Machine-learning-augmented analysis of textual data: application in transit disruption management

Peyman Noursalehi, Haris N. Koutsopoulos, Jinhua Zhao

Despite rapid advances in automated text processing, many related tasks in transit and other transportation agencies are still performed manually. For example, incident management reports are often manually processed and subsequently stored in a standardized format for later use. The information contained in such reports can be valuable for many reasons: identification of issues with response actions, underlying causes of each incident, impacts on the system, etc. In this paper, we develop a comprehensive, pragmatic automated framework for analyzing rail incident reports to support a wide range of applications and functions, depending on the constraints of the available data. The objectives are twofold: a) extract information that is required in the standard report forms (automation), and b) extract other useful content and insights from the unstructured text in the original report that would have otherwise been lost/ignored (knowledge discovery). The approach is demonstrated through a case study involving analysis of 23,728 records of general incidents in the London Underground (LU). The results show that it is possible to automatically extract delays, impacts on trains, mitigating strategies, underlying incident causes, and insights related to the potential actions and causes, as well as accurate classification of incidents into predefined categories.

Index Terms—**Incidents, Information extraction, Natural Language Processing, Deep Learning, BERT**

I. INTRODUCTION

MANY transit agency functions, such as incident or customer feedback analysis, still rely on manual processing of information that is often embedded in unstructured text. Incidents and unexpected events, such as train malfunctions, station closures due to overcrowding, and signaling problems, are mainly reported by staff members and operators in a variety of forms, including handwritten documents and emails. Textual data collected in this fashion is unstructured, often filled with the jargon used in a transit agency and might not be easily accessible to someone lacking such knowledge. These reports are then manually processed, summarized and stored in a database of past events, usually in the format of large spreadsheets. These spreadsheets include predefined fields that record the main attributes of the incidents. After the template is filled, the raw text is probably never used again despite the fact that it includes, as we argue in the paper, valuable information that can help the agency identify trends, faults in procedures, and other information (as outlined in section 2, Table 1). Current, manual approaches are time-consuming and error-prone, and present a major obstacle when it comes to identifying and testing new hypotheses about possible event causes, extracting new attributes, or quickly evaluating previous decisions and summarizing the lessons learned. Most importantly, useful information that is embedded in the incident reports is often lost. For example, while the textual data contains detail information about the cause of incidents or the mitigating strategies, the reporting form does not include a separate field for recording such info. As a result, the final database does not contain such an information and it cannot be used for further analysis.

P. Noursalehi is with the Department of Urban Studies and Planning, Massachusetts Institute of Technology, Cambridge, MA, 02139 USA (e-mail: peymano@mit.edu)

H.N.Koutsopoulos is with the Department of Civil and Environmental Engineering, Northeastern University, Boston, MA 02115, 02115

J. Zhao is with the Department of Urban Studies and Planning, Massachusetts Institute of Technology, Cambridge, MA, 02139 USA

In this paper, we present a comprehensive methodology for automated analysis of such reports. It aims at processing massive historical textual databases where information was not extracted beforehand. We show that it is possible to automatically extract delays, impacts on trains, mitigating strategies, underlying incident causes, and insights related to the potential actions that could have prevented the incidents, as well as accurate classification of incidents into predefined categories. As such, the main focus of the paper is to a) test the hypothesis that archived unstructured text on incidents contains valuable information that is often lost and can be used to shape policies and procedures; and b) that automated methods can be developed to extract this useful information. In recent years, natural language processing and text mining techniques have experienced substantial growth and are currently used to automate knowledge discovery and information extraction from natural-language documents in various fields. There has been only a handful of attempts at automatic information extraction from transportation related incident reports. (1) used Latent Dirichlet Allocation (LDA) on railroad equipment accidents reported by the Federal Railroad Administration, to identify the recurring themes in major railroad accidents. (2) used LDA to uncover major themes in rail accidents from 2001 to 2011. (3) propose a bilevel feature extraction method for classification of fault classes. (4) proposed a methodology for predicting clearance time, the period between incident reporting and road clearance, of traffic incidents in real time, by incorporating information extracted from incident reports. They categorized 10,000 traffic incident records into topics using LDA which were subsequently used as explanatory variables to predict clearance time. Similarly, (5) classify police logs and incorporate this information into models to predict the impact of a given traffic incident. (6) analyzed 17,163 articles published in 22 transportation journals, identifying research trends over time using topic modeling. In a related work, (7) analyzed papers from the Transportation Research Board annual meetings using LDA, to identify research trends.

The main approaches used in the above research are based on statistical methods, such as LDA. The techniques for these tasks leverage expert knowledge and linguistic properties of the text. For example, (8) developed hand-coded rules to automatically analyze construction injury reports, and extract injury type and the body parts involved in each incident. Such hand-coded rules, which are often difficult and time consuming to develop, are an effective way of capturing application-specific knowledge. (9) provide a comprehensive review of text mining methods and their applications, and (10) give an overview of text mining techniques for decision support applications.

Our paper's contributions include the following:

1) Developing a comprehensive, pragmatic automated framework for analyzing rail incident reports to support a wide range of applications and functions. It includes a comprehensive suite of methods that work for the problem, each chosen depending on the constraints of the available data. For example, when large, labelled data was available (e.g., fault classification), we used the state-of-the-art deep learning methods and compared the accuracy with other methods. In the absence of such data (i.e., for extracting insights), we use rule-based approaches.
2) Leveraging pre-trained word embeddings for NER and document classification tasks
3) Leveraging the dependency graph to extract useful information. To the best of our knowledge, other related papers have not used this methodology.
4) Applying the methods to a realistic case study, including a comprehensive analysis of the implications of the extracted information, and demonstrate the practical use and value of the text mining methods for transit agencies.

The paper is organized as follows. Section II describes the rule-based and machine learning techniques that are proposed for information extraction related to transit incidents. Section III presents a case study, using incident reports from Transport for London (TfL). It demonstrates how the methods discussed in section II are used for automated extraction of delays, insights on how incidents could have been handled more effectively, impacts of incidents on trains, implemented mitigating strategies, and categorization of event causes. Section IV presents examples of further analysis facilitated by the initial information extraction tasks. Section V concludes the paper.

