Scalable Selective Traffic Congestion Notification

Győző Gidófalvi
Division of Geoinformatics
Deptartment of Urban Planning and Environment
KTH Royal Institution of Technology, Sweden
gyozo@kth.se

ABSTRACT

Congestion is a major problem in most metropolitan areas. Systems that can in a timely manner inform drivers about relevant, current or predicted traffic congestion are paramount for effective traffic management. Without loss of generality, this paper proposes such a system that by adopting a grid-based discretization of space, can flexibly scale the computation cost and the geographic level of detail of traffic information that it provides. From the continuous stream of grid-based position and speed reports from vehicles, the system incrementally derives 1) statistics for detecting directional traffic congestions and 2) model parameters for a time-inhomogeneous, Markov jump process that is used to predict the likelihood that a given vehicle will encounter a detected directional congestion within the notification horizon. A simple but efficient SQL-based prototype implementation of the system that can naturally be ported to Big Data processing frameworks is also explained in detail. Empirical evaluations on millions of object trajectories show that 1) the proposed movement model captures the topology of the underlying road network space and the directional aspects of movement on it, 2) the congestion notification accuracy of the system is superior to a linear movement model based system, and 3) the prototype implementation of the system (i) scales linearly with its input load, notification horizon and spatio-temporal resolution and (ii) can in real-time process 1.14 million object trajectories.

Categories and Subject Descriptors

H.2.8 [**Database Applications**]:
Data mining, Spatial Databases and GIS

General Terms

Algorithms, Performance

Keywords

Trajectory Data Mining, Congestion Detection and Notification, Floating Car Data, LBS, Intelligent Transport Systems

MobiGIS'15, November 03-06, 2015, Bellevue, WA, USA

DOI: http://dx.doi.org/10.1145/2834126.2834134

1. INTRODUCTION

Congestion is a major problem in most metropolitan areas. Systems that can in a timely manner inform drivers about relevant, current or predicted traffic congestion are paramount for effective traffic management tasks like congestion-avoidance routing. Provided the current and growing availability- and the task-specific utility of Floating Car Data (FCD), this paper proposes a data-driven approach and a directional grid-based, time-inhomogeneous, Markov jump process model for the detection congestion and the selective dissemination of this congestion information to vehicles. The paper also discusses in detail a scalable implementation of the method using off-the-shelf relational database technology. Empirical evaluations on millions of real-world object trajectories showed that (1) the selective dissemination accuracy of the proposed model is superior to a linear movement based model, (2) the proposed can effectively capture the topology of the underlying road network and the directional object movement on it, and (3) a simple, high-level SQL-implementation of the proposed model is scalable in terms of its input size and model parameters.

The unique features of the proposed model and method as well as the contributions of the paper are as follows:

1. *Grid-based model*: The proposed grid-based modeling without loss of generality has several advantages. First, it does not necessitate the need for complete and accurate road network information and map matching. Second, the model can easily be scaled to any geographical (spatio-temporal) level of detail.
2. *Markov jump process model*: The proposed method directly estimates the probability of a future location of an object which is not prone to error propagation as the common used conventional Markov chain model is that estimates the probability of future locations indirectly through intermediate locations.
3. *Representation of direction*: The proposed model effectively represents of the direction of movement and flow within the grid-based framework.
4. *Time-inhomogeneous model*: The proposed method defines and extracts parameters for normally observed traffic conditions and object mobility for different days of the week and different hours of the day.
5. *Adoption of novel congestion definition*: The proposed method defines a grid cell to be congested if (i) there is enough evidence for this, (ii) the evidence is reasonably unanimous, (iii) the evidence is statistically different from the normally observed traffic conditions, and

(iv) the measured traffic conditions reflected in the evidence are relatively annoying for the drivers compared to the normally observed traffic conditions.

6. *Simple, scalable and portable implementation*: The proposed prototype uses off-the-shelf RDBMS technology, which can be implemented in a few lines of SQL code and can be ported to Big Data processing frameworks.
7. *Relevant performance evaluations*: The prediction accuracy and scalability of the system is evaluated for predicting the locations of relevant subset of the objects for a relevant set of spatio-temporal events.

The rest of this paper is organized as follows. Section 2 presents the related work. Section 3 defines the problem of directional congestion detection and selective notification. Section 4 presents the proposed method its simple prototype implementation. Section 5 presents the evaluations and Section 6 concludes and points to future research directions.

2. RELATED WORK

Considerable research has been conducted to extract and use the regularities in object movement to predict future movement of objects.[1] Two popular extraction and prediction methods emerged: discrete-time Markov model based [4, 5,12,16,19] and sequential rule / trajectory pattern based [6, 10, 11, 15, 23, 25, 27, 28]. The methods can also be classified based on what information is used to model the movement of objects into methods with (1) a general model for all objects [6, 10, 11, 15, 16, 19, 23, 25], (2) a type-based model for similar (types of) objects [4, 28], or (3) a specific model for each individual object or set of individual objects [5, 12, 19, 27]. Alternatively, they may be classified according to their definition of Regions Of Interest (ROIs) for prediction and consequently their spatial and temporal scale and granularity into methods using (1) application-specific ROIs (road segment, network cell, sensors etc.) [4, 6, 11, 16, 19, 23], (2) density-based ROIs [5, 10, 12, 15, 25, 27, 28], or (3) grid-based ROIs [6, 10, 12, 23, 25]. Finally, they can be classified according to their prediction provision into methods that provide (1) only sequential spatial predictions (location of next ROI) [4, 5, 19, 28] or (2) spatio-temporal predictions [6, 10–12, 15, 16, 23, 25, 27] and into methods that provide (i) time-continuous [6, 11, 15, 16, 19] or (ii) time-punctuated [10, 12, 23, 25, 27] predictions. Other prominent approaches are [18, 20], in which predictions rely on the assumption that the observed, short-term, partial trajectory of an object is part of a (approximately [20]) shortest path to the future unknown destination of the object.

In comparison, the herein proposed method (1) models movement as a discrete-time, time-inhomogeneous Markov jump process that directly estimates the probability possible future locations within the prediction horizon, (2) has general model for all objects, (3) uses a grid to define its application-specific ROIs, and (4) provides spatio-temporal movement predictions implicitly as all its predictions are valid for the short-term prediction horizon. Additional notable features of the proposed method are that (1) it models directional movement and flows using a spatial grid-based decomposition of space and (2) the grid-based model makes the method network-data independent and geographically scalable; yet the directional grid-based Markov model adequately captures the topology of the underlying road-network and the purposeful, directional movement on it.

[1]The following is a summary of a recent classification [12] of such methods with only some of the most relevant references.

