DxNAT - Deep Neural Networks for Explaining Non-Recurring Traffic Congestion

Fangzhou Sun, Abhishek Dubey, Jules White

Institute for Software Integrated Systems, Vanderbilt University

Nashville, TN, USA

{fangzhou.sun,abhishek.dubey,jules.white}@vanderbilt.edu

Abstract—Non-recurring traffic congestion is caused by temporary disruptions, such as accidents, sports games, adverse weather, etc. We use data related to real-time traffic speed, jam factors (a traffic congestion indicator), and events collected over a year from Nashville, TN to train a multi-layered deep neural network. The traffic dataset contains over 900 million data records. The network is thereafter used to classify the real-time data and identify anomalous operations. Compared with traditional approaches of using statistical or machine learning techniques, our model reaches an accuracy of 98.73 percent when identifying traffic congestion caused by football games. Our approach first encodes the traffic across a region as a scaled image. After that the image data from different timestamps is fused with event- and time-related data. Then a crossover operator is used as a data augmentation method to generate training datasets with more balanced classes. Finally, we use the receiver operating characteristic (ROC) analysis to tune the sensitivity of the classifier. We present the analysis of the training time and the inference time separately.

Keywords—public transportation; deep learning; neural network; anomaly detection; traffic congestion;

I. INTRODUCTION

Emerging Trends. Traffic congestion in urban areas has become a significant issue in recent years. Because of traffic congestion, people in the United States traveled an extra 6.9 billion hours and purchased an extra 3.1 billion gallons of fuel in 2014. The extra time and fuel cost were valued up to 160 billion dollars [1]. Congestion that is caused by accidents, roadwork, special events, or adverse weather is called non-recurring congestion (NRC) [2]. Compared with the recurring congestion that happens repeatedly at particular times in the day, weekday and peak hours, NRC makes people unprepared and has a significant impact on urban mobility. For example, in the US, NRC accounts for two-thirds of the overall traffic delay in urban areas with a population of over one million [3].

Driven by the concepts of the Internet of Things (IoT) and smart cities, various traffic sensors have been deployed in urban environments on a large scale. A number of techniques have been developed for knowledge discovery and data mining by integrating and utilizing the sensor data. Traffic data is widely available by using static sensors (e.g., loop detectors, radars, cameras, etc.) as well as mobile sensors (e.g., in-vehicle GPS and other crowdsensing techniques that use mobile phones). The fast development of sensor techniques enables the possibility of in-depth analysis of congestion and causes.

The problem of finding anomalous traffic patterns is called traffic anomaly detection. Understanding and analyzing traffic anomalies, especially congestion patterns, is critical to helping city planners make better decisions to optimize urban transportation systems and reduce congestion conditions. To identify faulty sensors, many data-driven and model-driven methods have been proposed to incorporate historical and real-time data [4], [5], [6], [7]. Some researchers [8], [9], [10], [11] have worked on detecting traffic events such as car accidents and congestion using videos, traffic, and vehicular ad hoc data. There are also researchers who have explored the root causes of anomalous traffic [12], [13], [14], [15], [16], [17].

Most existing work still mainly focuses on a road section or a small network region to identify traffic congestion, but few studies explore non-recurring congestion and its causes for a large urban area. Recently, deep learning techniques have gained great success in many research fields (including image processing, speech recognition, bioinformatics, etc.), and provide a great opportunity to potentially solve the NRC identification and classification problem. There are still many open problems: (1) using feature vectors to represent traffic conditions loses the spatial information of the road segments, (2) using small and unbalanced dataset (traffic data with event labels is limited) to train neural networks downgrades the performance, a proper data augmentation mechanism is needed to balance the training data with different class labels, (3) building deep neural networks to model the traffic conditions of both recurring and non-recurring congestion.

Contributions. In this paper, we propose DxNAT, a deep neural network model to identify non-recurring traffic congestion and explain its causes. To the best of our knowledge, our work is one of the first efforts to utilize deep learning techniques to study traffic congestion patterns and explain non-recurring congestion using events. The main contributions of our research are summarized as follows:

- We present an algorithm to efficiently convert traffic data in Traffic Message Channel (TMC) format to images
- We introduce a crossover operator as a data augmentation method for training class balancing.
- A convolutional neural network (CNN) is proposed to identify non-recurring traffic anomalies that are caused by events.
- We create three scenarios to evaluate the performance

of the proposed model by using real-world data of three events types (football games, hockey games, and traffic incidents).

Paper Organization. The remainder of this paper is organized as follows: Section II compares our work with related work; Section III presents the dataset and a motivating example that explores the impact of football games on traffic congestion; Section IV formulates the problem; Section V presents the solution approach; Section VI evaluates the performance of our model; Section VII gives concluding remarks;

II. RELATED WORK AND CHALLENGES

This section presents an overview of the related work on traffic anomaly detection, which includes studies about faulty traffic sensor detection, traffic event detection, and congestion cause indication. Three key research challenges and our contributions for detecting NRC are discussed in the end.

Faulty Traffic Sensor Detection. Robinson et al. [4] proposed an approach that used data from inductive loop detectors to estimate travel time on road segments. His approach included a data cleaning method to clean the collected traffic data. Lu et al. [5] reviewed previous work on faulty inductive loops data analysis. Widhalm et al. [18] presented a traffic anomaly detection method that used Floating-Car Data (FCD) as an independent information source. They developed a non-linear regression model to fit the traffic sensor data and FCD data. Zygouras et al. [6] proposed a method comparing correlations among nearby sensors to identify faulty sensor readings. Their system was based on MapReduce paradigm to work for crowdsourcing data. Ghafouri et al. [7] presented a faulty traffic sensor detection model based on Gaussian Processes. Particularly, they provided an effective approach for computing the parameters of detectors to minimize the loss due to false-positive and false-negative errors.