II. Methods and Procedures

Figure 1 summarizes the typical information that is of interest to agencies in dealing with incidents and disruptions. Note that these categories do not aim to provide comprehensive information about incidents. For example, in the United Kingdom, the Delay Attribution Guide provides an extensive summary of the known significant causes of delays. Here, we focus on a broader categorization of the incidents instead. Generally, text mining methods fall under two categories: a) Data-driven b) Knowledge-driven. Data-driven approaches extract information from text using machine learning techniques, such as Support Vector Machines (SVM) (11), Neural Networks (12), or Latent Dirichlet Allocation (13). A drawback of these methods is the requirement of having large, often manually annotated datasets in order to extract useful information. Knowledge-based methods use pre-defined lexico-syntactic patterns that are extracted by experts and encode domain knowledge. These patterns, often defined using regular expressions, are useful when annotated data is not available, although their development is time-consuming and requires domain knowledge.

In this paper, we use both, knowledge-based and machine learning techniques. Figure 2 summarizes the information that can be extracted from each incident's report along with the methods used for each task.

A fundamental step for further analysis of text data is recognizing the important entities, for example, the location where the incident occurred. Named entity recognition (NER) aims to automatically identify named entities and classifying them into predefined categories, such as train id, station name, transit line, or any other classes of interest. Traditional NER methods rely on hand-coded features and hard-coded rules. However, their use is limited and costly as they need to be constantly manually updated with new rules, and more importantly cannot capture the contextual information in a sentence, which often results in high error rates. Recently, deep learning-based architectures that use no hard-coded rules have witnessed significant success for this task (14; 15).

Integral to their success has been learning dense, contextual representation of tokens. These token embeddings are often learned from massive corpora of unlabeled data and have dramatically increased the performance of other downstream tasks (for example, Google reported improved results for an estimated 10 percent of search queries). In the sections II-A and II-B, we briefly describe pre-trained word embeddings and the state-of-the-art architecture for NER task.

The answers to some of the general questions posed in Figure 1 are inherently qualitative. For example, the answer to an inquiry about the underlying cause of an incident cannot be summarized in a numerical value, nor do we know a priori the set of possible answers. It consists of natural language elements that were originally used in the report to describe the incident. This is where methods that exploit the statistical characteristics of the text should be used. Topic modeling techniques are one category of models that have been developed for this task (section II-C).

A. Word embedding

Word embeddings are dense vector representation of words, and can capture both the semantic and syntactic information. Pre-trained embeddings trained from a large unlabeled corpus in an unsupervised manner have been shown to improve many downstream tasks (16). The earliest models, word2vec (17) and GloVe (18) learned static embeddings of each word. Recently, Transformer-based models (19) have been introduced to learn representations that are dependent on the particular context of occurrence in a sentence. One such model is BERT (Bidirectional Encoder Representations from

Transformers) (20), which has obtained state-of-the-art results in many applications. There are two strategies to applying BERT to downstream tasks. One is the feature-based approach, in which the pre-trained representations are used as input features to other machine learning models. In this approach, the term "embedding" refers to the output vector of the final Transformer layer (21). The other approach is the fine-tuning-based approach, which trains the downstream models by fine-tuning pre-trained parameters. In this paper, we adopt the former approach, in part because it is less computationally demanding.

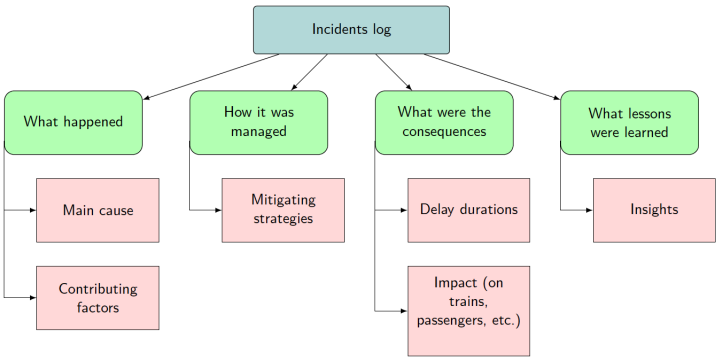


Fig. 1: Typical questions for analysis of incidents

B. Bi-LSTM CRF for NER

Figure 3 gives an example of the Bi-LSTM CRF model. Each sentence is represented as a sequence of tokens, typically labeled with the BIO (beginning, inside, outside) scheme. For example, "Oxford Station" is tagged as "B-station" and "I-station". Let $\mathbf{x} = \{x_1, x_2, \ldots, x_n\}$ and $\mathbf{y} = \{y_1, y_2, \ldots, y_n\}$ denote the input token sequence and their tags, respectively, where each token $x_i \in \mathbb{R}^d$ is represented by a d-dimensional vector. A bidirectional Long Short-Term Memory (Bi-LSTM) (22) then computes two hidden representations $h_t \in \mathbb{R}^H$ and $h'_t \in \mathbb{R}^H$ of the sentence, capturing the left and right context at each word. The final representation is obtained by concatenating the two, $\hat{h_t} = [h_t; h'_t]$, which now effectively possesses a representation of a word in context. Then a linear layer on top of the Bi-LSTM is used to predict the score of each tag for each word $e_t = tanh(W\hat{h_t})$. A simple tagging model could use the computed scores to make predictions on the labels of each word, by directly applying a softmax function for example. Such model ignores the dependence between consecutive tags which is essential in NER tasks. For example, "I-Station" cannot follow a "B-train" tag, but the independence assumption of the previous approach does not prevent such situations. Therefore, a Conditional Random Field (CRF) is used to capture such dependencies. Let $\mathbf{P} \in \mathbb{R}^{n \times k}$ be the scores matrix computed previously, where k is the number of tags. To capture the dependence among tags, define the tagging transition matrix $\mathbf{T} \in \mathbb{R}^{k+2 \times k+2}$, where $\mathbf{T_{i,j}}$ represents the score of transitioning from tag i to tag j. Typically, two additional tags are added to each sentence to denote the start and end of sequence, hence $\mathbf{T}$ is a square matrix of size $k+2$. Following (15), the score of each sentence is defined as

$$s(\mathbf{x}, \mathbf{y}) = \sum_{i=0}^{n} \mathbf{T}_{\mathbf{y_i}, \mathbf{y_j}} + \sum_{i=1}^{n} \mathbf{P}_{\mathbf{i}, \mathbf{y_i}}$$

The model is trained to maximize the log-probability of the correct tag sequence:

$$\log(p(\mathbf{y}|\mathbf{X})) = s(\mathbf{X}, \mathbf{y}) - \log\left(\sum_{\tilde{\mathbf{y}} \in \mathbf{Y_X}} e^{s(\mathbf{X}, \tilde{\mathbf{y}})}\right)$$

where $\mathbf{Y_X}$ are all possible tag sequences. For more details, the reader is referred to (15).

