function translates the distance to confidence value non-linearly. If the distance of a sample to a centre is small, it can be said with high confidence that the sample belongs to the centre. The α can be used to control the significance of the distance. Once the confidence values of the two smallest distances are computed, the difference of the confidence values are calculated. There are two possible cases for the difference values: the difference is within a specified small tolerance value, i.e., there is a confusion in the cluster assignment, or the difference is quite large, i.e, the sample belongs to the cluster with the highest confidence. In the former case, the confusion can be modelled by considering the boundary of the two neighboring clusters as an evidence. These cases determine the number of focal elements in the power set. Based on the number of clusters, the number of elements in the power set is $2^4 = 16$. Consider the following example: the confidence of cluster L is $C(d_L) = 0.41$, and the confidence of cluster M is $C(d_M) = 0.4$. If the difference tolerance is 0.2, then the confidence of the the element LM is computed by:

$$C(d_{LM}) = C(d_L) + C(d_M) - |C(d_L) - C(d_M)| = 0.8 \tag{15}$$

While the summation term denotes the confidence value assigned to the boundary of the cluster, the absolute difference term denotes the uncertainty in the boundary. The three confidence values then are marginalized to 1. Hence, the final confidence values are

$$C(d_L) = 0.2547, \quad C(d_M) = 0.2484, \quad C(d_{LM}) = 0.4969 \tag{16}$$

In this illustrative example, the confidence values of the *low traffic* and *medium traffic* clusters are adjusted, and a confidence value is assigned to the boundary. The other elements in the power set are assigned zero masses. In the second case, only the two highest confidence values are considered. These values are then marginalized to 1, and other elements in the power set are assigned zero masses. Our method of mass assignment is illustrated in Algorithm 2.

Algorithm 2 Mass assignment algorithm

```
Input: two smallest distances
Output: set of masses
Initialization: difference tolerance, α
for all data points do
  Compute the the distance difference |d1 − d2|
  for i = 1 : 2 do
    C(di) = exp(−α di)
  end for
  if |d1 − d2| ≤ difference tolerance then
    C(d12) = C(d1) + C(d2) − |C(d1) − C(d2)|
    marginalize C(d12), C(d1), C(d2) into masses m1, m2, m12
    m_others = 0
  else
    marginalize C(d1), C(d2) into masses m1, m2
    m_others = 0
  end if
  return m_all
end for
```

C. Dempster-Shafer Conditional Rules for Updating Belief

To fuse decisions coming from both streams of data and event-based data processing units, Dempster's conditioning rules for updating belief is used. One of the main characteristics of event-based data is that they are not continuously available. In this work, the unavailable data at certain time steps are estimated using Principal Component Analysis with Alternating Least Square (PCA-ALS) algorithm. PCA with missing values is studied in [16]. The estimated Twitter data are considered to be unreliable. Hence, λ will be used as a reliability indicator in the streams of data and event-based data fusion.

The implementation of this data fusion follows Algorithm 1, where the mass of an event conditioned by another event is computed. Specifically for this work, prediction coming from event-based data module is considered as the new information that conditions the fundamental evidences provided by the streams of data processing.

V. Experiments and Results Analysis

To investigate the merits of the proposed big-data-based traffic flow prediction framework, real-world datasets are used in the experiments. The San Francisco, Bay area is chosen as the case study to perform traffic flow prediction and data fusion using DSET. In the following, the datasets and scenario settings are described.

A. Datasets

As illustrated in Figure 3, three datasets are needed to perform traffic flow prediction and data fusion using respectively DBNs and DSET in the Bay Area, District 4. In this experiment, the following database are selected:

- Caltrans Performance Measurements Systems (PeMS) [17] to obtain traffic flow measurements. There are 47 roads, where the traffic flow of each road is calculated based on the average of all the loop detectors in that particular road. Traffic data are aggregated into 5-minute duration by PeMS. Furthermore, based on the recommendation of Highway Capacity Manual 2010 [18], data then are aggregated into 15-minute duration.
- MesoWest project [19] to collect Weather data. 16 National Weather Service (NWS) network stations scattered throughout the district provide data about the temperature, visibility, wind gust and general weather condition. The weather data are generated every one hour and then re-sampled into 15-minute duration for the purpose of alignment with the traffic flow data.
- CityPulse Dataset Collection [20] to obtain event reports and tweets over four months. Traffic related events are then extracted and annotated using techniques proposed in [21].

In this work, the traffic flow, weather, and Twitter data are collected in the weekdays and weekends from August 1 to November 25th, 2013. We use the data of August to October for the training and November for the testing.