3. DEFINITIONS

Let $\mathcal{G}$ denote a *grid* with grid cells $g_1, g_2, \ldots$ with side length *glen* that uniformly partitions the 2D Euclidean space $\mathbb{R}^2$. Let the time domain be denoted by $\mathbb{T} \equiv \mathbb{N}_0$. Let $O = \{o_1, \ldots, o_M\}$ be a set of moving objects, i.e., vehicles, that periodically send timestamped *status reports* which reference grid cells in the grid $\mathcal{G}$. Let a status report r be one of three *types* (*position_update*, *speed_update*, or *stopped*) and contain the following information: the object ID of the object o that submits the report, the timestamp t of the report, the grid cell g that the object o is inside at time t, the speed s of object o at time t, the direction *dir* from which object o enters grid cell g, and the type T of the report. Let $\mathcal{SSR}$ denote the time-ordered stream of status reports, i.e., an unbounded ordered sequence $\langle r_1, r_2, \ldots \rangle$ of status reports such that for all $i > 1$, $r_i.t \geq r_{i-1}.t$. Let $\mathcal{SSR}_{t_c}$ denote the subsequence of $\mathcal{SSR}$ that contains all the reports up to time instance t_c. Then the directional congestion detection and selective notification tasks are defined as follows:

Definition 1. Directional Congestion Detection: Given a sequence of status reports $\mathcal{SSR}_{t_c}$ for a set of objects O up to time t_c and a *temporal analysis window size* $\Delta t_{awin} \in \mathbb{N}^+$, find all *directional congestions* (g, dir), i.e., grid cell–direction combinations, such that the speeds of the objects that have entered grid cell g from direction *dir* during the temporal analysis window $[t_c - \Delta t_{awin}, t_c)$ is significantly and substantially bellow "normal".

Definition 2. Selective Directional Congestion Notification: Given a stream status reports $\mathcal{SSR}$ for a set of objects O, a set of directional congestions $C = \{c_1 = (g_1, dir_1), \ldots, c_m = (g_m, dir_m))\}$ that are detected from $\mathcal{SSR}_{t_c}$ for a temporal analysis window size Δt_{awin}, and a *prediction horizon length* $\Delta t_{pred} \in \mathbb{N}^+$, using only information in $\mathcal{SSR}_{t_c}$, selectively notify an object $o_i \in O$ about a directional congestion $c_j = (g_j, dir_j) \in C$ if an only if, according to $\mathcal{SSR}$, the object o_i enters the grid cell g_j from the direction dir_j during the *prediction horizon* $[t_c, t_c + \Delta t_{pred})$.

The above congestions concepts and related tasks can be relaxed by omitting the direction requirements from the definitions; these relaxed concepts and tasks are referred to with the *non-directional* adjective. For brevity the adjectives "directional" and "selective", if they can be inferred from the context are dropped from names of the above concepts and tasks. The above tasks can naturally also be extended for the entire stream of status reports by performing the congestion detection and notification tasks periodically.

4. CONGESTION NOTIFICATION

As it is foreshadowed in the definitions, the proposed methodology adopts a grid-based discretization of space, which by changing the resolution of the grid allows the system to scale in terms of its computation cost (time and storage) and the geographical level of detail of traffic information that it manages. Given this grid-based framework, the outline of the directional congestion detection and selective directional congestion notification method is as follows.

1. Map the directional movement / flow of objects in $\mathcal{SSR}$ to the grid-based framework.
2. Form tumbling windows over the mapped input stream and treat them as temporal analysis windows.
3. Extract *Current Directional Flow Statistics* (CDFS) and *Current Directional Mobility Statistics* (CDMS) from the *Recent Trajectories* (RT) that are within the current tumbling / temporal analysis window.
4. Incorporate the CDFS / CDMS into *Historical Directional Flow / Mobility Statistics* (HDFS / HDMS) for different temporal domain projections.
5. Detect a grid cell g to be congested from a particular direction *dir* if the current mean speed of vehicles that have entered the grid cell g from the direction *dir* is significantly and substantially below the normal according to the temporally relevant HDFS.
6. Notify an object o about a detected directional congestion (g, dir) if, based on HDMS and the current position and movement direction of o, the likelihood that o will enter the grid cell g from the direction *dir* during the prediction horizon is greater or equal than a user / system defined *minimum notification probability threshold min_prob*.

The subsequent paragraphs present theoretical and implementation details of the above described processing stages.

4.1 Grid-based Directional Flow Statistics

While the grid-based discretization of space is inherently non-directional, the movements and flows of objects on the underlying road network is inherently directional. To capture the directional aspects of movement and flow, the proposed model defines directional movement and flow in terms of a grid cell and its eight immediate cell neighbors. Specifically, the proposed model / method, without loss of generality, assumes that grid-based trajectories are spatially contiguous and defines the direction *dir* for entering a grid cell g from one of g's eight immediate neighboring grid cells n as the positive angle in degrees from n to g with respect to North. Using this definition, the proposed method for each observed grid cell–direction combination (g, dir) extracts three basic directional flow statistics from the grid-based trajectories of objects: the number-, the average speed-, and the standard deviation of the speeds of objects that enter grid cell g from direction *dir*.

4.2 Grid-based Directional Mobility Statistics

Similarly, using the same directional grid-based movement definition, the proposed method extracts two basic directional mobility statistics from the grid-based trajectories of objects: 1) the number of objects $n_{(g_s,dir_s)\rightarrow(\cdot,\cdot)}$ that enter grid cell g_s from direction dir_s and subsequently proceed to another not necessarily neighboring grid cell and 2) the number of objects $n_{(g_s,dir_s)\rightarrow(g_d,dir_d)}$ that enter grid cell g_s from direction dir_s and subsequently enter another not necessarily neighboring grid cell g_d from direction dir_d.

4.3 Stream Processing Model

The online processing of $\mathcal{SSR}$, i.e., the directional congestion detection and selective notification, is facilitated by adopting a commonly used temporal sliding window model:

Definition 3. Temporal Sliding Window Model: Given a stream of ordered time-stamped elements, $\mathcal{S} = \langle (e_1, t_1), (e_2, t_2), \ldots \rangle$, and temporal sliding window parameters, *window size*, $t_{wsize} \in \mathbb{N}$, and *window stride*, $t_{wstride} \in \mathbb{N}$, the Temporal Sliding Window Model (TSWM) at every *window slide* time instance, $t_{slide} = a \times t_{wstride} + t_{wsize}$ where $a \in \mathbb{N}^0$, processes the elements of the stream that are within the time interval of the *window* $(t_{slide} - t_{wsize}, t_{slide}]$. Consequently, a TSWM is defined by the pair $SW = (t_{wsize}, t_{wstride})$.

Given the above definition of the TSWM, the stream of status reports $\mathcal{SSR}$ is processed in *tumbling windows*, according to $SW = (t_{wsize}, t_{wstride} = t_{wsize})$, as follows. At every window slide / tumble time the current tumbling window is equated to the temporal analysis window, i.e., $\Delta t_{awin} = t_{wsize}$, and is used to perform the directional congestion detection- and selective notification tasks based on the current and the long-term, historical directional flow and mobility statistics. Given the fact that the directional flow statistics are derived from windows of size $t_{wsize} = \Delta t_{awin}$, i.e., the statistics are implicitly assumed to be valid for a Δt_{awin}-long period in the future, i.e., the period of the succeeding tumbling window is treated as the *prediction horizon*, constituting an implicitly assumed *short-term congestion prediction model*.

4.4 Incremental Historical Summary Statistics

To be able to efficiently extract long-term, historical directional flow statistics from the stream of status reports $\mathcal{SSR}$, the proposed method takes advantage of the fact the CDFS that are extracted from tumbling windows are based on non-overlapping subsets X and Y of $\mathcal{SSR}$ and hence can be combined in an incremental fashion according to the following equations [26]:

$$\mu_{X\cup Y} = \frac{n_X \mu_X + n_Y \mu_Y}{n_X + n_Y} \tag{1}$$

$$\sigma_{X\cup Y} = \sqrt{\frac{n_X \sigma_X^2 + n_Y \sigma_Y^2}{n_X + n_Y} + \frac{n_X n_Y}{(n_X + n_Y)^2}(\mu_X - \mu_Y)^2} \tag{2}$$

where n, μ, and σ denote the size-, mean- and standard deviation of a given sample. Using Equations 1 and 2, the CDFS are incrementally combined and compressed into long-term Historical Directional Flow Statistics (HDFS). The distributive count-measures of CDMS are incrementally combined and compressed into long-term Historical Directional Mobility Statistics (HDMS) using simple summation.