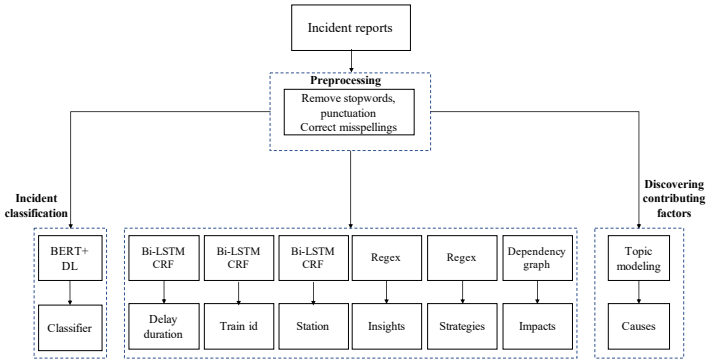


Fig. 2: Overview of the analysis process and methods

C. Topic modeling

Topic modeling, a type of unsupervised learning algorithm, deals with the problem of automatically organizing and understanding textual documents. Probabilistic Latent Semantic Indexing (pLSI) was one of the first attempts at probabilistic modeling of documents (23; 24), where each document is assumed to arise from a set of (latent) topics, and each topic itself is a mixture of words. (13) generalized this model by adopting a Bayesian approach, introducing a Dirichlet prior on topic distributions. The resulting model is known as Latent Dirichlet Allocation (LDA). It has the advantage of producing results that are usually easily interpretable.

Define document d as a vector of words $\boldsymbol{w_d} = \{w_{d,1}, \ldots, w_{d,n}\}$ where n is the number of distinct words in the document. We assume that the number of topics K is known. Each $k \in K$ has a distribution over the words

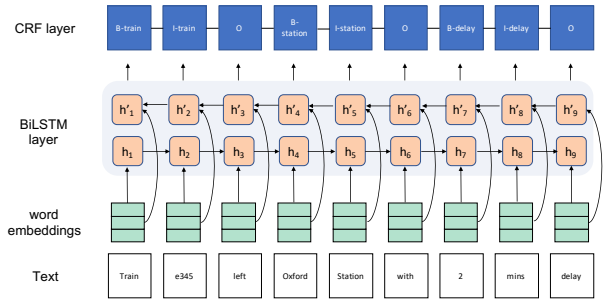


Fig. 3: NER using the Bi-LSTM CRF method

in the vocabulary, β_k, reflecting which words have a higher probability of occurrence under this topic. Since the words are discrete, β_k is modeled as a multinomial distribution, with a Dirichlet prior with parameter η. Each document d is generated using a distribution of the topics in K, specified with another multinomial distribution θ_d. The prior distribution for this per-document topic distribution is specified through another Dirichlet distribution with parameter α. If $\alpha < 1$, the model is biased towards sparsity, i.e. favoring models with just a few topics. Then the mains steps of the LDA algorithm are summarized as follows (25):

1) For each topic k, draw a distribution over words $\beta_k \sim Dirichlet(\eta)$
2) For each document d:
   a) Draw a vector of topic proportions $\theta_d \sim Dirichlet(\alpha)$
   b) For each word $w_{d,n}$:
      i) Draw a topic assignment $z_{d,n} \sim Multinomial(\theta_d), z_{d,n}\varepsilon\{1,\ldots,K\}$
      ii) Draw a word $w_{d,n}$ from $p(w_{d,n}|z_{d,n},\beta)$, a multinomial probability distribution conditioned on the topic z_n

The number of topics K, as well as hyperparameters η and α have to be set prior to fitting the LDA model. Several methods for performing Bayesian inference of an LDA model have been proposed in the literature, including variational Bayes (13; 26), Gibbs sampling (27) and expectation propagation (28). For an overview of these methods, the interested reader is referred to (29).

D. Document classification

A common task in text mining is to automatically label documents. The objective is, given a training dataset of (d, c) where d is a document and c is its label, to learn a mapping from d to c. Hence, in contrast to topic modeling, this is a supervised learning task. As an example, consider the problem of labeling the cause category of incidents (e.g. "Public and Customers", "Staff", "Fleet", etc.) based on their incident report. Applications include automatically labeling a database of old incident reports, and labeling a newly submitted report in real-time. Section III-F presents such an application.

III. Application

Transport for London (TfL) stores incident reports along with numerous categorized information associated with each one. For this case study, we use 23,728 incidents from January till mid-October 2016. This large database illustrates the need for efficient, automatic information extraction. Each incident is manually reported in an unstructured natural language text format and manually processed to populate standard predefined fields: service line, date, delay duration, and cause category. Fields are often not properly filled, even though the report contains relevant information. Incidents are then manually assigned to one of the predefined categories, such as "Customers and Public", "Staff", "Fleet", etc., and stored under the field "Cause Category". In this paper, we use the incident text description not only to automatically extract information to fill predefined fields, such as those mentioned above, but also to extract novel information that is not currently being explicitly captured. We focus on factors causing and contributing to a disruption, the actions that were taken to alleviate it, as well as the impact on the system and its major actors. As incident reports are usually filled out by the staff member at the scene, they sometimes include insights and recommendations that are not available through other means and are often lost. We also explore the potential of text mining methods to automatically extract such insights.

Fig. 4: Example of an annotated report

A. Preprocessing

Preprocessing techniques ensure a standardized text representation, which can significantly improve the performance of text mining tasks. We used TextBlob (30) package for spelling correction. Then, all the words were converted to lower case, and punctuations were removed. Further preprocessing is needed prior to the application of LDA, as described in section II-C. Stop words and numbers from the incident reports were removed and the text was tokenized into sentences. To reduce redundancy and gain more informative topics, words were stemmed prior to the analysis, i.e. different forms of the same word were consolidated into a single word. For example, "canceled", "cancel" and "cancellation" are all reduced to "cancel".