B. Experimental Settings

To show the merit of our proposed big-data-driven traffic flow prediction framework, the following scenarios are introduced:

1) Traffic flow prediction using historical traffic flow data only.
2) Traffic flow prediction using historical traffic flow and weather data.
3) Traffic flow prediction using historical traffic flow, weather and traffic related Twitter data. To highlight the advantage of fusing streams of data and event-based data while mitigating the unreliability issue of Tweets, two methods of fusing beliefs are investigated:
   - Dempster-Shafer basic combination rule
   - Conditioning rules for updating belief

To find the suitable configuration of the DBNs, namely the number of hidden units and layers, 10-fold cross validation is performed in the training data. The performance of different scenarios is evaluated using two performance indices: Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE). These indices are defined as:

$$\mathrm{RMSE} = \left(\frac{1}{T} \sum_{t=1}^{T} (y_t - \hat{y}_t)^2 \right)^{\frac{1}{2}}$$

$$\mathrm{MAE} = \frac{1}{T} \sum_{t=1}^{T} |y_t - \hat{y}_t| \tag{17}$$

where y_t and $\hat{y}_t$ are the actual and predicted traffic flow at time t respectively. These indices are used to measure the linear score that averages the error with the same weight, and to measure the residuals by assigning larger weights to larger errors. These measures are represented by MAE and RMSE respectively.

C. Traffic Flow Prediction Using DBNs

The selection of DBNs as our predictor is justified by experimenting the data using other predicting techniques such as ARIMA and Artificial Neural Networks (ANNs). Table I shows that DBNs produce better performance than ARIMA and ANNs, in terms of RMSE and MAE. Hence, DBNs are selected as the main predictor in this work.

TABLE I: Performance of DBNs compared to other predictors

	ARIMA	ANN	DBN
RMSE	0.2643	0.0790	0.0701
MAE	0.2114	0.0583	0.0487

DBNs require several parameters that should be defined and tuned. The hidden layer numbers are varied from 2 to 5 layers. The number of nodes in each layer is chosen from 50 to 300 in the step of 50. The number of epochs is crucial to the learning phase. Therefore, the epochs range from 50 to 500 in the step of 50 are varied. The best architecture for DBNs consists of 3 hidden layers, 250, 200, 100 hidden units in the first, second, and third hidden layers respectively. The best number of epoch of the DBNs training is found to be 100 epochs. Another cross-validation process is implemented to find the best configuration of DBNs using weather data.

After the decisions about traffic flow coming from both traffic and weather data predictors are obtained, a decision fusion algorithm using DBNs is implemented. For illustration purpose, the profile of predicted traffic flow values using decision-level data fusion based on DBNs versus the ground truth for the three representative freeways (17S, 87N and 15E) are plotted in Figure 4. From this figure, it can be seen that

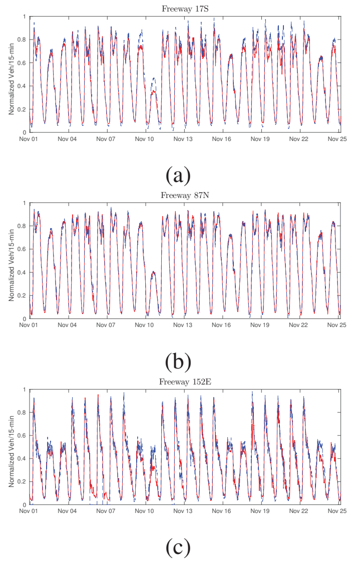

Fig. 4: Traffic flow prediction of freeway with various traffic volume. Freeway with low (a), medium (b), and high (c) traffic flow. (blue: ground truth, red: predicted traffic flow)

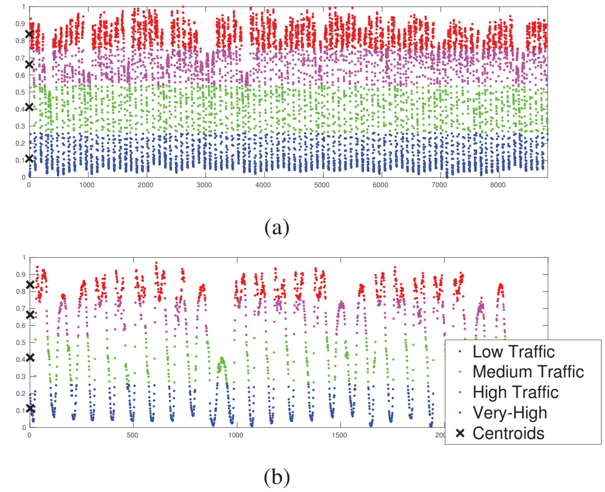

Fig. 5: Traffic clustering of road 87N. (a) training data (b) testing data

using traffic history enhanced with weather data as input for predictor better tracks the traffic flow pattern in medium traffic flow (87N freeway). This result is attributed to the fact that from 47 roads, the majority of them belongs to medium traffic state.