4.5 Temporal Domain Projections

Human mobility exhibits a large degree of regularity [24] that movement models try to capture. There are at least three different types of regularities in movement: temporal, periodical, and sequential [12]. To capture these potential regularities in object flows and movements, the proposed method extracts HDFS and HDMS for different values of the day-of-week and hour-of-day temporal domain projections.

4.6 Directional Congestion Detection

Let $(\dot{n}, \dot{\mu}, \dot{\sigma})$ and $(\bar{n}, \bar{\mu}, \bar{\sigma})$ respectively denote the CDFS and HDFS of a given grid cell g from a given direction *dir*. Then, the proposed method defines and detects the grid cell g as being congested from direction *dir* when all of the following four criteria are satisfied:

1. Sample size criterion: $\dot{n} >= min\_veh$
2. Sample dispersion criterion: $\dot{\sigma}/\dot{\mu} < max\_cv$
3. Statistical power criterion: $(\dot{\mu} - \bar{\mu})/(\bar{\sigma}/\sqrt{\dot{n}}) < max\_z$
4. Speed difference criterion: $(\dot{\mu} - \bar{\mu})/\bar{\mu} < max\_relspddiff$

In other words, the criteria require that the recent status reports for grid cell g from direction dir are sufficiently many (1), and the reported speeds in them are in close agreement with one another (2), are (according to a z-test) significantly- (3) and substantially relatively (4) lower than the historical ("normal") speeds. To account for the time-inhomogeneity of directional flow statistics, in addition to using the global (atemporal) HDFS, the above criteria are also separately evaluated using the HDFS for the day-of-week and the hour-of-day of the CDFS for (g, dir), and the direction congestion (g, dir) is detected if the criteria hold for any of the global-, the day-of-week projected-, or the hour-of-day projected congestion models.

4.7 Directional Congestion Notification

To selectively notify objects about a detected congestion a movement model is needed. The proposed method uses a grid-based, directional movement model based on a Markov *jump* process model for which the HDMS store the parameter estimates. The adjective "jump" is emphasized to differentiate from a more conventional Markov chain process model that given the current state s_i estimates the probability of a future state s_{i+k} for $k > 2$ indirectly through states $s_{i+1}, \ldots, s_{i+k-1}$, whereas the Markov jump process model estimates the probability of s_{i+k} directly. It is conjectured, that this direct estimation is more accurate as it is not prone to error propagation. The subsequent sections define a mobility statistics based- and a simpler, linear movement model based congestion notification criterion (for baseline comparison) as follows.

4.7.1 Mobility Statistic Criterion (MSC)

Notify an object o about a directional congestion (g_d, dir_d) if o has currently entered a grid cell d_s from a direction dir_s and the conditional probability of an object entering the grid cell g_d from direction dir_d given that the object has previously entered the grid cell d_s from the direction dir_s is greater or equal than a user / system defined *minimum notification probability threshold* min_prob, i.e:

$$\frac{n_{(g_s, dir_s) \rightarrow (g_d, dir_d)}}{n_{(g_s, dir_s) \rightarrow (\cdot, \cdot)}} \geq min\_prob. \quad (3)$$

A non-directional variant of the MSC, which is also evaluated in Section 5.5, is defined by omitting the directional constraints from Equation 3.

4.7.2 Linear Movement Criterion (LMC)

Notify an object o about a directional congestion (g_d, dir_d) if the cosine of the *heading offset*, i.e., the angle between the *general heading* of o defined as the direction from o's least recent historical grid position g_h to o's current grid position g_s, and the direction of the congestion relative to the current grid cell of o, i.e., the direction from g_s to g_d, is greater or equal than a user / system defined *minimum heading offset cosine* min_cos and the distance between o and the congested grid cell g_d is smaller or equal than a user / system defined *maximum notification range* max_r. A min_cos value of 1 represents a situation when the general heading of the object is directly pointing towards the congested grid cell, while a value of 0 represents a situation when the general heading is perpendicular to relative direction of the congested grid cell. To be able to compare LMC with MSC, in the experiments max_r is set to the maximum number of grid cells an object can theoretically move during a temporal analysis window / prediction horizon assuming a maximum object speed of 50 m/sec.

4.8 SQL-based Implementation

A prototype system that performs the described congestion detection task can be conveniently and effectively implemented using the power of off-the-shelf Relational Database Management Systems (RDBMS), e.g., PostgreSQL, and the simplicity of declarative programming languages, e.g., SQL. The paragraphs bellow explain the details of such a prototype implementation whose performance is empirically evaluated in Section 5. The aims of the detailed explanation are to illustrate the simplicity of the proposed solution and to highlight the portability of the proposed solution to *Big Data* processing paradigms that in a scalable manner support the basic relational algebra operators, e.g., MapReduce-based data processing frameworks like Apache™ Hadoop® [2] and main-memory, streaming variants like Apache Spark™ [3].

4.8.1 Relational Database Schema

The prototype implementation stores recent trajectories and current and historical directional flow and mobility statistics in the following five database tables:

- `RT = <oid, seqnr, dgid, spd>`
- `CDFS = <dgid, nr, mu, sig, nr_suc>`
- `CDMS = <dst_dgid, src_dgid, nr_src2dst>`
- `HDFS = <dgid, nr, mu, sig, nr_suc>`
- `HDMS = <dst_dgid, src_dgid, nr_src2dst>`

The information stored in the five tables are as follows. The `RT` table records the status reports that have been received from the clients during the most recent tumbling window. More specifically, a row in `RT` stores the information that at the time of the report the vehicle with object ID `oid` entered the grid cell with grid cell coordinates (`gx`, `gy`) in the direction `dir` —which is uniquely encoded as the integer concatenation of the three values (`gx`, `gy`, `dir`) into a *directional grid ID* `dgid = gx_gy_dir`— with the speed `spd`. In addition, the prototype implementation assumes[2] that consecutive status reports from a given vehicle that refer to the same grid cell (i.e., an initial *position_update* status report is followed by one or more *speed_update* status reports) are aggregated into one status report that has the timestamp and speed information of the most recent *speed_update* status report. Furthermore, status reports that preceded a *stopped* status report of a vehicles are excluded from the most recent tumbling window and `RT`, i.e., records of stopped objects do not contribute in congestion detection and mobility prediction and these stopped vehicles are not subject to notification of any congestion. To model the sequential

[2]A scalable implementation of the described window semantics can be implemented using appropriate data stream processing frameworks, e.g., Apache Flink [1], that provides flexible windowing semantics where window boundaries and content can also be defined based on any custom user defined logic. The prototype implementation employs a custom driver program that emulates the windowing of the status report stream.

nature of a trajectory, RT stores a sequence number seqnr that denotes the relative position of the status report within the grid-based trajectory of the vehicle oid that is inside the most current tumbling window, i.e., the row / record that contains the most current element of the grid-based trajectory has seqnr = 1 and the row / record that stores the n-th most current element has seqnr = n. In an operational setting, all the information in RT can be calculated by the clients of the system (i.e., a software on a position aware computing device, e.g., navigation system or mobile phone, in the vehicles) provided some conventions for grid-based trajectory reporting.