1) Annotating data

The NER model described in section II-B requires annotated data. We used the open-source software Doccano to manually annotate the following entities: train id, station, and delay duration. Figure 4 shows an example of an annotated (truncated) incident report. For the NER task, we post processed the annotations to be in the BIO format.

B. Impact on the passengers and trains

1) Delay duration

An important metric to understand the consequence of an incident is the subsequent delay. We selected 75 percent of the data for training and 25 for testing the model. We used BERT-base model for obtaining word embeddings, where each word is represented as a vector of size 768. The LSTM layers each have 32 hidden units. To assess the prediction accuracy, we used the precision, recall, and F1 score metrics, defined as:

$$\text{Precision} = P = \frac{\text{true positives}}{\text{true positives} + \text{false positives}}$$

$$\text{Recall} = R = \frac{\text{true positives}}{\text{true positives} + \text{false negatives}}$$

$$F_1 = \frac{2PR}{P+R}$$

TABLE I: Prediction accuracy for NER

	Bi-LSTM CRF		
Category	Precision	Recall	F1
Train id	0.92	0.95	0.94
Station	0.89	0.88	0.88
Delay	0.93	0.95	0.94

TABLE II: Discrepancies between recorded and extracted delay duration

Text	Manually Recorded	Automatically Extracted	Possible reason for discrepancy
"a delay of 16 minutes recorded"	6	16	Manual entry error
"3 min delay minus dwell time resulting in 2 mins."	2	3	Further numerical modifications
"2 mins delay. ... attribution is 0 mins"	0	2	later attribution
"An initial delay of 107 minutes resulted with the Jubilee service suspended in consequence"	4	107	Likely manual entry error

F_1 is the harmonic mean of the precision and recall. It assumes values between zero and one, and is the most frequently used metric in practice (31). Table I shows the prediction accuracy of the Bi-LSTM CRF model for each of the categories over the test set.

There are 149 instances in the test set where the recorded delay is zero or null, while the corresponding raw text in the report indicates otherwise. These represent cases where delay information was present in the original text but not previously recorded. There are 434 cases where the extracted and reported delays differ. Table II shows a few representative cases, along with the possible explanation for each one. Note that in the majority of cases the discrepancies are caused by incorrect manual reporting, which the proposed algorithm rectifies.

2) Impact on trains

Incident reports usually contain information about the trains that were affected, and actions that were subsequently taken to address the problem. For example, a report might contain the sentence " ...t402 was canceled due to... ". It expresses the impact ("canceled") on a specific train (t402). In these cases, simple regular expressions are not effective, because such relationships are often expressed in a long-distance form, with several phrases in-between the verb and subject that are non-essential (e.g. "t402 which was traveling from station A to B, was canceled"). For tasks of this kind, standard methods exploit the grammatical features of the sentences to capture such long-distance dependencies. In our case, we exploit the fact that the subject of the impact (i.e. "canceled") is a train. The following rule was used for detecting impacts on a train: if "train" is the passive nominal subject (nsubjpass) of a past tense verb (VBD), then that verb describes the impact, and the numerical modifier (nummod) is the train number. "nsubjpass", "nummod", and "VBD" are based on the Stanford typed dependencies representation (32), which tries to standardize grammatical representations of words. Figure 5 illustrates this through an example. Here, the extracted relation is ("402", "delayed"). The Stanford CoreNLP library (33), a set of natural language analysis tools, is used for generating the dependency tree, which represents grammatical relations between words in a sentence. Figure 6 summarizes the results of the analysis, illustrating the most frequent impacts. As expected, the most common impact is "delay". Often, trains are canceled, held, or withdrawn. Note that a few of the extracted

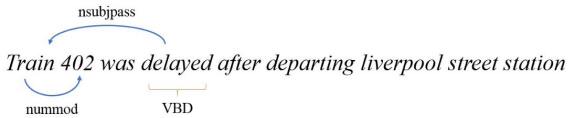


Fig. 5: Example of syntactic structure of a sentence which represents the impact of an incident on a train

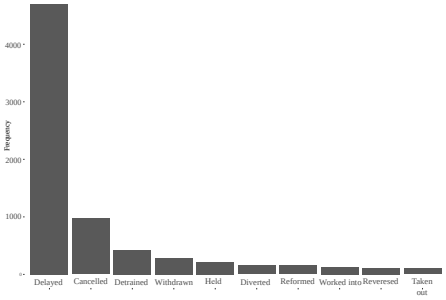


Fig. 6: Most frequent impact of incidents on trains

impacts overlap. For example, "withdrawn" and "taken out" both refer to the same action, but expressed differently.

C. Mitigating strategies

Operators deploy various strategies to deal with disruptions and mitigate their impacts. (34) present a comprehensive review of these strategies in the London Underground. They identified cancellation, short-turning, holding, renumbering, skipping stations (non-stopping), reversing and diverting direction, and withdrawing trains as the most common strategies. It is useful to extract from the reports the strategies that were used, as it can inform future approaches of how disruptions are handled. We used simple regular expressions to search reports for the aforementioned strategies. Table III shows the frequencies of each strategy, as extracted from the text. As expected, the most common strategy is cancellation and withdrawal from service. Renumbering is mentioned only in 44 incidents, despite the fact that it is a very common strategy at TfL. This suggests that such actions are typically not recorded. There might be two possible explanations for this. One is that renumbering happens so frequently that it is not deemed noteworthy to record them, or is implied from the context of the report. It also could be that it is applied in combination with other strategies.

TABLE III: Frequency of employed control strategies

Strategy	Frequency
Cancelled	4086
Withdrawn	1558
Diverted	992
Reversed	977
Non-stopped	479
Hold	346
Renumbered	44

TABLE IV: Regular expression for extracting insights from text

Regular expression
`(.*(?:\bif\b|\bhad\b)).*(?:(would|could) have been (?:prevented|minimized|minimised|avoided))`
`(.*(would|could) have been.* (but|if))`

TABLE V: Examples of extracted insights

"...also note, that this incident could have been significantly reduced if LU staff were allowed to access the track."
"The delay could have been avoided if staff reported this as soon as it was found."
"The delay could have been minimised if ... reporting the fault had informed the relieving driver."
"The initial problem of ... would have been rectified if the correct procedure had been followed."