Event Detection Using Traffic Data. Monitoring traffic flow at intersections is important in the traffic event detection research. Kamijo et al. [8] developed an algorithm based on spatiotemporal Markov random field (MRF) for processing traffic images and tracking vehicles at intersections. Using the timeseries observed behaviors of vehicles, a hidden Markov model for accident detection is then proposed. Veeraraghavan et al. [9] presented a multiple cue-based approach combined with a switching Kalman filter for detecting vehicle events such as turning, stopping and slow moving. Terroso-Senz et al. [19] presented an event-driven architecture (EDA) that used vehicular ad hoc network and external data sources like weather conditions to detect traffic congestions. Yang et al. [10] proposed a coupled Bayesian RPCA (BRPCA) model for detecting traffic events that used multiple traffic data streams. Kong et al. [11] proposed LoTAD to explore anomalous regions with long-term poor traffic situations. To model the traffic condition, crowd-sourced bus data is grouped into spatiotemporal segments. The segments with high anomaly indexes were combined to get anomalous regions. Wang et al. [20] proposed a two-stage solution to detect road traffic anomalies: (1) a Collaborative Path Inference (CPI) model that performs path inference incorporating static and dynamic

features into a Conditional Random Field (CRF); (2) a road Anomaly Test (RAT) model calculates the anomalous degree for each road segment.

Congestion Cause Indication. Liu et al. [12] studied both known (planned) and unknown (unplanned) events behaving differently from daily network traffics as anomalies, and proposed algorithms that construct outlier causality trees based on temporal and spatial properties of detected outliers. Xu et al. [13] introduced an approach to identify urban congestion patterns based on the data cube. They proposed a multi-dimensional data analysis method for data cube. Chow et al. [14] presented an automatic number plate recognition technology to analyze urban traffic congestions and introduced a linear regression model to indicate the causes of the congestions. Kwoczek et al. [16] proposed an Artificial Neural Network (ANN) based classifier to detect the road segments affected by planned events. Mallah et al. [17] evaluated the performance of machine learning techniques for classifying congestions into different causes.

A. Research Challenge 1: Representing Heterogeneous Traffic Data and Event Labels Using Multi-Dimensional Images

A feature vector is an n-dimensional vector and is the most popular representation of data objects. Besides numerical values, feature vectors can also represent texts and images. However, feature vectors may not be the best solutions for representing traffic and corresponding event labels.

Traffic conditions are highly affected by different influencing factors [21], such as incidents, sports games, road work, weather, etc. The events and their physical locations are used as the labels. But since feature vectors have fixed length, it is not practical to manually encode the labels to a specific fixed length feature vector. More importantly, in pattern recognition and machine learning, features matter the most. When converting an image to a feature vector, you can directly convert the two-dimensional pixels to a one-dimensional vector, or you can first take the histogram of the image and then construct a feature vector that has several comparison metrics, such as mean, standard deviation, etc. Both methods will lose some relative spatial information in the original images.

In contrast to feature vectors, images can preserve the original spatial relations by locating points on different pixels and can integrate multiple data sources by simply adding layers. Kwoczek et al. [16] showed a factor representation that integrates multiple features like event and weather into different layers in a data cube. However, though they mentioned the idea as a possible future work, they did not present any concrete solution to it. Ma et al. [22] proposed a CNN-based approach for traffic prediction. They represented the traffic speed and time using a time-space matrix. The problem with the time-space matrix is that the spatial information between segments is lost, which is important in detecting traffic patterns because nearby roads usually show similar or related patterns. Additionally, their model simply considered traffic data, but there are many other factors affecting the future traffic conditions. Thus representing heterogeneous traffic and

corresponding event labels using images remains a research gap.

One of the key differences between our proposed approach and the existing ones is that we are trying to visualize the wide area sensor data distribution as Traffic Condition Images (TCIs), so that we can use CNN and other deep learning techniques for analyses.

B. Research Challenge 2: Training Deep Learning Models Using Limited Data Instances

The performance of deep learning techniques highly relies on the quality of training data instances. However, the collected urban data may not provide enough data for training because of the data sampling rates. For example, our proposed model first converts traffic data to Traffic Condition Images (TCIs) and then trains different models using these images. But the traffic data we obtain from HERE [23] is requested every minute. So for a day that consists of 1440 minutes, we will only have 1440 traffic images, which are too few for effective training purposes. The availability issue of data instances becomes worse considering there is also limited label data. It remains a research challenge of getting more training data using the existing data.

Traditional ways of solving this problem are: (1) waiting and collecting until enough training data is collected, (2) manually labeling the data, (3) adding data sources, e.g., collecting more data from social media. Our solution uses the idea of crossover from the genetic algorithm. We assume that traffic conditions within a short time range are associated with the same events. So we can apply a crossover operator on the TCIs to generate more TCIs with the same event label.

C. Research Challenge 3: Modeling Traffic Patterns of Non-Recurring Events

The existing work on traffic event detection focused on analyzing traffic videos or traffic sensor data streams to detect events that are directly related to traffic, such as vehicle stopping, car accidents, and road congestion. But few studies explored the contextual non-recurrent events whose impacts are also highly associated with certain traffic patterns.

