12 years of mining road traffic data

Ian Marsh and Ahmad Al-Shistawy and Cosar Ghandeharioon and Björn Bjurling

Abstract Predicting traffic congestion is an important tool for users, authorities, fleet management companies, and road planners. Typically road traffic authorities know the long-term demands on road sections, however, the shorter-term prediction is a research problem. That said, algorithms, data, and processing have advanced to a point where new road analysis can be done, which is very much the topic of this contribution. This paper applies big data practices on 12 years of data from the Swedish road traffic authority to predict traffic flow congestion around the city of Stockholm. We present a simple transparent model to detect traffic build up in space and time, and hence find the end of any queue. An important insight in this work is the use of traffic density as a different measure for congestion detection, rather than the average velocity or flow values.

Ian Marsh
RISE SICS
Γ ian.marsh@ri.se

Ahmad Al-Shistawy
RISE SICS
Γ ahmad.al-shistawy@ri.se

Cosar Ghandeharioon
KTH
Γ cosarg@kth.se

Björn Bjurling
RISE SICS
Γ bjorn.bjurling@ri.se

ARCHIVES OF DATA SCIENCE, SERIES A
(ONLINE FIRST)
KIT SCIENTIFIC PUBLISHING
Vol. -, No. -, -

DOI 10.5445/KSP/XXXXXXXX/XX
ISSN 2363-9881

1 Introduction

1.1 What is a traffic queue?

To define unambiguously what a queue is not straightforward. Quantitative single metrics such as "greater than 100 vehicles in a 1 km road section" or "lane occupancy of more than two-thirds" are plausible. Naturally, combinations of metrics can narrow down the notion of a traffic queue. Subjective formulations such as "this journey is taking 1 hour longer than normal" might also indicate that drivers ran into queues. Mathematical formulations such as "we are the negative gradient of the flow-density relation" indicate that the density of vehicles has reached a point where the flow rates decrease due to the number of vehicles present. We explore our current thoughts in the future work section, however it is not straightforward to say what a queue is, and more difficult still to ascertain where its end might be.

1.2 Queue detection applications

Except the often stated applications of traffic congestion such as diversion advice, traffic planning, pollution countermeasures, we consider improved journey planning. Note, GPS is not congestion aware, it uses only location and average velocity and can give poor estimates of arrival times when stuck. An application we are focusing on is fleet management where haulage companies deal with consignments of loads. Typically, these are signed off when the last one arrives at is its intended destination. Therefore, detecting queues, possibly if a single truck enters one, as well as the dissipation rates are important.

1.3 Why do we need 12 years of data?

Depending on the need for the data, different timescales of data might be needed:

1. Daily prediction needs information about hourly events
    - E.g. commuting and non-commuting hrs

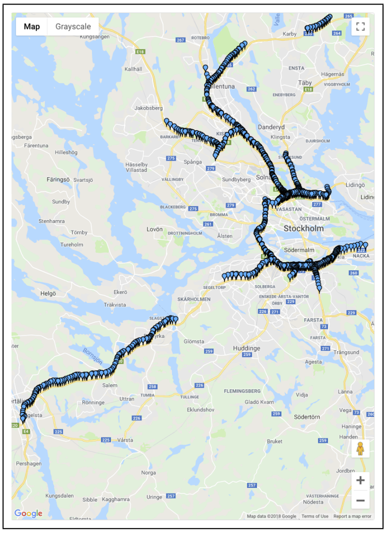

Fig. 1 The road section considered in this paper: E4N Stockholm, northbound, 50 km, 1-4 lanes, 147 radar sensors, live version at `http://resmonster.se/dash/s/pages/mcs.php`.

2. Weekly prediction needs information about daily events
   - E.g. Mon-Fri and weekends
3. Monthly predictions needs information about weekly events
   - E.g. Working days and holidays
4. Yearly prediction needs information about seasonal events
   - E.g summer or winter

Therefore, we might need many years of data, if not only as above, but to look in previous months or years for similiar events. A concrete example is the road traffic on Black Friday for congestion over several years.

2 Related work

2.1 Projects

Most of the German car industry, academia plus government joined forces in the *UR:BAN* project, 2012-15 for Economic Affairs and Energy (2012). They looked at routing through metropolitan areas using information at the network level. Up to that point, decisions had been based on static data such as travel time curves and rudimentary knowledge of the current traffic state. A close integration of intelligent infrastructure with vehicles as a prerequisite for reducing emissions and enhancing traffic efficiency along a route.