D. Insights

Incidents are often reported by the staff members who dealt with them first. As such, their reports sometimes contain their opinions or judgments about the causes of the incidents, or the procedures which would have prevented them from happening. Such comments can provide invaluable insights for future disruption management decision making. We identified two common patterns in which these insights are expressed and encoded them as a regular expression (Table IV). They are based on the observation that insights are typically expressed through sentences such as "would have been avoided if ...", or "had ... could have been prevented". Overall, 27 sentences were extracted. All were classified as true insights after manual examination. Table V shows a few examples. The most common thread among the uncovered insights is the insufficient sharing of information among staff.

E. Contributing factors

Typically, incidents are organized under broad categories, such as "Customers and Public", "Signals", "Fleet", etc. These predefined categories, however, are restrictive and do not lend themselves to more detailed analysis. We cast discovering contributing factors as a topic modeling problem. We use LDA for extracting and understanding the events that cause disruptions in the system. As mentioned in section II-C, the number of topics, and values of the hyperparameters α and η are specified prior to analysis. In this study, we set the number of topics K=20. The number was chosen empirically, based on a) our understanding of the data b) interpretability of the resultant topics c) clear separation of significant words within each topic. We set $\alpha = 0.1$ to ensure a sparse topic distribution. η is set to 0.01. This relatively small value for hyperparameter η is expected to result in the discovery of fine-grained topics, i.e. each topic is a mixture of only a few words.

In this section, we focus on incidents categorized as "Customers & Public" and "Staff", which represent the two most frequent categories. There are 5002 and 6406 incidents under these two categories, respectively. The vocabulary size is 1607 words for "Customers & Public" and 1729 for "Staff" reports, occurring a total of 20611 and 27204 times, respectively. Table VI summarizes the inferred topics for each of the two categories. It is important to note that the lexical differences between reports do not necessarily correspond to independent topics. For example, for "Customers and Public", both topic one and three capture incidents due to activation of the passenger emergency alarm. However, in one set of documents, the acronym (pea) is used instead of the complete form. Hence, topics that are closely related can be merged into one. In Table VI, we report the results as inferred by the LDA method without any post processing to merge related topics. In practice, a simple step would be to re-categorize topics that are expressed differently in the text but refer to the same contributing factor, hence unifying the related factors. Note that this issue would not be resolved by reducing K, the number of topics, since these two are still significantly different in terms of important words.

TABLE VI: Extracted topics and their interpretation as contributing factors

Topic	"Customers and public"	"Staff"
1	PEA	Operator not available (ONA)
2	Trespassing	Rear cab door open
3	Passenger emergency alarm	Wrong Signal Lowered
4	Doors failing to close, loss of door closed visual (DCV)	Short-notice staff sickness
5	Soiled car, vomit	Becoming front tripped. Speed control
6	Passenger requiring staff assistance.	Train Operator not in position
7	Platform Train Interface incident (PTI)	Failed to gain a door closed visual
8	Customer action (e.g. stuck between doors)	Personal Needs Relief (PNR)
9	Operator contacted service control for assistance and advice	Train operator not available
10	Passenger illness	All spare operators were being utilized
11	Service gap (delay recorded)	Operator late for duty
12	Retrieve items from track, mainly cellphones	Signal passed at danger
13	Train delayed berthing into platform due to customer service supervisor request	Service Operator failing to clear signal
14	Vomit, broken window	No forward movement
15	General train movement (departure, arrival, etc.)	Traction current being switched off
16	Overcrowding	Awaiting train operator, PNR
17	Customer falling ill, collapsing, fainting	Staff shortage
18	Sensitive Door Edge (SDE) activation.	Staff error
19	Offensive graffiti in the car cabin	Insufficient cover
20	Customer altercation on train, refusing to leave	ONA was not forecasted

F. Automatic incident classification

We describe the application of supervised machine learning techniques to automatically assign incident reports to one of the predefined categories. These categories are manually assigned to each incident, classifying it under the appropriate broad category. We maintain the 5 most frequent categories, and group every other category under "Other". As shown in Table VII, there is a class imbalance as some categories are more frequent than the others. If not accounted for, the model would achieve high overall prediction accuracy even if it performs poorly on the infrequent categories. We therefore change our loss function to account for this class imbalance. Let $y_{ic} = \mathbb{I}(y_i = c)$ be the one-hot encoding of y_i. Then the weighted negative log likelihood loss function is defined as

$$L = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C} w_c y_{ic} \log \hat{p}(y_c|\mathbf{x}) \quad (1)$$

where the weight of class c, w_c, is the size of largest class divided by the size of class c, and $\hat{p}(y_c|\mathbf{x})$ is the predicted probability of class c. This loss function penalizes the model more for performing poorly on classes with lower observed

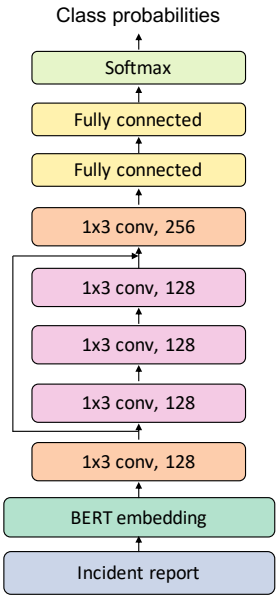


Fig. 7: Deep neural network architecture for text classification (resnet)

TABLE VII: Frequency of incidents, by category

Category and Public	Frequency
Customers and Public	5002
Fleet	5577
Signals	1505
Staff	6406
Station	3115
Other	2123

frequency. We use two approaches for converting the text data into numerical features: tf-idf weighting scheme, and pre-trained word embeddings trained on large corpora. For the latter, we use the BERT-base model weights (20). In this approach, the embedding weights are kept frozen and only the classifier model is trained. We hypothesis that this feature-based transfer learning can improve the prediction accuracy over the tf-idf approach, since it can leverage the latent contextual word representations learned from massive corpora. The deep learning model architecture (resnet) is shown in Figure 7. First, each sentence in an incident report is encoded using BERT. Then it passes through 5 layers of 1-dimensional convolutional layers with residual connections, progressively capturing and learning the local dependencies between words. Finally, the output is passed to two fully connected dense neural layers each with 128 units, and the final class probabilities are computed using the Softmax function. The network was implemented in PyTorch (35), and we used the HuggingFace library (36) to obtain pre-trained BERT models in PyTorch. We used logistic regression (lr) to benchmark the performance resnet model.