The CDFS table stores for each directional grid ID dgid = gx_gy_dir the number of vehicles nr and the mean mu and standard deviation sig of the speeds of these vehicles that, during the current tumbling window, have entered the grid cell (gx, gy) in the direction dir. While it is logically unrelated, but because it makes the computation of conditional probabilities of the proposed movement model computationally efficient, the table CDFS for each directional grid ID dgid also stores in nr_suc the number of occurrences that dgid is succeeded by some other directional grid ID in the partial grid-based trajectories of objects during the current tumbling window, i.e., in RT. The CDMS table stores 1) the number of vehicles nr_src2dst that, during the current tumbling window, have moved from the source directional grid ID src_dgid to the destination directional grid ID dst_dgid. Finally, the HDFS and HDMS tables store long-term, historical aggregates of the statistical values of the CDFS and CDMS tables, respectively.

With the exception of the columns spd, mu, and sig, which are of type float, all other columns in the tables are of type int or bigint. Unlike in conventional relational table schema notation, in the above list the underlining denotes that the given columns have a hash index, or in the case of the column seqnr a B-tree index, to speed up the join, selection, and aggregation operations during the processing of the queries that implement the directional congestion detection and notification tasks[3]. It is once more worth to emphasize the design choice for the column dgid. As explained before, dgid contains the unique concatenation of the planar / projected grid coordinates gx and gy and the direction of movement dir that results in an integer. Effectively, given that all subsequently described queries that implement the directional congestion detection task only involve equijoins on dgid, the 1-dimensional hash index on dgid efficiently indexes information about movement/flow in the 2-dimensional space.

4.8.2 Calculation of CDFS and CDMS

As it can be seen in code listings SQL 1 and SQL 2, both the CDFS and CDMS are computed based on simple aggregations of a single source of information, namely, the recent grid-based partial trajectories of the vehicles in table RT. SQL 1 and SQL 2, as well as all SQL-code in the subsequent sections, show the bodies of SQL functions that at definition time the Query Planer and Optimizer (QPO) of the RDBMS compiles into executable query plans. During the processing of each tumbling window, the plan for SQL 1 is executed and its results are stored in the table CDFS, as described in Section 4.8.1. The logic implemented in SQL 1 is straight forward: the query groups all recent status reports in the current tumbling window by the directional grid ID dgid and for each dgid selects the corresponding CDFS that conform the table schema of CDFS and its application semantics that are described in Section 4.8.1.

[3]Indexes on the table RT are dropped and recreated before and after the status reports of the new tumbling window are inserted into RT.

SQL 1 FUNCTION calc_CDFS()

```
1  SELECT dgid, count(*) AS nr, avg(spd) AS mu,
2         COALESCE(stddev(spd),0) AS sig
3  FROM RT
4  GROUP BY dgid;
```

SQL 2 FUNCTION calc_CDMS()

```
1  SELECT dst.dst_dgid, src.src_dgid,
          count(*) AS nr_srs2dst
2  FROM (SELECT oid, seqnr, dgid AS dst_dgid
3        FROM RT) AS dst,
4       (SELECT oid, seqnr, dgid AS src_dgid
5        FROM RT) AS src
6  WHERE dst.oid = src.oid
7    AND dst.seqnr < src.seqnr
8  GROUP BY dst.dst_dgid, src.src_dgid;
```

SQL 3 FUNCTION ud_HDFS()

```
1  UPDATE HDFS AS gh
2  SET nr = (c.nr+gh.nr),
3      mu = (c.nr*c.mu+gh.nr*gh.mu)/(c.nr + gh.nr),
4      sig = sqrt((gh.nr * gh.sig^2 + c.nr * c.sig^2) /
5                 (gh.nr + c.nr) +
6                 (gh.nr * c.nr * (gh.sig - c.sig)^2) /
7                 (gh.nr + c.nr)^2),
8      nr_suc = (c.nr_suc+gh.nr_suc)
9  FROM CDFS AS c
10 WHERE gh.dgid = c.dgid;

11 INSERT INTO HDFS (dgid, nr, mu, sig, nr_suc)
12 SELECT c.gid, c.dir, c.nr, c.mu, c.sig
13 FROM CDFS AS c
14 LEFT JOIN HDFS AS gh
15 ON (gh.dgid = c.dgid)
16 WHERE gh.dgid IS NULL;
```

SQL 2 computes the CDMS that conform the table schema of CDMS and its application semantics that are described in Section 4.8.1, i.e., for each directional source grid cell src_dgid and for each directional destination grid cell dst_dgid, it returns the number of vehicles nr_src2dst that moved from src_dgid to dst_dgid. In particular, the query, based on a self join (Lines 2-7) and a grouping operation (Line 8), counts the number of occurrences nr_src2dst of (dst_dgid, src_dgid)-combinations where the directional destination grid cell dst_dgid succeeds the directional source grid cell src_dgid (Line 6) in the recent partial grid-based trajectories in RT.

4.8.3 Incremental Calculation of HDFS and HDMS

As described in Section 4.8.1, the table HDFS stores long-term, historical aggregates of the statistics of the CDFS table. These historical statistics are, according to the formulas in Section 4.4, incrementally updated in two phases: first statistics for previously observed directional flows are incrementally updated, then statistics for previously not ob-

SQL 4 FUNCTION CongCell(min_veh, max_cv, max_z, max_relspddiff)

```
1  SELECT c.dgid AS dgid
2  FROM HDFS AS gh, CDFS AS c
3  WHERE gh.dgid = c.dgid
4    AND c.nr >= min_veh
5    AND c.sig / c.mu < max_cv
6    AND (c.mu - gh.mu) / (gh.sig / sqrt(c.nr)) < max_z
7    AND (c.mu - gh.mu) / gh.mu < max_relspddiff;
```

SQL 5 FUNCTION CongNotif(min_veh, max_cv, max_z, max_relspddiff, min_notif_prob)

```
1  WITH cond_prob AS
2       (SELECT m.src_dgid, m.dst_dgid,
3               m.nr_src2dst::float / f.nr_suc AS cond_p
4        FROM HDMS m, HDFS f
5        WHERE m.src_dgid = f.dgid)
6  SELECT t.oid, c.dgid AS con_dgid
7  FROM cond_prob AS gcp, RT AS t,
8       CongCell(min_veh, max_cv,
9                max_z,max_relspddiff) AS c
10 WHERE t.seqnr = 1
11   AND gcp.src_dgid = t.dgid
12   AND gcp.dst_dgid = c.dgid
13   AND gcp.cond_p >= min_prob;
```

SQL 6 Non-directional conditional probabilities

```
1  SELECT m.src_gid, m.dst_dgid,
2         m.nr_src2dst::float/f.nr_suc AS nr_src2any
3  FROM (SELECT src_dgid/10 AS src_gid, dst_dgid,
4               sum(nr_src2dst) AS nr_src2dst
5        FROM HDMS
6        GROUP BY src_dgid/10, dst_dgid) m,
7       (SELECT dgid/10 AS gid, sum(nr_suc) as nr_suc
8        FROM HDFS
9        GROUP BY dgid/10) f
WHERE m.src_gid = f.gid;
```

served directional flows are recorded. These two phases are illustrated in SQL 3. In particular, the `UPDATE`-query (Lines 1-10), according to Equations 1 and 2 updates the statistics (Lines 2-8) for the previously observed directional flows in `HDFS` for the directional flows that are also found in `CDFS` (Line 10). Subsequently, the `INSERT`-query, based on a left join between tables `CDFS` and `HDFS` selects the currently observed directional flows and statistics from `CDFS` that are *not* present (Lines 15-16) among the previously observed directional flows in `HDFS` and inserts them into `HDFS`. The computation of incremental HDMS is implemented using analogous `UPDATE-INSERT` sequence of operations for the previously observed vs. previously not observed mobility mobility patterns and statistics.