Recently there has been an explosion in research of using deep neural networks. But still, few have applied deep learning on studying traffic patterns. Deep learning techniques have gained great success in research fields like image processing, speech recognition, bioinformatics, etc. Convolutional neural networks are similar to original neural networks but convolutional layers are added in the front of the model to learn patterns in the original images. If traffic and label data can be converted to images, then CNN can be employed to learn their labeled patterns. It is still a research gap of how to develop an effective and efficient deep learning network for identifying and classifying traffic patterns of non-recurring events.

We formulate the problem of identifying the specific traffic patterns associated with events in Section V-A, and then present the details of our proposed approach that uses convolutional neural networks in Section V-C.

TABLE I.   DATA

	Traffic	Sports Game	Accident
Format	TMC	JSON	JSON
Source	HERE API [23]	ESPN, hockey-reference.com	Fire Department
Update	Every Minute	Manually	Manually
Size	155 GB	28 Games	387 MB
Range	10/2016 - Present	10/2016 - 12/2016	03/2014 - 03/2017

III.   DATA AND MOTIVATING EXAMPLE

This section first introduces the datasets that we have integrated into the system, and then describes a motivating example in which we use the collected datasets to study the impact of football games on the traffic congestion in the city.

A. Datasets

Since October 2016, we have been continuously collecting and storing real-time traffic data from HERE API [23] for all major roads in the Nashville area. In order to explore the impact of contextual events on urban mobility, we also collect the data about incidents and sports games. We cooperate with the Nashville Fire Department [24] to access their incident datasets, and manually collect the information about sports games from the web. As illustrated in Table I, the details of the datasets that we have integrated into the system are as follows:

- The traffic dataset provides the real-time traffic information on road segments, such as speed limit, real-time speed, jam factor (JF), etc. The dataset contains historical traffic data for 3049 TMC road segments in the Nashville area.
- The sports game dataset contains the operation information about sports games, such as game type, start and end time, attendance, location, etc.
- The incident dataset provides the detailed records of incidents and the responding vehicles. For each incident, it provides the coordinates, incident type, alert time, vehicle arrival and departure time, weather condition, etc.

B. Motivating Example

The motivation for our research comes from a brief experiment, in which we study the impact of football games on the traffic congestion in the city.

During the studied period between Sept. 1, 2016 and Jan. 1, 2017, there were eight football games (as listed in Table II) at the Nissan Stadium at downtown Nashville. During this time we collected data related to traffic (speed limit, real-time speed) and the football games (date, start time, duration, location)[1].

To indicate the congestion condition, HERE [23] provides a jam factor (JF) that ranges between 0.0 and 10.0 for each

---

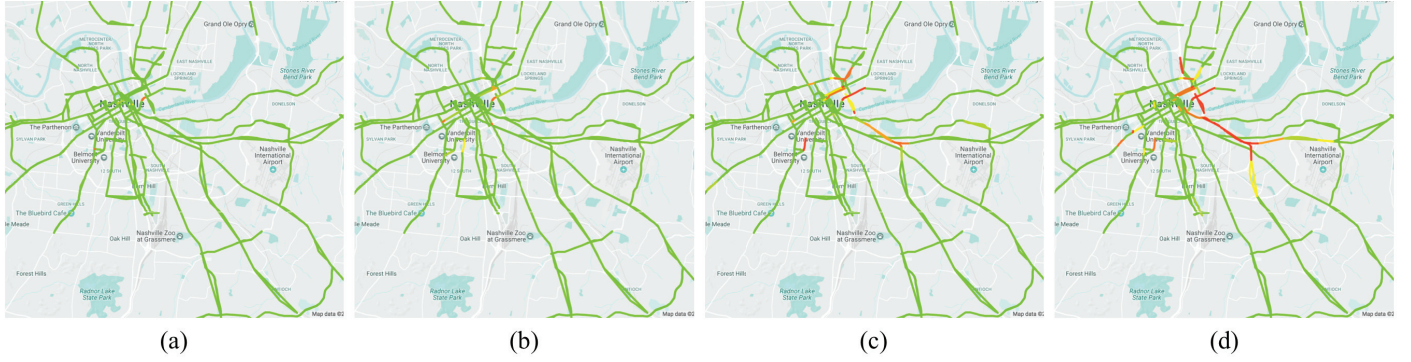[1]Our dataset is larger. However, in this study we are focusing on these 4 months

Fig. 1. Impact of football games on traffic congestion in four one-hour time windows before football games: (a) from 4 hours to 3 hours, (b) from 3 hours to 2 hours, (c) from 2 hours to 1 hour, (d) from 1 hour to 0 hour.

TABLE II. THE INFORMATION OF THE EIGHT FOOTBALL GAMES STUDIED IN THE MOTIVATING EXAMPLE

Date	Start Time (CST)	Stadium	Attendance	Duration
1/1/17	12:00 PM	Nissan Stadium	65205	3:11
12/11/16	12:00 PM	Nissan Stadium	68780	3:02
11/13/16	12:00 PM	Nissan Stadium	69116	3:36
10/27/16	7:26 PM	Nissan Stadium	61619	3:08
10/23/16	12:02 PM	Nissan Stadium	65470	3:21
10/16/16	12:02 PM	Nissan Stadium	60897	3:12
9/25/16	12:02 PM	Nissan Stadium	62370	3:03
9/11/16	12:05 PM	Nissan Stadium	63816	2:57

TMC road segment. In this study we compare the JF between the days when there is a football game and the days when there is no football game, during four one-hour time window directly before the games: $[-4, -3]$, $[-3, -2]$, $[-2, -1]$, and $[-1, 0]$ relative to the time when the game was scheduled[2].