We randomly selected 75 percent of the data for training and 25 for testing the model. We further used three-fold cross validation for estimating the model from the training set. Table VIII summarizes the performance of the models. It

TABLE VIII: Classification accuracy

	BERT-resnet			BERT-lr			tf-idf resnet			tf-idf lr		
Category	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
All categories	0.86	0.87	0.87	0.79	0.80	0.79	0.79	0.78	0.78	0.73	0.71	0.71
Customers and Public	0.88	0.90	0.89	0.84	0.79	0.81	0.86	0.75	0.80	0.74	0.76	0.75
Fleet	0.87	0.91	0.89	0.76	0.86	0.81	0.80	0.80	0.80	0.76	0.71	0.73
Signals	0.79	0.74	0.76	0.68	0.68	0.68	0.60	0.78	0.68	0.45	0.59	0.51
Staff	0.89	0.90	0.89	0.81	0.83	0.82	0.84	0.78	0.81	0.81	0.70	0.75
Station infrastructure	0.89	0.95	0.92	0.86	0.94	0.90	0.87	0.93	0.90	0.90	0.82	0.85
Other	0.75	0.54	0.63	0.69	0.42	0.52	0.48	0.55	0.51	0.35	0.49	0.41

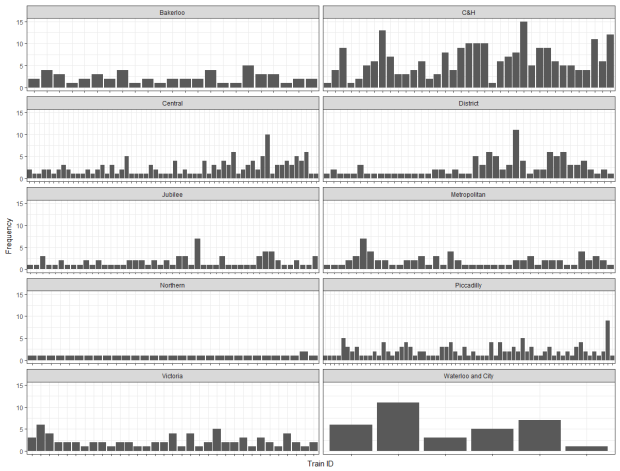
shows that using pre-trained BERT embeddings significantly improved the prediction accuracy for both the deep learning (resnet) and logistic regression models. Overall, the resnet model provides the most accurate predictions. All models perform relatively poorly on the incidents labeled as "Other". A possible explanation for this is that, since we grouped all the low-frequency categories under this label, the texts are highly heterogeneous compared to the other five categories.

IV. Further analysis

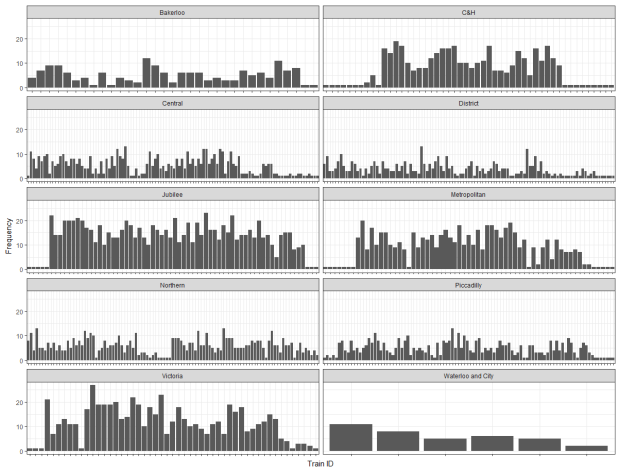
The previous sections described the information that can be extracted from raw incident reports. The new information can augment the incident database with additional structured fields. As such, it can facilitate the more detailed analysis of incidents. For example, since the methodology to extract the impact of incidents on trains also records the train number associated with the incident, we can analyze how specific trains were affected. Figures 8a and 8b show the frequency with which each train was canceled or delayed, grouped by the service line.

Although the impacts are uniformly distributed across trains, there are some train numbers that are associated with more than twice as many cancellations or delays (e.g. Circle and Hammersmith (C&H), Central, and District lines). This information can be useful for operators to further investigate trains individually in search of causes, including possible faulty train equipment, trends, and systematic patterns that may lead to corrective actions. The mitigating strategies that were used to manage incidents can be further analyzed based on the the reported cause of the disruption. The overwhelming majority belong to four categories: "Customers and Public", "Fleet", "Signals", and "Staff". Figure 9 shows a mosaic plot of cause categories and the common strategies. In this plot, the width of each column is proportional to the frequency of each strategy, and within each strategy, the height of each box is proportional to the frequency of the cause category. It reveals that most of the cancellations are due to "Fleet" and "Staff". Incidents due to signal issues result in trains mostly diverted or reversed. Trains are rarely non-stopped or withdrawn because of it. However, when incidents due to "Customers and Public" and "Staff" take place, trains usually skip stops.

We further inspect the frequency and prevalence of the extracted contributing factors (Section III-E) to uncover recurring issues for incidents classified under "Customers and Public" (Fig 10) for two of the lines. Line 1, which mostly serves areas on the periphery of the city, suffers frequent delays due to trespassing (topic 2). In contrast, line two is frequently facing delays due to doors not closing properly (topic 4), likely because of overcrowding (topic 16). A detailed analysis of these contributing factors (some of which could have been