4.8.4 Calculation of Directionally Congested Cells

The CongCell(min_veh, max_cv, max_z, max_relspddiff) function in SQL 4, provided the current- and the long-term, historical directional flow statistics (Line 2), as the proposed methodology suggests in Section 4.6, identifies all directional grid cells `dgid` (Line 1) where 1) the sample size criterion- (Line 4), 2) the sample dispersion criterion- (Line 5), 3) the statistical power criterion- (Line 6), and 4) the speed difference criterion (Line 7) are satisfied.

4.8.5 Calculation of Directional Congestion Notifications

Finally, the function CongNotif(min_veh, max_cv, max_z, max_relspddiff, min_prob) in SQL 5, based on the conditional probabilities of directional grid cells that are derived from the long-term HDMS in `HDMS` and the information stored in the `nr_suc` column of the table `HDFS` (Lines 1-5 and 7), the recent trajectories (Line 7), and the identified directional congestions (Lines 8-9), as the proposed methodology suggests in Section 4.6, the function CongNotif(min_veh, max_cv, max_z, max_relspddiff, min_prob) notifies every vehicle `oid` that is currently located in a directional grid cell `src_dgid` (Lines 10 and 11) from which the conditional probabilities suggest that the vehicle will encounter a detected congestion within the prediction horizon, i.e., the vehicle will move from the directional source grid cell `src_dgid` to the congested directional destination grid cell `dst_dgid` (Lines 11 and 12) with a probability that is larger or equal than the user / system defined parameter `min_prob` (Line 13).

4.8.6 Alternative movement models

As it is described in Section 4.7, in addition to the directional mobility statistics based criteria presented in Section 4.8.5, directional congestion notifications can be sent out according to a number of different criteria. The implementations of the proposed alternative criteria are as follows. A simple derived implementation of the non-directional mobility statistics criterion calculates the conditional probabilities from non-directional *source* grid cells by appropriately aggregating and relating the long-term HDFS and HDMS based on non-directional source grid IDs that are derived through integer division, i.e., `gid = dgid/10`, and replacing Lines 1-5 in SQL 5 with SQL 6. A simple prototype implementation of the LMC accesses a grid cell's coordinates from its grid ID by integer division and implements the minimum heading offset cosine- and maximum notification range criteria as join conditions between the recent trajectories and the detected directional congestions.

4.8.7 Temporal Domain Projections

To preserve clarity, the above description of the SQL implementation of the prototype system does not contain the temporal domain projection aspects of the proposed model. However, these aspects have been implemented as follows. First, clients calculate and submit with each status report the day-of-week (`dow`) and hour-of-week (`hod`) projections of the timestamp of the status report. These temporal domain projected values are stored in- or are propagated throughout the computations to each of the five tables of the relational database schema, i.e., each table has `dow` and `hod` as `int`-type columns that in the case of the `HDFS` and `HDFS` tables are also indexed. All temporal domain projected, long-term, HDFS and HDMS are stored in `HDFS` and `HDMS`, respectively. The value of `-1` for `dow` and `hod` are used to denote the "any" value for the domain projections, in general, and is used to distinguish between `dow`-projected-, `hod`-projected-, and global statistics. While the current directional flow and mobility statistics queries (SQL 1 and SQL 2 respectively) are modified to additionally return the current values of `dow` and `hod` from `RT`, the queries for maintaining historical summary statistics (SQL 3 and the analogous query for HDMS) are extended to `UPDATE` and `INSERT` statistics for the current values of `dow` and `hod`. The directional conges-

tion detection- (SQL 4) and notification queries (SQL 5) are modified to contain additional conditions so that they relate temporally domain projected historical information that match the current values of `dow` and `hod`. Specifically, the directional congestion detection and notification queries combine the different temporally domain projected information as a disjunction (logical OR) in their respective decision criteria. That is, a directional congestion is detected if the statistical power criterion and the speed difference criterion are satisfied *either* based on the `dow`-projected-, `hod`-projected- *or* the global statistics. Similarly, a notification is issued if, with a high likelihood, a vehicle is expected to encounter a detected directional congestion *either* based on the `dow`-projected-, `hod`-projected- *or* the global conditional probabilities of directional grid cells.

4.9 Generality of the Model and the Method

The proposed grid-based model is without limitations. In fact, the proposed methodology can be directly applied to a geographical road network model [14] by replacing the grid cell IDs with road network segment IDs and replacing the direction for entering a given grid cell from a given neighbor with the road network segment ID that precedes the given road network segment in a given continuous road network based trajectory [11].

The proposed gapless / spatially contiguous trajectory representation is without limitation. However, the quality of the detected congestions and the accuracy of the notifications of an adapted model are expected to decreases with the size / duration of the gaps relative to the size / duration of the trajectories because the adapted model needs to be learned from significantly less amount of information.

The congestion model of the method is without limitation and can be replaced with alternative, preferably grid-based, perhaps more sophisticated and holistic, congestion models without much effort.

Finally, the presented ITS application is only one of the possible applications of the dissemination system. For example, the proposed methodology can be used in Location Based Advertising (LBA) platforms to selectively send relevant offers to users not only based on their current location but their predicted near-future locations.

5. EMPIRICAL EVALUATIONS

5.1 Test Environment

The empirical evaluations have been carried out on a personal laptop with Intel® Core™ i7-5600U CPU with 16 GB of main memory and a 512 GB solid state drive running a 64-bit Ubuntu 14.04 LTS installation with PostgreSQL 9.3.9.

5.2 Real-world Data Set

The proposed method is evaluated on a six day long (Mon, Tue, Thu, Fri, Sat, Sun) sample of the near real-time stream of raw GPS positions of around 11,000 taxis moving on the streets of Wuhan, China [21]. In this sample, positions of moving vehicles are read approximately every 20 to 60 seconds, totaling about 85 million records. The time-stamped readings include vehicle ID, location, speed and heading. After removing obvious outliers, sampling gaps longer than 120 seconds are used to identify trips in individual trajectories. To adapt the raw GPS data set to the proposed framework, two consecutive Cartesian coordinate locations within a trip are linearly interpolated by approximating the interpolating line with a sequence of contiguous grid cells and corresponding speeds that are calculated by a modified Bresenham line algorithm [7]. After eliminating short trajectories (less than 300 seconds or 10 grid cells), approximately 2.26 million trips have been identified that are within an 18km-by-18km rectangular boundary that is centered at the mean coordinates of the measurements which approximates the city center. The identified trips have an average length of 1265 seconds and 82 grid cells and refer to 24783 100-meter grid cells. The resulting data set contains approximately 185 million 100-meter grid based status reports. A heat map of the trips is shown in Figure 2(a). The average length of trips in grid cells or the number status reports in the data set and the number of grid cells that are referenced therein increase approximately linearly with the inverse of *glen*, which is also termed as the *geographical level of detail* or *spatio-temporal resolution* of the model[4].