As shown in Figure 1, the results of the JF difference on road segments in different time windows are visualized using heat maps. In the figure, colors ranging from green to red are used to indicate the small and big JF differences. The results show that the impact of football games on traffic congestion begins to increase from 4 hours before games. We have observed this pattern across several game events in the city. Our hypothesis is that every event has a unique pattern and we can learn that pattern over time and use it to identify if a current congestion pattern matches with the expected pattern. If the pattern does not match then we can classify it as an anomaly.

---

[2] Most football games are scheduled at 1 PM.

IV. PROBLEM FORMULATION

In this section, we first provide a formal definition of the problem and then describe the assumptions for solving the problem.

A. Definition

The goal of this research is to model traffic patterns around the locations of non-recurrent events so that we can use the model to identify non-recurring congestion and its causes. The traffic pattern that we use here refers to the spatiotemporal relations of traffic speeds on many road segments in an area, which can be modeled and detected by a classifier. The definitions of all relative notions can be found in Table III.

The inputs to the system are data about traffic and events. Since the traffic data that we collected from HERE API is defined using Traffic Message Channel Location Code [25] format (a standard for encoding geographic information), the road segments used in this study are also defined using the same TMC location codes. Event data is categorized with labels for training and validating purposes. The labels used are as follows:

- Event-related: event indicator l_{event}, time window relative to the event t_{event},
- Time-related: time in the day t_{day}, weekday t_{week}

One of the key differences between our approach and the existing ones is that we are trying to visualize the wide area sensor data distribution as Traffic Condition Images (TCI), so that we can use CNN and other deep learning techniques to analyze and model the spatiotemporal relations. TCI is a I_w by I_w pixels image. Each pixel p corresponds to a road segment in the real world and the grayscale value of each pixel represents the real-time traffic speed v_r of the road segment r.

Formulation of the Non-recurring Congestion Identification Problem. Given a set of traffic data S_t that contains speed limit and real-time speed for a set of road segments S_r at a specific time t_{day} on weekday t_{week}, and a set of event labels S_e, the model should determine l_{event} that indicates whether the given traffic data contains congestion caused by a subset S_e' of event set S_e. If l_{event} is true, the model should also

TABLE III.   SYMBOLS USED IN THE FORMULATED PROBLEM

T	a timestamp
t_{day}	time in the day in seconds
t_{week}	weekdays encoded using integers (e.g., 0 for Sunday, 1 for Monday, etc.)
t_{event}	time windows relative to events (e.g., 1 for the 1-hour time window before events)
l_{event}	indicator of whether the current time is within a time window near the occurrence of an event
S_e	a set of events in the city
r	a road segment
S_r	a set of road segments defined by TMC location codes
S_t	a set of traffic data that contains speed limit and real-time speed for a set of road segments S_r
TMC_{key}	a string representing a road segment in S_r
TCI	traffic Condition Image, a gray-scale image to represent traffic conditions in a bounding box
I_w	the width of a TCI
p	a pixel in TCI. Its value shows the normalized traffic speed on a road segment
v_r	the real-time traffic speed (miles per hour) on a road segment r
TH	a threshold for the classifier to determine whether the input traffic data contains recurring or non-recurring congestion

provide the time window t_{event} relative to events (i.e. t_{event} can be used to estimate the event occurrence time).

Figure 2 illustrates an example of the problem. Given raw traffic data at a specific time, the model should identify the possible non-recurring congestion and also provide an estimation of the event occurrence time.

B. Assumptions

The following assumptions are made when we design and formulate the non-recurring traffic congestion identification system:

- We assume the availability of both traffic speed data and event information for the studied area and period.
- We assume the traffic condition on a short road segment in a direction is the same anywhere on the segment.
- We assume that an event happening in the urban environment will affect the traffic conditions of nearby road segments.
- We assume that there is a robust correlation between the road segments affected by an event, and that the patterns can be identified by the image classification techniques of deep learning.

V. OUR APPROACH

In order to identify the specific traffic patterns associated with non-recurring events as defined in Section IV, we present

the details of our proposed approach in this section. The overall workflow of the system is shown in Figure 2. There are three key components in the system: (1) an algorithm that converts raw traffic data to images, (2) a convolutional neural network that classifies the traffic condition images, (3) ROC analysis that tunes the classification threshold to reduce the false positive and false negative rates.

A. Feature Extraction by Mapping Traffic Data to Images

Research challenge 1 describes the problem that feature vectors have limitations when representing urban data. To solve this issue, the first step is to convert the collected traffic data into images. We have been collecting real-time traffic data of Nashville area from Here Traffic API [23] since Oct. 2016. The traffic data is encoded in TMC location codes. Since the TMC database is not open to the public, here we present an algorithm to convert traffic data for a specific time T coded by TMC locations to traffic images. In order to project the traffic conditions to the pixels of images, we first initialize a gridded map and then re-sample the road segments defined by TMC location codes to the grids. The algorithm's input, output, and step details are as follows (for a set of road segments S_r, step 1 and 2 will run only once, but step 3 will run once for each timestamp):

Input: Traffic dataset S_t, road segment set S_r, and timestamp T. The raw traffic data of road segments S_r for timestamp T is queried from Traffic dataset S_t in the database.