Microsoft used multi-year data to infer and forecast traffic flow in the ClearFlow project Microsoft (2012). The work leverages machine learning to build services that make use of both live streams of sensed information and large amounts of heterogeneous historical data. This has led to multiple prototypes and real-world services such as traffic-sensitive directions now used in Bing Maps. Their seminal work stimulated new efforts in related areas, such as privacy and routing Horvitz et al (2012). More recently, systems are running in Berlin, using a recent API developed for Flink implementing the connected car event stream Artisans (2017).

2.2 Publications

On the more academic level, deep learning is a form of machine learning which provides good short-term forecasts of traffic flows by exploiting the dependency between space and time of flows. Polson uses Bayesian learning to correct the estimation bias that is present in the model with fixed parameters using methodology on road sensor data from the Interstate I-55 near Chicago Polson and Sokolov (2015). They describe a methodology to update the posterior uncertainty for the critical density and capacity parameters and develop a deep learning model to predict traffic flows in Polson and Sokolov (2017). Their linear model is fitted using l1 regularisation and a sequence of `tanh` activation layers. They predict traffic congestion for two different events; a Chicago Bears football game and an extreme snowstorm event, both from using 40 minutes of historic data.

3 Three large datasets: Traffic flow, location, weather

3.1 Traffic flow

The primary dataset we use in this work is from road sensors. It basically consists of flow and speed data, sent per minute at locations around Stockholm and Gothenburg. The Stockholm section was shown in the problem statement. A typical road sensor uses radar signals, mounted as shown in figure 2. A summary of the road data we used is in table 1. An example of the Mcs CSV data format (plus an added column) is below.

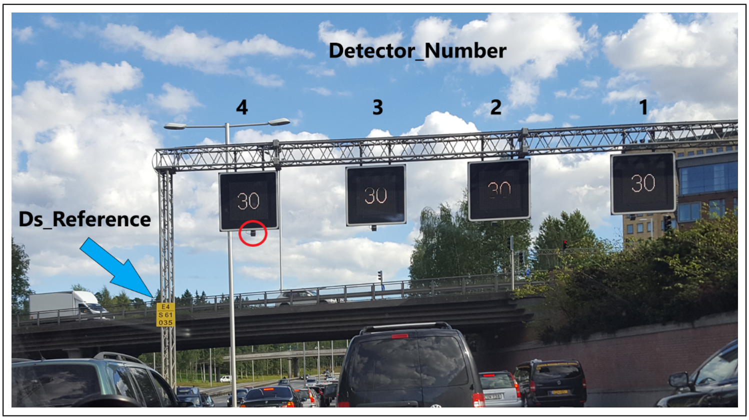

Fig. 2 Swedish flow sensors, Ds_reference is the road name + distance from the road start in meters.

	1 road (E4N)	**1 month**	**All**
Duration	1 month	1 month	12 years
Period	Nov. 2016	Nov. 2016	Jan. 2004 - Dec. 2016
Size on disk	1 Gigabyte	5 Gigabytes	390 Gigabytes
Entries	20 million	88 million	7 billion
No. of Detect.	147	2040	2059

Table 1 Summary of the traffic flow data from the Swedish road authority.

```
df_E4N_D.show(10)
+--------------------+------------+---------------+-------+----------------+-----------+
|Timestamp           |Ds_Reference|Detector_Number|Flow_In|Average_Speed   |Density    |
+--------------------+------------+---------------+-------+---------------=+-----------+
|2016-11-01 00:00:00| [E4N,48290]|               3|      8|            82  |5.85      |
|2016-11-01 00:00:00| [E4N,48935]|               3|      4|            72  |3.33      |
|2016-11-01 00:00:00| [E4N,49370]|               1|      1|            77  |0.77      |
+-------------------+------------+----------------+-------+----------------+----------+
```

To verify the metadata we plotted all detectors on a Google map. We extracted the coordinates from the Spark DataFrame into a local Python list object. Next, the coordinates were converted into a GeoJson polygon and shown. Specifically, we used the Python Leaflet package. Hops `www.hops.io` is our chosen platform, as it offers "dataset as a service" as many users require weather and location as meta-information for analysis. Additionally, GPU support, multi-tenancy and in-house support makes it an attractive platform.