5.3 Experiment Setup

The empirical evaluations are divided in two large groups of experiments: accuracy assessment (Section 5.5) and scalability assessment (Section 5.6) experiments. To be able to evaluate the accuracy and scalability of the proposed method on a large, spatio-temporally dense data set, depending on the given experiment, trip trajectories are *temporally* aligned so that they occupy the same relevant spatio-temporal region and yet are reasonably representative for the given experiment scenario. In particular, for accuracy assessment experiments the six days worth of trajectory data is temporally aligned so that the trajectories take place on the same "fictional" day at the time that is indicated by their original timestamp. This data alignment is referred to as the hod-alignment. For scalability assessment experiments trajectories are temporally aligned to start at the same time instance of the same "fictional" day. This data alignment is referred to as the fixed-alignment. To ensure the statistical significance of the results for both sets of experiments n-fold cross-validations are performed by randomly partitioning the trajectories into n equal subsets. In the case of the accuracy experiments, $n-1$ of the subsets are used as training set and the remaining (hold-out) subset is used as test set, the experiments are run for each of the n possible hold-out sets, and the results of the experiments are averaged. In the case of the scalability experiments, the experiments are run for each of the n subsets and the results of the experiments are averaged. The experiments evaluate the performance of four different notification systems: (1) a system using hod-projected HDMS ($DMSC_{hod}$), (2) a system using global HDMS ($DMSC_{global}$), (3) a system using *non-directional*, hod-projected HDMS ($NDMSC_{hod}$), and (4) a system using the LMC for notifications (LMC). Unless otherwise stated in a given experiment, the default parameter values of the models are as follows: temporal analysis window size / prediction horizon $\Delta t_{awin} = \Delta t_{pred} = 60$ seconds, minimum number of current status reports $min\_veh = 2$, maximum sample dispersion $max\_cv = 0.5$, maximum negative z-score $max\_z = -1.65$ (which for a left-sided z-test represents a significance level of $\alpha = 0.05$), maximum negative relative speed difference $max\_relspddiff = -0.5$, and minimum notification probability threshold $min\_prob = 0.06$.

[4]See Section 5.6 for an explanation for this linear behavior.

5.4 Evaluation Framework

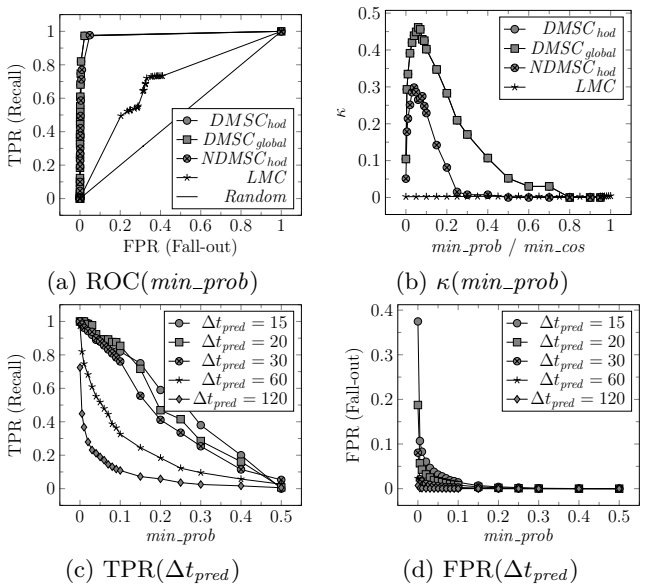
The following paragraphs describe and motivate the choices of evaluation framework that is used in the assessment of the proposed model and its prototype implementation.

The premise of the proposed methodology and the evaluations is that the selective directional congestion notifications need to be computed and evaluated for realistic / representative directional congestions. In lack of ground truth information on congestions, based on the qualitative analysis in [13] which has shown that the spatio-temporal distribution and clustering of the detected congestions are reasonable given the common notions about where congestions present themselves and how they evolve in space-time, the evaluations in this work treat the detected congestions as ground truth and are hence not subject of assessment.

Given the semantics of the congestions detections and notifications within the proposed framework, the binary classification assessment framework [9] is adapted for the assessment of the congestion notification quality as follows. First, while theoretically *any* moving object can be notified of *any* detected congestion, because it is far less likely that an arbitrary congestion will affect an arbitrary moving object than it will not, the baseline for notifications only considers possible notification cases when the moving object at the prediction time is within the maximal notification range max_r of a congestion from which the congestion can be theoretically reached assuming a linear movement model and a given maximum speed, which is set to 50 m/sec in the experiments. Consequently, while true positive (TN)-, false positive (FP)-, and false negative (FN) cases of notifications are calculated based on the notifications that are sent out (or complementary are not sent out) and the movement observations, i.e., whether an object's grid based trajectory does or does not include the directional grid cell ID of a given congestion within the prediction horizon, the true negative (TN) congestion cases are calculated as the complement of the other three cases w.r.t. the baseline for notifications (B), i.e., $TN = B - TP - FP - FN$.

Second, a congestion (g, dir) that is detected based on the reports of a temporal analysis window at time t_i is different from the congestion (g, dir) that is detected based on the reports of another (possibly consecutive) temporal analysis window at time t_j. This choice of treatment is motivated by the fact that while the two congestion notifications may refer to the same traffic phenomena, they have a different temporal observation and prediction validity. In particular, while the congestion (g, dir) that is detected and is sent out to an object o at time t_i can be a false positive notification, i.e., a notification that is incorrectly issued because o's trajectory does not include the directional grid cell ID of the congestion (g, dir) within the prediction horizon $(t_i, t_i + t_w size]$, the same congestion (g, dir) that is detected and sent out to o at time $t_i + t_w size$ can be a true positive notification.

Given the above described adaption of the binary classification assessment framework, the accuracy of congestion notifications of a system is evaluated as follows. First, to find a suitable trade-off between the sensitivity (i.e., the proportion of positive cases / notifications that are correctly identified / issued) and the specificity (i.e., the proportion of negative cases / non-notifications that are correctly identified / not issued) of a notification system, the Receiver Operator Characteristic (ROC) of the notification systems are explored by plotting the *True Positive Rate*, $TPR = TP/(TP + FN)$ (i.e., sensitivity) against the *False Positive Rate*, $FPR = FP/(FP + TN)$ (i.e., 1 - specificity) of the systems for varying decision threshold values (i.e., min_prob and min_cos) [9]. Second, as a statistically more robust measure, for each system's classifications under a given decision threshold value the Cohen's kappa coefficient [8] is calculated to account for any classification agreement that one can expect to arise by chance because of the highly unbalanced class priors. Finally, the AUC (Area Under the [ROC] Curve) metric, which represents the probability that a classifier assigns a higher positive-class probability to a randomly chosen positive case than to a randomly chosen negative case, is used as a single metric to evaluate overall performance of a classifier in the present case of highly unbalanced classes [9].