Output: A Traffic Condition Image (TCI).

Step 1: Map grid initialization. The map of the area containing the road segment set S_r is divided into a map grid of squares. The length of each square is about 8.97 meters, so each grid cell covers about 80.51 square meters on the map.

Step 2: Road segment path re-sampling and smoothing. The points from road segments are re-sampled to the centers of grid cells if the points are covered by the cell. Also, if the distance between two original points is large enough that there are blank cells between the two cells projected by the two points, then points will be interpolated to fill the blank cells. After this step, we get a two-dimensional array, in which each cell contains a list of TMC keys TMC_{key} corresponding to points from road segments.

Step 3: Traffic data projection to the images. The two-dimensional array acts as a projecting table from original road segments to the image pixels. Now we can fill the images with traffic data by querying the traffic data using segment keys TMC_{key} and timestamp T. We use the following equation to convert a traffic speed to a pixel value:

$$p = \begin{cases} \frac{(80 - v_s^t)*255}{80}, & \text{if } 0 \leqslant v_s^t \leqslant 80 \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

where p denotes the pixel value (0-255) and v_s^t denotes the real-time speed (miles per hour).

After getting initial projected TCI, simple image processing techniques are used to resize TCI to the desired size (I_w by I_w).
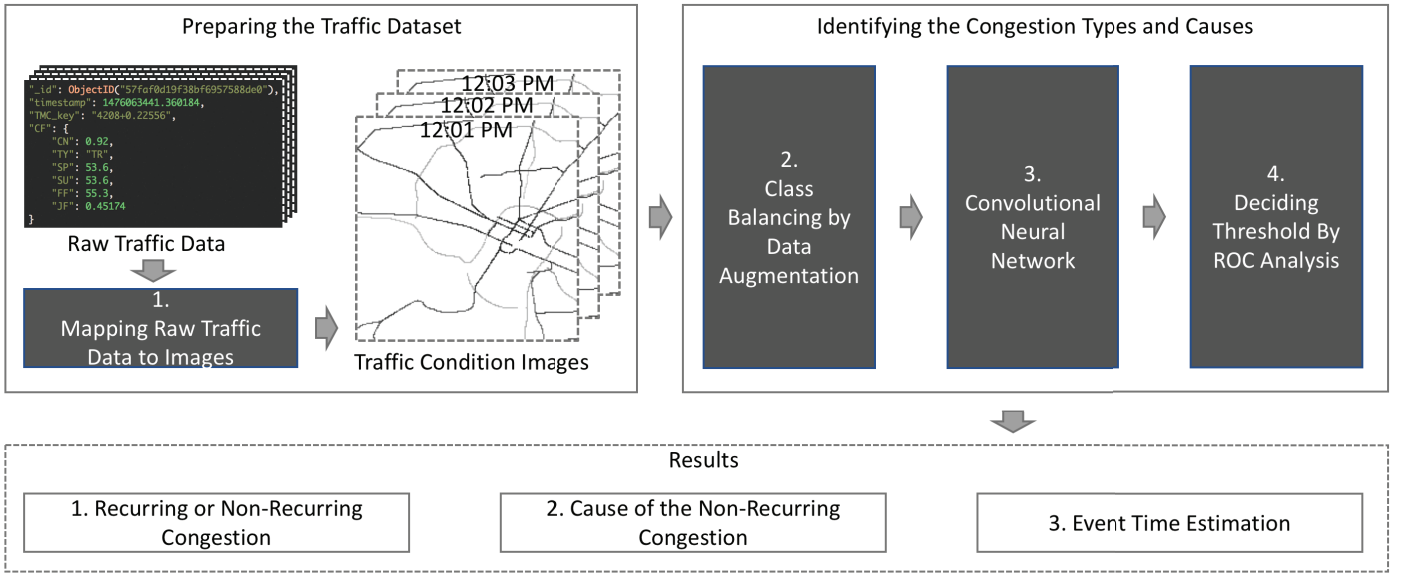
"_id": ObjectID("57faf0d19f38bf6957588de0"),
"timestamp": 1476063441.360184,
"TMC_key": "4208+0.22556",
"CF": {
    "CN": 0.92,
    "TY": "TR",
    "SP": 53.6,
    "SU": 53.6,
    "FF": 55.3,
    "JF": 0.45174
}

Preparing the Traffic Dataset

Raw Traffic Data

1. Mapping Raw Traffic Data to Images

12:03 PM
12:02 PM
12:01 PM

Traffic Condition Images

Identifying the Congestion Types and Causes

2. Class Balancing by Data Augmentation

3. Convolutional Neural Network

4. Deciding Threshold By ROC Analysis

Results

1. Recurring or Non-Recurring Congestion

2. Cause of the Non-Recurring Congestion

3. Event Time Estimation

Fig. 2. Overall workflow of the non-recurring congestion identification system

B. Data Augmentation by Crossover Operations

TCI is our image representation of traffic speeds on road segments. Since traffic data is collected every minute, without data augmentation we can only get 1440 (the number of minutes in a day) TCIs for one day, which is usually not enough for training deep learning image processing models. To address research challenge 2 (i.e., the lack of enough traffic data with labels), we create a crossover operator to generate more labeled traffic condition images for training deep learning models. Crossover is originally a genetic operator from genetic algorithms to vary the chromosomes of individuals from one generation to another. We are motivated by a similar idea and present the crossover operator for our system:

1) *Getting TCI candidates.* For a given timestamp T, instead of only getting one TCI for T, we generate n TCIs for time range $[T - t, T]$ (t denotes a time length to extend T, e.g., 3 minutes). While these TCIs are the same in image size, they differ in the pixel values because they correspond to traffic speeds at different times.