3.2 File formats and performance

File formats significantly affect CPU, memory and disk usage and the processing time. In this case with 12 years of data and nearly 400 Gigabytes of data, the format and hence processing time become relevant.

Options include i) text/binary encoding, ii) row/column storage, iii) compression/splitability, and iv) the schema. We compare and evaluate three common file formats, CSV, JSON, and Parquet. Plain text is portable across different applications and human readable, however requires more storage space and processing requirements as it needs to be parsed, split, and usually converted. Binary encodings are preferred for many data intensive applications.

Traffic flow applications obviously benefit from compression; disk space, moving data in the cluster. Compression reduces the load time into memory, as applications are I/O bound. Zip and gzip are non splittable, i.e. cannot be decompressed in parallel. Large files are stored in a distributed file system (HDFS) and processed by splitting the file and processing each piece (block) in parallel. Container formats, sequence Files, Avro, or Parquet split and indexes data into blocks and compressed individually. Generally, a lightweight compression is preferred for data intensive applications.

Listing 1 Code for Query 1 applied to a parquet file format

```
# Query 1 – Parquet
spark.read.parquet('/path/to/my/data')
.select('Road', 'Km_Ref', 'Detector_Number')
.distinct().count()
```

Listing 2 Code for Query 2 applied to a parquet file format

```
# Query 2 –Parquet
q2 = spark.read.parquet('/path/to/my/data')
.select('Flow_In', 'Average_Speed')
.where('Status == 3 AND Road == "E4N"')
.withColumn('Density',
col('Flow_In')*60/col('Average_Speed'))
.select('Density')
.rdd.flatMap(lambda x: x)
.histogram(list(range(0,55,5)))
```

Format	Size (MB)	% of original	Convert & save time (secs)	Notes
Original CSV	5064	100	-	Row, txt, no schema
Original zip CSV	480	10	-	Row, txt, no schema, comp.
Cleaned CSV	4937	97	162	Row, txt, no schema
Json	20919	413	410	Row, txt, schema
Parquet	175	3	166	Col, binary, schema, comp.

Table 2 Comparing the storage efficiency of three different file formats storing the same data of one month (Nov 2016). Running on a single server with Spark in stand alone mode.

File	Query 1		Query 2	
Format	Avg. (sec)	% of CSV	Avg. (sec)	% CSV
CSV	23	100%	29	100%
Json	95	413%	101	354%
Parquet	3	14%	13	46%

Table 3 Comparing the computation efficiency of three different file formats for executing two different queries. Running on a single server with Spark in stand alone mode.

3.3 Location

Location is important for any traffic situation, as the flow is dependent on both time and location, as well as weather conditions discussed next. To plot the data on a map we need the GPS coordinates of the detector. This metadata exists in a separate Geographic Information System (GIS), which was exported into a CSV file and then converted from coordinates in the Swedish standard

File	Query 1		Query 2	
Format	Avg. (sec)	% of CSV	Avg. (sec)	% CSV
CSV	23	100%	29	100%
Json	30	413%	101	354%
Parquet	6	14%	6	46%

Table 4 Comparing the computation efficiency of three different file formats for executing two different queries. Running on a cluster with Spark, Yarn, and HDFS

(SWEREF99) to the GPS standard (WGS84) for plotting. Later in the location inference section of this paper we show how better positioning can be obtained.

3.4 Weather

Clearly the weather has a pivotal role in traffic flow, hence cross-referencing weather data is always needed. The Swedish Meteorological and Hydrological Institute (SMHI) has provided daily forecasts to the general public with societal weather functions since 1874. The open data comes with licensing terms specified in Creative Commons Attribution 4.0. SMHI's weather forecasts are presented in the media, `smhi.se` and via their SMHI app. Importantly for us, they provide an API to their databases. There is a REST API which can be queried thus:

```
GET /api/version/version/parameter/parameter/station/station/period/period/data.ext
```

The period can be one of latest-hour, latest-day, latest-months or corrected-archive. A particular station can be queried or a set. Parameters can be specified, the return type, JSON, XML or CSV as well as up to nearly 300 types of weather conditions, e.g. "High and tight snow drift, strong winds, thunderstorm". A full list is at `http://opendata.smhi.se/apidocs/metobs/codes.html`.