(a) ROC(min_prob) (b) $\kappa(min\_prob)$

(c) TPR(Δt_{pred}) (d) FPR(Δt_{pred})

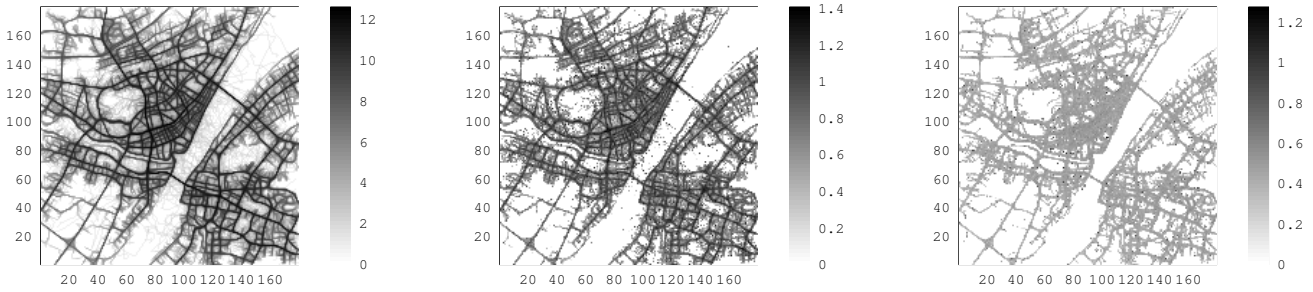
Figure 1: ROC (Fig. 1(a)) and Cohen's kappa coefficient (Fig. 1(b)) for varying min_prob / min_cos values for four notification systems. TPR (Fig. 1(c)) and FPR (Fig. 1(d)) for varying min_prob values for five $DMSC_{global}$-systems with different prediction horizon and temporal analysis window.

The scalability of the proposed notification systems is measured in terms of the time and the storage (i.e., number of rows in tables) that the computation phases use.

5.5 Accuracy Assessments

5.5.1 Sensitivity to Notification Criteria Thresholds

The results of the accuracy assessment of the four notification systems for varying notification criteria thresholds using the hod-alignment data set are presented in Figures 1(a) and 1(b). Although the thresholds of each system can be tuned to find an optimal value for TPR and FPR, it is clear that all of the HDMS based models, regardless of temporal domain projection or directional encoding, significantly outperform LMC, which inherently cannot capture the topology of the underlying road network. In particular, while the AUC-value of the HDMS-based models range from 0.9799 to 0.9831 (a nearly perfect classification), the AUC value of LMC is 0.6907 which is not sig-



(a) $\log(\#reports)$ (b) $CV_{dir\_hod}(g)$ (c) $CV_{dircond\_hod}(g)$

Figure 2: Spatial distribution of status reports and directional and temporal variability of HDMS.

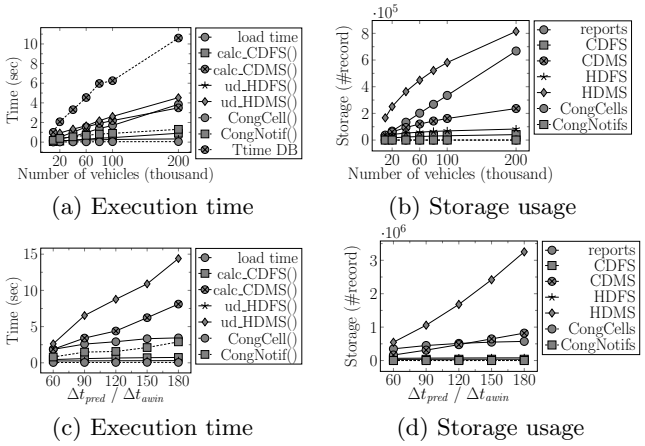
nificantly better than a random classifier, i.e., $\kappa \approx 0$ for all values of *min_cos* (Figure 1(b)). *LMC* is also unacceptable from an operational perspective because while it achieves a maximum TPR of 0.7334 and sends out 105 correct notifications during the processing an average temporal analysis window when *min_cos* = 0, in the given application setting, the corresponding FPR of 0.4099 represents approximately 41 thousand false notifications that are sent out to approximately 55 thousand vehicles about approximately 20 congestions, which is highly undesirable for the users of the system. In comparison the best HDFS based system ($DMSC_{hod}$) achieves a maximum TPR of 0.9731 and sends out 137 correct notifications FPR when *min_prob* = 0 at a corresponding FPR of 0.0230 representing approximately 2300 false notification. Although the classification performance differences between the different HDMS-based models are minimal the relative ranking, in part, is as expected: AUC($DMSC_{hod}$) = AUC($DMSC_{global}$) = 0.9831 > AUC($NDMSC_{hod}$) = 0.9799. The reason for the equal prediction performance of the first two system is further investigated in Section 5.5.4.

5.5.2 Sensitivity to Prediction Horizon Length

The results of the accuracy assessment of the most competitive $DMSC_{global}$-system for varying prediction horizon using the hod-alignment data set are presented in Figures 1(c) and 1(d). On can observe that TPRs and FPRs of a system increase more rapidly and for smaller values of *min_prob* as Δt_{pred} is increased. The figures also show the TPRs / FPRs decrease / increase for all values of *min_prob* as Δt_{pred} is increased. The reason for this behavior is that while the "shorter" mobility patterns for lower values of Δt_{pred} cover most of the positive cases they are not specific enough and incorrectly cover a lot of negative cases, while the "longer" mobility patters are spatially more specific.

5.5.3 Sensitivity to Spatio-temporal Resolution

The results of the accuracy assessment for varying geographical levels of detail / spatio-temporal resolutions (i.e., inverse of *glen*) has shown a nearly identical behavior to the behavior that is observed when one varies the prediction horizon length. Hence, due to space limitations, these results are not presented here in detail. This result is somewhat counterintuitive, but explainable by the fact that the vehicles move on a linear road network and their trajectories occupy only an approximately linearly increasing number of grid cells as the resolution is increased, which is effectively the same when the prediction horizon length is increased.



(a) Execution time (b) Storage usage

(c) Execution time (d) Storage usage

Figure 3: Execution time and space usage of different phases of the congestion detection and notification tasks in the $DMSC_{global}$ system for varying number of vehicles and values of prediction horizon / temporal analysis window size.

5.5.4 Variability of HDFS

As it is reported in Section 5.5.1, the prediction performance of $DMSC_{hod}$ and $DMSC_{global}$ are identical; indeed, the plots of $DMSC_{global}$ cover the plots of $DMSC_{hod}$ in Figures 1(a) and 1(b). To investigate the reason for this, the variability (i.e., Coefficient of Variation $CV = \sigma/\mu$) of the directional hod-projected mobility statistics $CV_{dir\_hod}(g)$ and the *directionally-conditioned* hod-projected mobility statistics $CV_{dircond\_hod}(g)$ are examined and shown in Figures 2(b) and 2(c), respectively. It can be seen that $CV_{dir\_hod}(g)$ is relatively very high on the main arteries of the road network, but $CV_{dircond\_hod}(g)$ is significantly lower and more even in the study area. This means that directional aspect of hod-projected HDMS captures most of the variability in movement and consequently the $DMSC_{global}$-system virtually provides the same predictions as the $DMSC_{hod}$-system, i.e., the short-term future position of an object is in large determined by its current position and movement direction.