2) *Generating new labeled TCIs.* While looping through the pixels in TCI, for each pixel row there is a probability p_m that its values will mutate and randomly select a new row from the same corresponding pixels in other TCI candidates. After the second step, we get a new TCI. Because we assume traffic conditions within a small time range are caused by the same events, we can give the new TCI the same event label.

The crossover operator can be executed for many times to generate many new data instances. Through crossover, we not only have more labeled data, but also reduce the probability of over-fitting in the training phase.

C. Classifying Non-Recurring Congestion

In the previous section, we have described an algorithm that converts raw traffic data to TCI. Since the inputs contain images, it makes sense to apply convolutional neural networks. This section introduces our CNN model to classify the TCI using event labels. CNN is a class of deep and feed-forward artificial neural networks that have shown great success in image analysis tasks. Here we apply CNN to our problem that assigns event and congestion labels to a TCI.

CNN. The architecture of the proposed CNN is shown in Figure 3. Generally, the model consists of a stack of convolutional, fully-connected neural, dropout and max-pooling layers. Dropout layers are used throughout the model to prevent over fitting. Max pooling layers are used for spatial down-sampling. In the middle of the CNN, feature vectors that represent time of day and day of week that correspond to the TCI are concatenated to be input into the CNN to help it make better decisions. Details of the layer configuration, such as dimension, activation function, and dropout rate, can be found in the Figure 3. Since we use one-hot encoding and the vectors are in categorical format (i.e., dimensional vector is all-zeros except for a one at the index corresponding to the class of the sample), categorical cross entropy is used as the loss function to train the model.

One-hot Encoding. In the proposed CNN model, the input feature vectors are time in the day and weekday, and output labels are (1) whether the congestion in input TCI is recurring or non-recurring (2) the relative time windows that the TCI belongs to if it is non-recurring congestion. We use one-hot encoding to convert both input and output vectors to binary class matrix. The input matrix has 31 classes, in which 24 classes correspond to 24 hours and 7 classes correspond to 7 days of the week. As illustrated in Figure 4, the output matrix has several classes, of which the first class represents
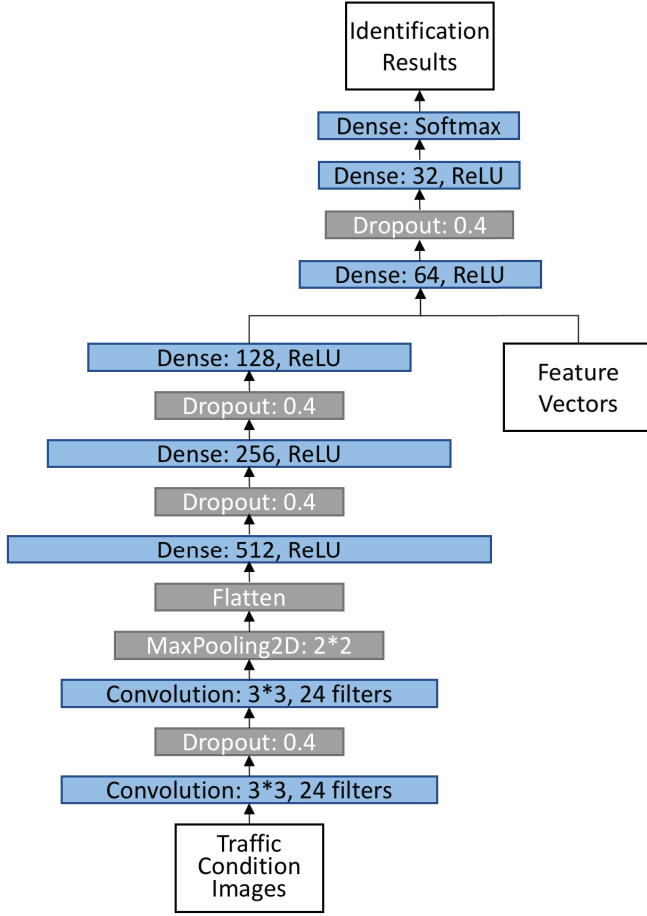
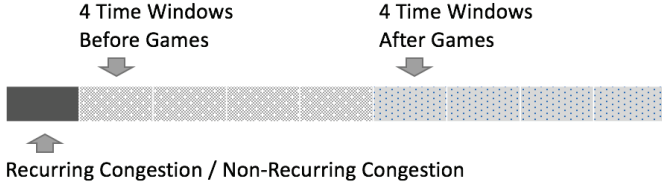Fig. 3. Our proposed convolutional neural network (CNN).

Fig. 4. An example of the one-hot encoding format. Event labels are encoded using 9 classes: (1) first digit represents whether the traffic condition belongs to recurring congestion or non-recurring congestion, (2) if it's non-recurring congestion, the next 8 digits represent 8 time windows before and after events.

whether the traffic condition belongs to recurring congestion or non-recurring congestion, and the next classes represent time windows before and after events. The first class is tunable since it directly determines whether the input traffic condition contains non-recurring congestion or not. If the value of the first class output is higher than a predefined threshold TH, then the classifier will output that the input traffic data does not contain non-recurring congestion (even if the values of other classes are higher than the first class). The details of the tuning steps are presented in the following section.