In the event-based data module, DBNs use traffic flow and tweets as input. The DBNs parameters are tuned as follows: the layer size from 2 to 5 layers are chosen. The number of nodes in each layer is chosen from $[50, 300]$ in the step of 50. The number of epochs is crucial to the learning phase. Therefore, the epochs range from 50 to 500 in the step of 50 are varied. The best architecture for DBNs consists of 3 hidden layers, 250, 225, 150 hidden units in the first, second, and third hidden layers respectively. The best number of epoch of the DBNs training is found to be 150 epochs.

D. Traffic Flow Clustering Results

After the prediction results of both the streams of data and event-based units are obtained, the next step is to cluster the traffic into four levels of intensity: L, M, H and VH. From here, the masses of each levels including the boundaries are calculated using the method proposed in section IV-B.

Figure 5 depicts the clustering results of the training and testing data. The testing data clustering is conducted using the centroids obtained from the training data clustering. The centroids will be used also in the predicted traffic flow clustering. The testing data clustering results will be used as the ground truth in the next experiment.

E. Traffic Prediction Results Using Traffic, Weather, and Tweets

In the following experiment, the performance of final decision fusion is investigated. Specifically, the traffic predictions coming from the two predictors are fused using DBNs. Dempster-Shafer based fusion methods are used to perform the final fusion of the streams of data and event-based data decisions.

The Dempster's conditional updating algorithm is implemented and compared to the classical Dempter's rule of combination (DRC). In the Dempster's conditional updating algorithm, the reliability parameter λ is tuned using cross-validation until the best result is acquired. The results of the individual predictions as well as final decisions after using DSET are presented in Table II.

The table shows that when the classical DRC is used, the performance of the combination of both modules, i.e., streams of data and event-based data modules, are worse than those of the individual modules. Indeed, using streams of data as input to DBNs provides higher accuracy (82%) than under event-based data (80.72%) or the classical DRC (80.31%). The DRC acts like an averaging method for both of the hard and soft evidences. This is in sharp contrast to Dempster's conditional updating rules. Note that Dempster's conditional updating reaches 88.91% accuracy while classical DRC achieves only 84.44%. This result is achieved when λ is small, i.e., $\lambda = 0.13$, which means that the information coming from event-based data module is considered not too reliable. Hence, Dempster's conditional updating makes the main decision about traffic flow from streams of data. Event-based evidence is only taken into account if it supports the predicted evidence and does not compromise its integrity.

Finally, the final decision produced by the Dempster's conditional updating is compared to the ground truth for road $87N$ (medium traffic). The merit of our approach is confirmed in Figure 6. It can be seen that the traffic flow is accurately predicted although there is some discrepancies that mostly occurs in the boundary between clusters.

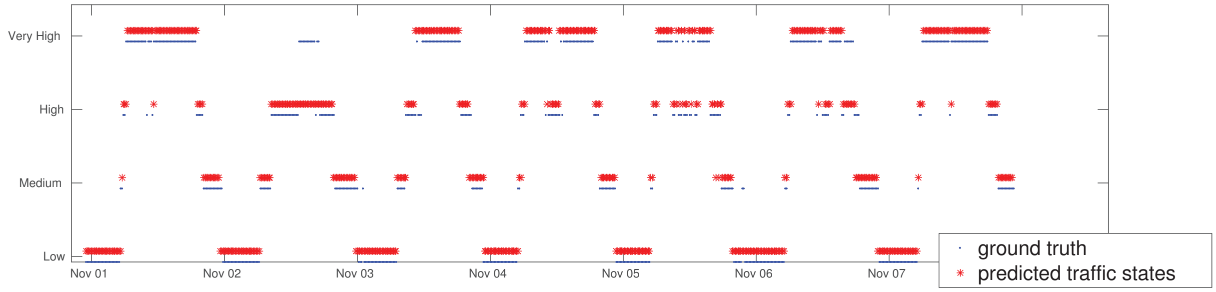

Fig. 6: Final predicted traffic states using DSET for road 87N

TABLE II: Final decision statistical measures

	Streams of Data	Event-Based Data	Classical DRC	Dempster's Conditional
Accuracy	86.73%	83.24%	84.44%	88.91%
Precision	82.25%	80.72%	80.31%	85.96%
Sensitivity	86.61%	83.50%	84.38%	89.89%
Specificity	84.01%	81.11%	82.12%	85.63%
G-Mean	84.12%	81.67%	82.61%	86.23%
F1-Meas	83.82%	81.17%	82.95%	86.33%

VI. Conclusion

In this paper, the short-term traffic flow prediction using Big Data originated from various and heterogeneous sources is investigated. DBNs are used to predict road traffic flow from heterogeneous input data: streams of data, i.e., historical traffic flow and weather data, and event-based data extracted from tweets in the region of San Francisco, Bay area. To tackle this problem, Dempster's conditional rule for updating belief is implemented to fuse decisions coming from streams of data and event-based data to achieve more accurate traffic flow prediction. Several scenarios outlining possible real-world configurations are introduced to highlight the merit of the proposed framework. Finally, it is shown that using Dempster's conditioning rules for updating belief outperforms the basic Dempster-Shafer combination rule.