5.6 Scalability Assessments

The results of the scalability assessment of the most competitive $DMSC_{global}$-system for varying input loads and prediction horizon lengths using the fixed-alignment data set are presented in Figure 3. Figures 3(a) and 3(b) show that

the implementation of all stages of the proposed method scale in execution time and storage usage (for processing an average temporal analysis window) at most linearly with the number of vehicles. Discounting the heavily dominating load time which can be likely be reduced using main.memory and stream based processing frameworks, provided the linear trends and the 60-minute real-time processing limit that is dictated by the size of the temporal analysis window, the proposed system can manage approximately $60/10.5 \times 0.2 \approx 1.14$ million vehicles. Figures 3(c) and 3(d) show that the system also scales approximately linearly with the length of the prediction horizon length. For the reasons provided in Section 5.5.3, the same linear behavior is observed when varying the spatio-temporal resolution.

6. CONCLUSIONS AND FUTURE WORK

This paper proposed a data-driven approach and a directional grid-based, time-inhomogeneous, Markov jump process model for the detection of and selective dissemination of traffic congestion information. Empirical evaluations on millions of object trajectories showed that (1) the selective dissemination accuracy of the proposed model is superior to a linear movement based model, (2) the proposed can effectively capture the topology of the underlying road network and the directional object movement on it, and (3) a simple, high-level SQL-implementation of the proposed model is scalable in terms of its input size and model parameters.

Future work plans include: (1) the performance evaluation of a road network based adaption of the proposed method and (2) the implementation of the model using main-memory and stream based Big Data processing frameworks.

Acknowledgements

Although the present research departs from Rui Zhu's master's thesis that the present author supervised [29], it was solely performed by the present author and represents a significant improvement in *all* aspects over its departure. The provision of trajectory preprocessing utilities is gratefully acknowledged to Christian Borgelt. Helpful comments on this work are also acknowledged to Adrian C. Prelipcean and Can Yang.

7. REFERENCES

[1] Apache Software Foundation. Apache Flink: An Open Source Platform for Scalable Batch and Stream Data Processing. Available from: `https://flink.apache.org/` [Accessed 15 July 2015].

[2] Apache Software Foundation. Apache Hadoop: Open-Source Software for Reliable, Scalable, Distributed Computing. Available from: `https://hadoop.apache.org/` [Accessed 15 July 2015].

[3] Apache Software Foundation. Apache Spark: A Fast and General Engine for Large-scale Data Processing. Available from: `https://spark.apache.org/` [Accessed 15 July 2015].

[4] A. Asahara, A. Sato, K. Maruyama, and K. Seto. Pedestrian-movement Prediction Based on Mixed Markov-chain Model. In *Proc. of ACM-GIS*, pp. 25–33, 2011.

[5] D. Ashbrook and T. Starner. Using GPS to Learn Significant Locations and Predict Movement Across Multiple Users. *Personal and Ubiquitous Computing*, 7(5):275–286, 2003.

[6] A. Bachmann, C. Borgelt, and G. Gidófalvi. Incremental Frequent Route Based Trajectory Prediction. *P*roc of. ACM SIGSPATIAL IWCTS, pp. 49–54, 2013.

[7] J.E. Bresenham. Algorithm for Computer Control of a Digital Plotter. *I*BM Systems Journal, 4(1):25–30, 1965.

[8] J. Cohen. A Coefficient of Agreement for Nominal Scales. *E*ducational and Psychological Measurement, 20(1):37–46, 1960.

[9] T. Fawcett. An Introduction to ROC Analysis. *P*attern Recognition Letters, 27(8):861–874, 2006.

[10] F. Giannotti, M. Nanni, F. Pinelli, and D. Pedreschi. Trajectory Pattern Mining. *Proc. of SIGKDD*, pp. 330–339, 2007.

[11] G. Gidófalvi, M. Kaul, C. Borgelt, and T.B. Pedersen. Frequent Route Based Continuous Moving Object Location- and Density Prediction on Road Networks. *P*roc. of ACM-GIS, pp. 381–384, 2011.

[12] G. Gidófalvi and F. Dong. When And Where Next: Individual Mobility Prediction. *P*roc. of MobiGIS, pp. 57–64, 2012.

[13] G. Gidófalvi and C. Yang. Scalable Detection of Traffic Congestion from Massive Floating Car Data Streams. *F*orthcoming in UrbanGIS, 2015.

[14] C. Jensen, T.B. Pedersen, L. Speicys, and I. Timko. Data Modeling for Mobile Services in the Real World. *P*roc. of SSTD, pp. 1–9, 2003.

[15] H. Jeung, Q. Liu, H. T. Shen, and X. Zhou. A Hybrid Prediction Model for Moving Objects. *Proc. of ICDE*, pp. 70–79, 2008.

[16] H. Jeung, M. Yiu, X. Zhou, and C. Jensen. Path Prediction and Predictive Range Querying in Road Network Databases. *Proc. of VLDB*, pp. 585–602, 2010.

[17] M. Kulldorff. Tests of Spatial Randomness Adjusted for an Inhomogeneity: A General Framework. *J*ournal of the American Statistical Association, 101(475):1289–1305, 2006.

[18] H. Kriegel, M. Renz, M. Schubert, and A. Zuefle. Statistical Density Prediction in Traffic Networks. *Proc. of SDM*, pp. 692–703, 2008.

[19] J. Krumm. A Markov Model for Driver Turn Prediction. In *Proc. of SAE World Congress*, 2008.

[20] J. Krumm, R. Gruena, and D. Delling. From Destination Prediction to Route Prediction. *JLBS*, 7(2):98–120, 2013.

[21] Li, Q., Zhang, T., and Y. Yu, Using Cloud Computing to Process Intensive Floating Car Data for Urban Traffic Surveillance. *I*JGIS, 25(8):1303–1322, 2011.

[22] Mantel, N. The Detection of Disease Clustering and a Generalized Regression Approach. *C*ancer Research, 27:209–220, 1967.

[23] A. Monreale, F. Pinelli, R. Trasarti, and F. Giannotti. WhereNext: A Location Predictor on Trajectory Pattern Mining. *Proc. of KDD*, pp. 637–645, 2009.

[24] Song, C., Qu, Z., Blumm, N., and Barabási, A.-L., Limits of Predictability in Human Mobility. *S*cience, 327:1018–1021, 2010.

[25] F. Verhein and S. Chawla. Mining Spatio-temporal Association Rules, Sources, Sinks, Stationary Regions and Thoroughfares in Object Mobility Databases. *Proc. of DASFAA*, pp. 187–201, 2006.

[26] Wikipedia, *S*tandard Deviation: Population-based Statistics Available from: https://en.wikipedia.org/wiki/Standard_deviation [Accessed July 15, 2015].

[27] Y. Ye, Y. Zheng, Y. Chen, J. Feng and X. Xie. Mining Individual Life Patterns Based on Location History. *Proc. of MDM*, pp. 1–10, 2009.

[28] J.J.-C. Ying, W.-C. Lee, T.-C. Weng and V.S. Tseng. Semantic Trajectory Mining for Location Prediction. *Proc. of ACM-GIS*, pp. 34–43, 2011.

[29] Zhu, R. Moving Object Trajectory Based Intelligent Traffic Information Hub. *M*asterâĂŹs of Science Thesis in Geoinformatics TRITA-GIT EX 13-011, Royal Institute of Technology (KTH), Stockholm, Sweden, pp. 1–51, 2013.