D. Tuning the Model Sensitivity by ROC Analysis

Our approach uses receiver operating characteristic (ROC) analysis to tune the sensitivity of the CNN classifier. ROC is a statistical plot that illustrates the diagnostic ability of a classifier system [26]. The ROC curve is a fundamental tool for diagnostic test evaluation. In an ROC curve, the true positive rate (TPR) is plotted in a function of the false positive rate (FPR). In machine learning, TPR represents sensitivity, recall or probability of detection, and FPR represents fall-out or false alarm [27]. By choosing a point from the curve, corresponding classification threshold can be decided.

In our model, the non-recurring congestion is considered as positive output and the recurring congestion is negative output.

We use the ROC analysis to tune the classification threshold that decides whether the traffic congestion in the input traffic data is recurring congestion or non-recurring congestion. We choose thresholds that range from 0.01 to 1.00 and the corresponding FPR and TPR of the training dataset are plotted (an example is shown in Figure 5). The curve's nearest point to the point (FPR: 0.0, TPR: 1.0) will be selected.
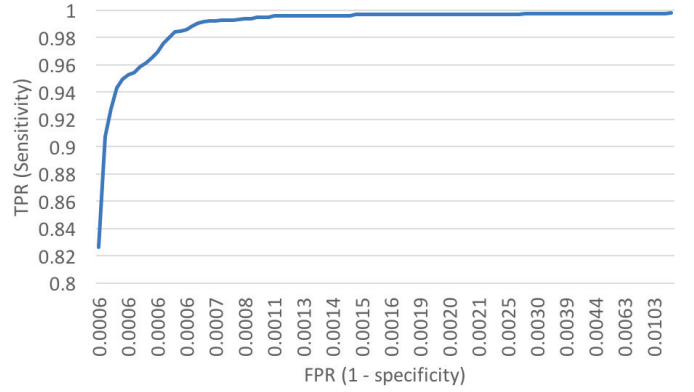

Fig. 5. Receiver operating characteristics (ROC) curve analysis on the prediction threshold.

VI. EXPERIMENTS

In this section, we evaluate the proposed deep neural network's ability to identify non-recurring traffic anomalies by using real-world data of three event types: football games, hockey games, and traffic incidents. Keras [28] Python deep learning library is used to construct the models and TensorFlow [29] is selected as the tensor manipulation library.

A. Scenarios

As illustrated in Figure 6, we create three scenarios to test the performance of the proposed model. In each scenario, we consider one of the three event categories for training and validating the proposed model:

- *Football Games.* Between Oct. 11, 2016, and Jan. 1, 2017, there were 8 NFL football games at the Nissan Stadium in Nashville. The traffic data in the bounding box (latitude
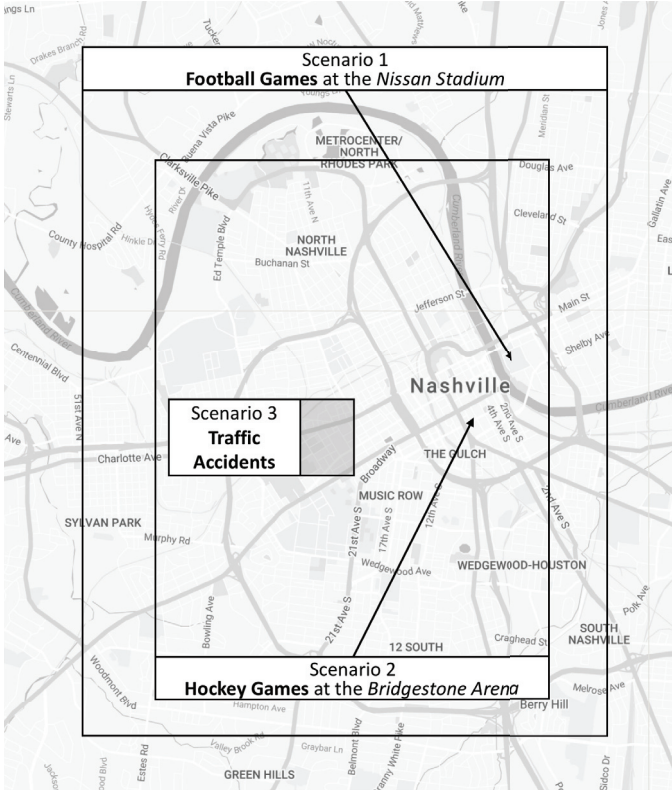
Fig. 6. Experimental scenarios and their coverage areas: (1) detecting NRC caused by football games, (2) detecting NRC caused by hockey games, (3) detecting NRC caused by traffic accidents.

	Accuracy	FPR	FNR
DxNAT	98.73%	1.57%	0.17%
Random Forest	84.06%	6.25%	2.17%

TABLE V. EXPERIMENT RESULTS IN SCENARIO 2: TRAINING DxNAT FOR IDENTIFYING NRC CAUSED BY HOCKEY GAMES

	Accuracy	FPR	FNR
DxNAT	90.76%	8.11%	23.19%

over 60,000 people, which shows a great impact on causing non-recurring traffic congestion. In the first scenario, we use the traffic data collected in 1-minute intervals between Oct. 11, 2016 and Jan. 1, 2017. Traffic data of 5 non-game days and two game days are used as the training dataset, and one non-game day and one game day are used as the validating dataset. As a comparison with the traditional machine learning techniques, we build a random forest model that uses the same training and validating dataset. Because random forests cannot use images directly as input, we first convert the traffic condition images to one-dimensional vectors, and then concentrate the traffic vectors with time of the day and day of the week vectors, and finally use the combined feature vector as input to the random forest model.