4 Density, Flow and Velocity

For the correlation between velocity and flow, over a week see 3. The dataset contains average velocity between sensors and the flow, which is the number of vehicles passing the sensor per minute. If the density was linearly related with the flow (or velocity) then the density would be simply the quotient of the flow / average speed. Since they are not, then density can be obtained from macroscopic flow diagram, which shows the relationship between these three characteristics of road traffic. One of the complicating factors of traffic flow is the lane effects, that is i) drivers change lanes, lanes are added and removed from main roads, iii) the occupancy, traffic mix (types of vehicles) are lane dependent and iv) lanes can be arbitrarily renamed.

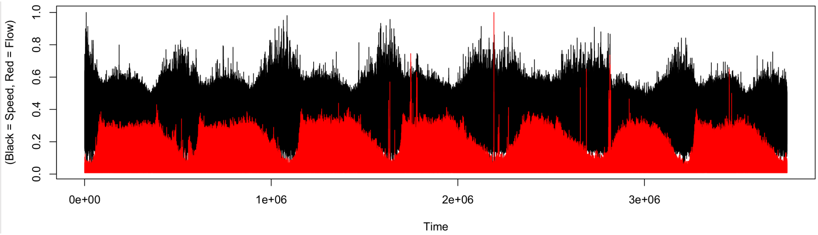

Fig. 3 Formalised velocities and flows over 7 days (a correlation coefficient of -0.38)

By density we mean the number of vehicles per road length. Seven reasons for traffic density over flow or average speed are:

1. Average flow rates are recorded by the road authorities and average speed by authorities and "seen" by individual drivers. However, both do not necessarily detect congestion.
2. Density is intuitive, we can think of the number of vehicles per road section.
3. Speed is dependent on the vehicle type: e.g trucks are slower than cars.
4. Density can be compared between road sections, countries, etc.
5. Average speed and flow can be difficult or sensitive to local conditions.
6. Density also provides values for both instantaneous and average values.
7. Speed has different units km/h or miles/h.

In some cases, queues accumulate in less than 300 meters, therefore we need extra inference to obtain such queue buildups. Basically higher resolution information than the Ds reference (the distance between sensors) is needed. Space does not permit its derivation in this paper, but density changes are obtained from the PDE relationship between flow-density-velocity, for example see figure 4 and in Marsh (2012).

Figure 5 shows how densities vary along a busy road in Stockholm, where the black lines are the density changes and the red an average over 10 samples. Where the density changes more abruptly than measurement points (Ds), it

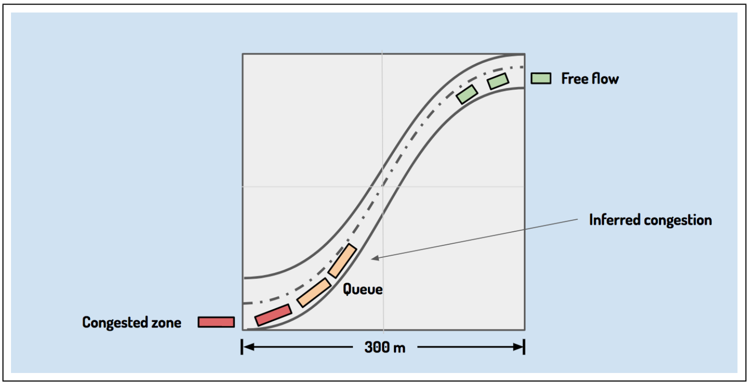

Fig. 4 Density can be inferred using a fluid flow model.

is a better indicator of congestion, with the proviso of inference, e.g. using techniques in the last paragraph.

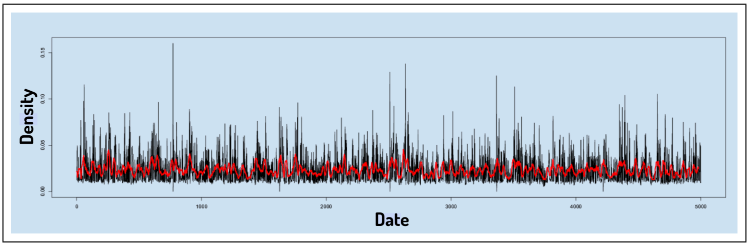

Fig. 5 Density changes along a road section in Stockholm.