References

[1] Transport Outlook 2012: Seamless Transport for Greener Growth. Organisation for Economic Co-operation and Development. Accessed June 20, 2015 .

[2] Cambridge Systematics. *Traffic Congestion and Reliability: Linking Solutions to Problems: Executive Summary*. Federal Highway Administration, 2004.

[3] Federal Highway Administration. Alternate Route Handbook. Available from $http : //ops.fhwa.dot.gov/publications/ar_handbook/arh1.htm$. Accessed June 20, 2015 .

[4] Cisco. The Internet of Everything is the New Economy. http://www.cisco.com/c/en/us/solutions/collateral/enterprise/cisco-on-cisco/Cisco_IT_Trends_IoE_Is_the_New_Economy.html, 2015. [Online; accessed 0-8 January-2016].

[5] Hubert Rehborn and Micha Koller. A study of the influence of severe environmental conditions on common traffic congestion features. *Journal of Advanced Transportation*, 48(8):1107–1120, 2014.

[6] S. Dunne and B. Ghosh. Weather adaptive traffic prediction using neurowavelet models. *Intelligent Transportation Systems, IEEE Transactions on*, 14(1):370–379, March 2013.

[7] Yunjie Zhao, Adel Sadek, and Daniel Fuglewicz. Modeling the impact of inclement weather on freeway traffic speed at macroscopic and microscopic levels. *Transportation Research Record: Journal of the Transportation Research Board*, 2272:173–180, 2012.

[8] Jingrui He, Wei Shen, Phani Divakaruni, Laura Wynter, and Rick Lawrence. Improving traffic prediction with tweet semantics. In *Proceedings of the Twenty-Third international joint conference on Artificial Intelligence*, pages 1387–1393. AAAI Press, 2013.

[9] Lei Lin, Ming Ni, Qing He, Jing Gao, Adel W Sadek, and Transportation Informatics Tier I Director. Modeling the impacts of inclement weather on freeway traffic speed: An exploratory study utilizing social media data. *to appear in Transportation Research Record: Journal of Transportation Research Board*, 2015.

[10] Yisheng Lv, Yanjie Duan, Wenwen Kang, Zhengxi Li, and Fei-Yue Wang. Traffic flow prediction with big data: A deep learning approach. *Intelligent Transportation Systems, IEEE Transactions on*, 16(2):865–873, April 2015.

[11] Wenhao Huang, Guojie Song, Haikun Hong, and Kunqing Xie. Deep architecture for traffic flow prediction: Deep belief networks with multitask learning. *Intelligent Transportation Systems, IEEE Transactions on*, 15(5):2191–2201, Oct 2014.

[12] Geoffrey E Hinton, Simon Osindero, and Yee-Whye Teh. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554, 2006.

[13] Glenn Shafer et al. *A mathematical theory of evidence*, volume 1. Princeton university press Princeton, 1976.

[14] Koichi Yamada, Vilany Kimala, and Muneyuki Unehara. A new conditioning rule, its generalization and evidential reasoning. In *IFSA/EUSFLAT Conf.*, pages 92–97, 2009.

[15] Pramod Anantharam, Payam Barnaghi, Krishnaprasad Thirunarayan, and Amit Sheth. Extracting city traffic events from social streams. *ACM Transactions on Intelligent Systems and Technology*, 9(4), 2014.

[16] Bjørn Grung and Rolf Manne. Missing values in principal component analysis. *Chemometrics and Intelligent Laboratory Systems*, 42(1):125–139, 1998.

[17] http://pems.dot.ca.gov/.

[18] Highway Capacity Manual. Volumes 1-4.(2010). *Transporation Research Board*, 2010.

[19] http://mesowest.utah.edu/.

[20] http://iot.ee.surrey.ac.uk:8080/.

[21] Pramod Anantharam, Payam Barnaghi, Krishnaprasad Thirunarayan, and Amit Sheth. Extracting city traffic events from social streams. *ACM Trans. Intell. Syst. Technol.*, 6(4):43:1–43:27, July 2015.