(a) Frequency of cancellations



(b) Frequency of delays

Fig. 8: Frequency of each impact per train id

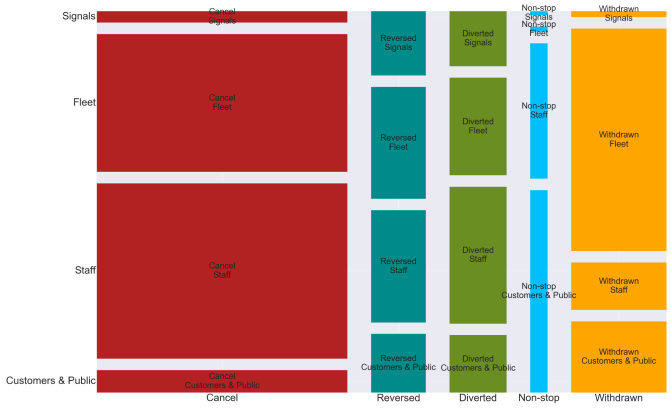


Fig. 9: Mosaic plot of the employed control strategies and their corresponding incident types

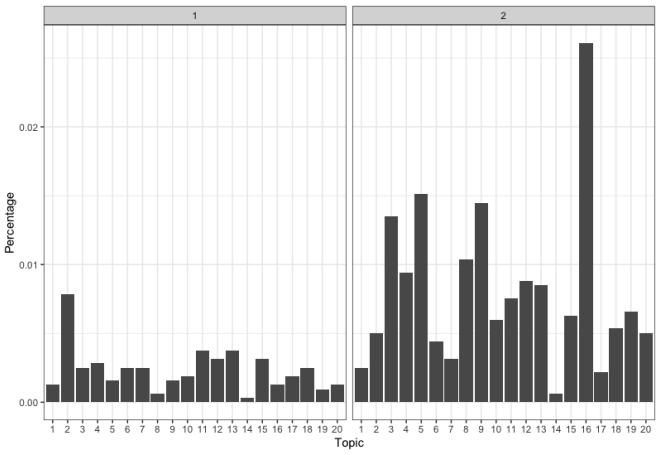


Fig. 10: Frequency of disruptions attributed to category ”Customers and Public”

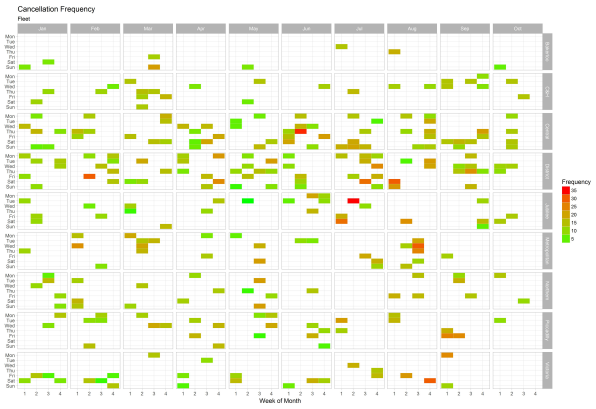


Fig. 11: Calender heatmap of frequency of canceled trips due to “fleet” problems

left unnoticed by being in the original textual format) can help decision makers to identify and rectify those, improving the reliability of the system.

Considering the frequency with which trains are canceled, we further break down these incidents by line, date, and cause category. Figure 11 shows the calendar heatmap of cancellations due to “Fleet”. It illustrates that District and Central lines are more frequently affected. In comparison with cancellations due to “Customers and Public” (Figure 12), these lines rarely experienced such incidents, while Jubilee and Northern lines were often affected.

V. Conclusion

In this paper, we present a methodology for automatic information extraction from unstructured text from incident reports, as applied to transit disruptions. We identify the underlying causes, impacts on the passengers and trains, implemented mitigating strategies, and recommendations for minimizing the effect of a disruption. We also show that reports can be automatically classified into different categories, possibly in real-time. The methodology, as developed in this paper, is also applicable to other functions of a transit agency, and processing of manually generated reports by transportation agencies

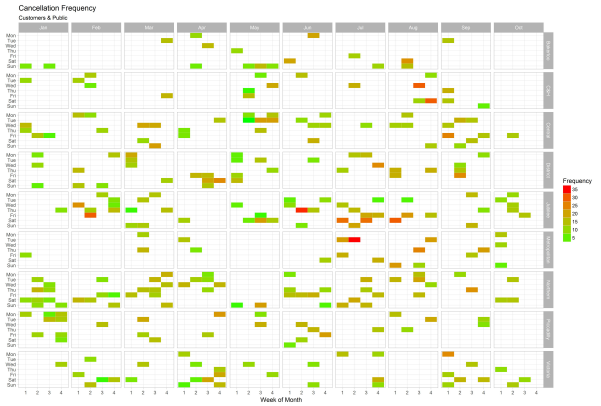


Fig. 12: Calender depiction of frequency of canceled trips due to "Customers and Public" problems

(e.g., accidents logs). Another interesting application could be extracting useful insights from passenger feedback. With the large number of complaints and suggestions that transit agencies receive from their customers, these automated text mining tools can help improve the efficiency with which they are processed. This can be accomplished by either extracting information directly from text, or by classifying a report and automatically sending it to the corresponding personnel, hence distributing the workload more efficiently.

ACKNOWLEDGMENTS

The authors would like to thank Transport for London for providing the data used in this study and many useful discussions and for the generous support provided for the research.

REFERENCES

[1] T. P. Williams, M. Asce, and J. F. Betak, "Identifying Themes in Railroad Equipment Accidents Using Text Mining and Text Visualization," in *International Conference on Transportation and Development*, 2016.

[2] D. E. Brown, "Text Mining the Contributors to Rail Accidents," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 2, pp. 346–355, 2016.

[3] F. Wang, T. Xu, T. Tang, M. Zhou, and H. Wang, "Bilevel feature extraction-based text mining for fault diagnosis of railway systems," *IEEE transactions on intelligent transportation systems*, vol. 18, no. 1, pp. 49–58, 2016.

[4] F. C. Pereira, F. Rodrigues, and M. Ben-Akiva, "Text analysis in incident duration prediction," *Transportation Research Part C: Emerging Technologies*, vol. 37, pp. 177–192, 2013.

[5] M. Miller and C. Gupta, "Mining traffic incidents to forecast impact," *Proceedings of the ACM SIGKDD International Workshop on Urban Computing - UrbComp '12*, p. 33, 2012.

[6] L. Sun and Y. Yin, "Discovering themes and trends in transportation research using topic modeling," *Transportation Research Part C: Emerging Technologies*, vol. 77, pp. 49–66, 2017.