5 Lane effects

Lane effects play an important role in traffic theory and practice. In most works one cannot ignore them (despite many works do). This can be seen in Figure 6. We use the kernel density estimate to avoid binning problems associated with histograms, but acknowledge the velocities are discretised to km/hour in steps of 1km/hr. Summary statistics of the same lane effects are in table 5. Data such as this is better represented as rolling in time or distance as a function of the variable of interest, we have created such a visualisation at

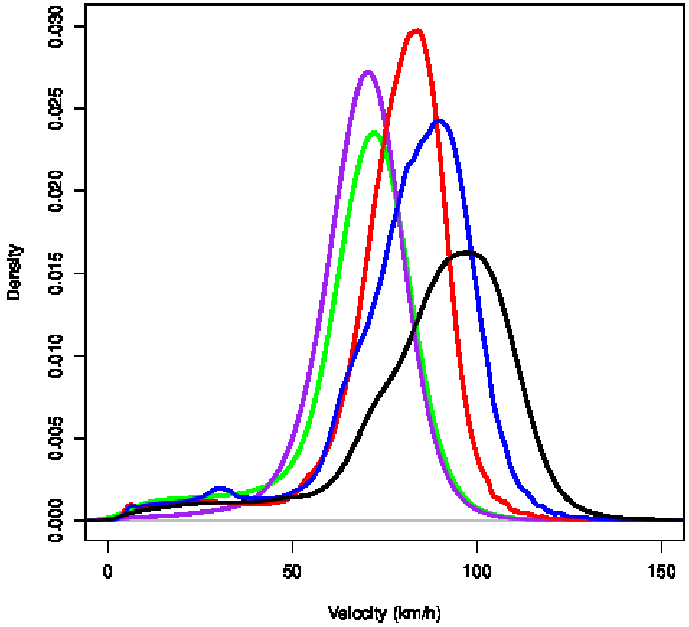

Fig. 6 Lane speed kernel densities. Slowest/Inside Lane 1:Purple, Lane 2:Green, Lane 3:Red, lane 4:Blue and Lane 5:Black the fastest/outside lane.

`https://goo.gl/yiQpxP` captures the velocity and flow, as a time series, discrete histogram and density estimate, essentially a journey along the E4.

Lane	Min.	1Q	Med.	Mean	3Q	Max.
1	2	79	92	89	103	231
2	2	73	84	81	93	243
3	2	72	80	78	87	232
4	2	63	71	68	77	205
5	2	64	70	69	75	153
6	8	51	56	57	62	150

Table 5 Average lane speeds in km/hour on a major road in Stockholm (E4N). Lane 1 is the rightmost lane in left hand drive countries. Note when the mean and median differ this means the distributions are skewed.

6 Machine learning congestion

In this paper, we use Long Short-Term Memory (LSTM), one of the better known deep learning models for time series prediction, it has been used on language sequences as well as time series (in general) Olah (2015). LSTMs address a problem with Recurrent Neural Networks which face the vanishing gradient problem. Essentially, the vanishing gradient problem is a difficulty found in training the network with gradient-based learning methods and back-propagation. The neurons receive an update proportional to the partial derivative of the error function with respect to the current weight in each iteration of training. This can become prohibitively small and hence training fails.

In our setting, before applying LSTM, differencing technique is used to make the data stationary, then it is changed to a supervised learning, and finally it is scaled [-1, 1]. To have an optimum model in LSTM, many experiments had to be done using a grid search.

By adding layers of neurons in the deep model, additional transitions can be predicted and by adding neurons at each layer, the training time can be reduced. Increasing epochs, leads to finding more transitions while increasing the training time. Based on our data, the optimum model in LSTM, has 5 hidden layers of 10 LSTM neurons, and one dense layer for the output. The loss functions we used were RMSE and used the Adam optimizer. The outcome of applying LSTM on full dataset and resampled dataset is depicted in Figure 7.

To train this model, it requires about 7 minutes on a standard laptop. Note this example uses the average speed as the road value to predict. This was due to the thesis which compared different approaches, ARIMA, a new deep model called Neural Decomposition (ND) for average speed. However, we believe the density gives the best indicator of congestion, as argued for earlier in the paper.