[7] S. Das, X. Sun, and A. Dutta, "Text Mining and Topic Modeling of Compendiums of Papers from Transportation Research Board Annual Meetings," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2552, pp. 48–56, 2016.

[8] A. J. Tixier, M. R. Hallowell, B. Rajagopalan, and D. Bowman, "Automated content analysis for construction safety: A natural language processing system to extract precursors and outcomes from unstructured injury reports," *Automation in Construction*, vol. 62, pp. 45–56, 2016.

[9] C. C. Aggarwal and C. X. Zhai, *Mining text data*. Springer Science & Business Media, 2013, vol. 8.

[10] F. Hogenboom, F. Frasincar, U. Kaymak, F. De Jong, and E. Caron, "A Survey of event extraction methods from text for decision support systems," *Decision Support Systems*, vol. 85, pp. 12–22, 2016.

[11] T. Joachims, "Text Categorization with Suport Vector Machines: Learning with Many Relevant Features," in *Proceedings of the 10th European Conference on Machine Learning ECML '98*. Springer, 1998, pp. 137–142.

[12] Y. Goldberg, "Neural Network Methods for Natural Language Processing," *Synthesis Lectures on Human Language Technologies*, vol. 10, no. 1, pp. 1–309, 2017.

[13] D. Blei, A. Ng, and M. Jordan, "Latent Dirichlet Allocation," *Journal of Machine Learning Research*, vol. 3, no. 4-5, pp. 993–1022, 2003.

[14] Z. Huang, W. Xu, and K. Yu, "Bidirectional lstm-crf models for sequence tagging," *arXiv preprint arXiv:1508.01991*, 2015.

[15] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer, "Neural architectures for named entity recognition," *arXiv preprint arXiv:1603.01360*, 2016.

[16] S. Lai, K. Liu, S. He, and J. Zhao, "How to generate a good word embedding," *IEEE Intelligent Systems*, vol. 31, no. 6, pp. 5–14, 2016.

[17] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, 2013, pp. 3111–3119.

[18] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532–1543.

[19] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[20] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[21] A. Rogers, O. Kovaleva, and A. Rumshisky, "A primer in bertology: What we know about how bert works," *arXiv preprint arXiv:2002.12327*, 2020.

[22] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional lstm and other neural network architectures," *Neural networks*, vol. 18, no. 5-6, pp. 602–610, 2005.
[23] T. Hofmann, "Probabilistic latent semantic indexing," in *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 1999, pp. 50–57.
[24] ——, "Unsupervised Learning by Probabilistic Latent Semantic Analysis," *Machine learning*, vol. 42, no. 1-2, pp. 177–196, 2001.
[25] D. M. Blei and J. D. Lafferty, "Topic models," *Text Mining: Theory and Applications*, vol. 10, no. 1, pp. 71–93, 2009.
[26] W. Buntine, "Variational Extensions to EM and Multinomial," in *Machine Learning ECML 2002*, vol. 317, no. 5-6. Springer, 2002, pp. 23–34.
[27] T. L. Griffiths and M. Steyvers, "Finding scientific topics," *Proceedings of the National Academy of Sciences*, vol. 101, no. Supplement 1, pp. 5228–5235, 2004.
[28] T. Minka and J. Lafferty, "Expectation-Propagation for the Generative Aspect Model," in *Uncertainty in Artificial Intelligence*. Morgan Kaufmann Publishers Inc., 2002, pp. 352–359.
[29] S. Cohen, "Bayesian analysis in natural language processing," *Synthesis Lectures on Human Language Technologies*, vol. 9, no. 2, pp. 1–274, 2016.
[30] S. Loria, P. Keen, M. Honnibal, R. Yankovsky, D. Karesh, E. Dempsey *et al.*, "Textblob: simplified text processing," *Secondary TextBlob: Simplified Text Processing*, 2014.
[31] D. Jurafsky and J. H. Martin, "Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition," *Speech and Language Processing*, 2008.
[32] M.-c. D. Marneffe and C. D. Manning, "Stanford typed dependencies manual," *20090110 Httpnlp Stanford*, vol. 40, no. September, pp. 1–22, 2010.
[33] C. D. Manning, J. Bauer, J. Finkel, S. J. Bethard, M. Surdeanu, and D. McClosky, "The Stanford CoreNLP Natural Language Processing Toolkit," in *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, 2014, pp. 55–60.
[34] A. Carrel *et al.*, "Diagnosis and assessment of operations control interventions: Framework and applications to a high frequency metro line," Ph.D. dissertation, Massachusetts Institute of Technology, 2009.
[35] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," 2017.
[36] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, and J. Brew, "Huggingface's transformers: State-of-the-art natural language processing," *ArXiv*, vol. abs/1910.03771, 2019.

Peyman Noursalehi is a a Postdoctoral Associate at the MIT Transit and Urban Mobility Lab. He earned his PhD in Transportation Systems from Northeastern University, and his BSc in Civil Engineering from Sharif University of Technology.

Haris N. Koutsopoulos is a Professor in the Department of Civil and Environmental Engineering at Northeastern University in Boston, and research affiliate with the Transit Research Program at MIT. He also holds a position as Guest Professor of Transport Science at KTH, the Royal Institute of Technology in Stockholm, Sweden. He has extensive experience in modeling Intelligent Transportation Systems and developing simulation-based dynamic traffic assignment methods, traffic simulation models at various levels of resolution, simulation tools for the analysis of urban transit operations, and methods for processing and using data from opportunistic sensors for monitoring and control of transport system operations (traffic and transit). At KTH he established the iMobility laboratory. The lab receives real time floating car and other traffic related data (e.g. traffic counts and speeds, weather), archived transit AFC and AVL data, and uses models to support applications such as monitoring and management of transport systems and multimodal travel planning services. For the work related to the iMobility lab he received the IBM Smarter Planet Award in 2010.

Jinhua Zhao is the Edward H. and Joyce Linde Associate Professor of City and Transportation Planning at the Massachusetts Institute of Technology (MIT). Prof. Zhao brings behavioral science and transportation technology together to shape travel behavior, design mobility system and reform urban policies. He develops methods to sense, predict, nudge and regulate travel behavior, and designs multimodal mobility system that integrates autonomous vehicles, shared mobility and public transport. Prof. Zhao directs the Urban Mobility Lab (mobility.mit.edu) at MIT.