7 Discussion

Explicit and implicit effects: Gathering data is always non-trivial in a vehicle setting. Firstly the explicit parameters need to be decided (speed, density, flow) as well as the traffic mix (cars, trucks, others). Also the road layout: lanes, friction significantly influence queue accumulation. Other implicit factors, influence the data, and may not be seen until comparisons of the data can done (typically after event). *Data quantities:* The fundamental road traffic relation-

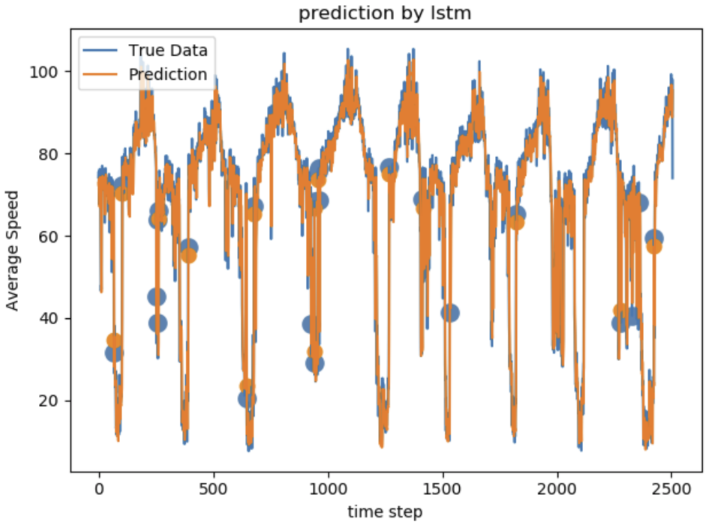

Fig. 7 Learned and real speeds from the LSTM model (an RMSE of 7.1).

ship seems to hold some validity from our data, however a significant amount of data is needed. We have shown where a small and larger amount of data has been available. This is another argument for having more, not less data. *Ground truth:* In the End of Queue detection, there is no real ground truth. Different people might say congestion starts and ends (if any) depending on their experience. Clear cut cases can be easily identified, but there are many cases which are not easy to identify, label universally as “the end” of the queue. *Driving style:* Situations where lane changes occur frequently can cause jumps in the density predictions, particularly using a Macroscopic Fluid model approach.

8 Conclusions

The main result of this work is a concept to implementation of analysis of traffic flows. In particular, we have looked at detecting congestion in road traffic. We have taken a large traffic set from the Swedish road authority, designed a congestion detection algorithm. We have compared an ARIMA model, based on classic time series and a deep learning technique, LSTM, to predict queue buildups. Furthermore, we implemented the algorithms in a cluster environment in northern Sweden, giving some indicators of the running time. Despite the positive results of this work, there are many factors, not included in the modelling, which are the weather, merge in/out points on the road (we chose a simple point on the road, not a complex junction). Finally, there is the point of **explainable AI** although the deep model predicts quite well the speeds in future, there is some issues around having a model that cannot be explained. An ARIMA model also works well, but has the advantage of being interpretable and may be preferable for a real deployed. A deep model can be used off-line to show some improvements can be achieved.

This paper has been made available by the funding program FFI: Strategic Vehicle Research and Innovation (DNR 2016- 02207), which is administered by VINNOVA, the Swedish Government Agency for Innovation Systems.

References

Artisans D (2017) Connected car project. URL http://training.data-artisans.com/exercises/connectedCar.html

for Economic Affairs GM, Energy (2012) Ur-ban

Horvitz E, Apacible J, Sarin R, Liao L (2012) Prediction, expectation, and surprise: Methods, designs, and study of a deployed traffic forecasting service. CoRR abs/1207.1352, URL http://arxiv.org/abs/1207.1352, 1207.1352

Marsh I (2012) VANET communication: A traffic flow approach. In: PIMRC, IEEE, pp 1043–1048

Microsoft (2012) Predictive analytics for traffic. URL https://www.microsoft.com/en-us/research/project/predictive-analytics-for-traffic/

Olah C (2015) Understanding lstm networks

Polson N, Sokolov V (2015) Bayesian analysis of traffic flow on interstate i-55: The lwr model. Ann Appl Stat 9(4):1864–1888, DOI 10.1214/15-AOAS853, URL https://doi.org/10.1214/15-AOAS853

Polson N, Sokolov V (2017) Deep learning for short-term traffic flow prediction. arXiv:160404527v3