The accuracy, false positive rate (FPR) and false negative rate (FNR) of our model and the random forest model are shown in Table VI. Our model outperforms the random forest model with higher accuracy and lower FPR and FNR.

C. Experiment 2: Identifying NRC Caused by Hockey Games

Compared with NFL football games, NHL hockey games in Nashville usually have less attendance (NHL 10,000 v.s. NFL 60,000). So we assume that an NHL hockey game has less impact on traffic conditions and it will be more difficult to detect the NRC related to hockey games.

In Scenario 2, we use traffic and hockey games data between Oct. 14, 2016, and Nov. 30, 2016, as the training dataset, and data of Dec. 15, 2016 (game day) and Dec. 16, 2016 (non-game day) as the validating dataset. The accuracy, FPR and FNR results are shown in Table V. Compared with the results in Scenario 1, the model has lower accuracy and higher FNR. Our assumption is validated that the NRC associated with hockey games with less attendance is harder to be detected.

D. Experiment 3: Identifying NRC Caused by Traffic Accidents

In scenario 3, we explore the model's ability to detect NRC caused by road accidents. For the selected block area, there were eight traffic accidents on 7 different days between Oct. 18, 2016, and Dec. 13, 2016. We use six days with accidents as the training dataset and one day with an accident as the validating dataset. The DxNet model archives an accuracy of 86.59% with FPR of 13.71% and FNR of 4.44%

range: [36.1120, 36.2052], longitude range: [-86.8475, -86.7543]) is used.

- *Hockey Games.* Between Oct. 14, 2016, and Jan. 03, 2017, there were 20 NHL hockey games at the Bridgestone Arena in Nashville. The traffic data in the bounding box (latitude range: [36.1237, 36.1936], longitude range: -86.8359, -86.7660]) is used.
- *Traffic Accidents.* Between Oct. 15, 2016, and Mar. 10, 2017, there were 23 traffic accidents at the selected block area. The traffic data in the bounding box (latitude range: [36.1470, 36.1586], longitude range: [-86.8126, -86.8009]) is used.

The traffic and event datasets are divided into two subsets for training and validation. The event and time information is encoded using one-hot encoding as described in Section V. An example of the output of our model is shown in Figure 4. Particularly, the following metrics are used to define if an output is positive or negative: (1) an output is considered to be positive if it determines that the input TCI contains non-recurring congestion, (2) an output is negative if it determines the input TCI only contains recurring congestion.

B. Experiment 1: Identifying NRC Caused by Football Games

As shown in the motivating example in Section III-B, the 8 selected NFL football games have an average attendance of

TABLE VI. Experiment results in scenario 3: training DxNAT for identifying NRC caused by traffic accidents

	Accuracy	FPR	FNR
DxNAT	86.59%	13.71%	4.44%

TABLE VII. Summary of architectural decisions

Challenge	Approach	Section
Representing Heterogeneous Traffic Data and Event Labels	Using Multi-dimensional Images	II-A
Training Deep Learning Models Using Limited Data Instances	Developing crossover operator on original data	II-B
Modeling Traffic Patterns of Non-Recurring Events	Employing convolutional neural networks	II-C

VII. Conclusion

In this paper, we propose a deep neural network model to identify non-recurring traffic congestion and explain its causes. To our best knowledge, our work is one of the first efforts to utilize deep learning techniques to study traffic congestion patterns and explain non-recurring congestion using events. Our main contributions are listed in Table VII. We present an algorithm to efficiently convert traffic data in Traffic Message Channel (TMC) format to images, as well as a crossover operator as a data augmentation mechanism for class balancing. A convolutional neural network is proposed to identify non-recurring traffic anomalies that are caused by events. We evaluate the proposed model by using three types of events (football games, hockey games, and traffic incidents).

The future work to extend the current proposed model includes:

- *Integrating more contextual features.* Existing work usually focuses just on traffic data, but there are many types of urban data available to help identify traffic patterns, like real-time bus travel time, speed, and weather. Besides events, traffic conditions are affected by multiple environmental factors. The current work only considers time of day and day of week as the environmental training features. In the next step, we will include various features about weather conditions (such as humidity, nearest storm distance, visibility, etc.).
- *Identifying sizes of block areas and length of time windows.* For each event type (e.g., sports games, accidents), the size of impacting block areas as well as the number of impacting time windows in the experimental scenarios are selected arbitrarily. A mechanism is needed to automatically select the best impacting area size and time windows for each event type.

References

[1] D. Schrank, B. Eisele, T. Lomax, and J. Bak, "2015 urban mobility scorecard," 2015.

[2] R. W. Hall, "Non-recurrent congestion: How big is the problem? are traveler information systems the solution?" *Transportation Research Part C: Emerging Technologies*, vol. 1, no. 1, pp. 89–103, 1993.

[3] S. Lockwood, "The 21st century operation oriented state dots, nchrp project 20–24," *Transportation research board, American Association of State Highway and Transportation Officials, Washington, DC*, 2006.

[4] S. P. Robinson, "The development and application of an urban link travel time model using data derived from inductive loop detectors," Ph.D. dissertation, University of London, 2006.

[5] X.-Y. Lu, P. Varaiya, R. Horowitz, and J. Palen, "Faulty loop data analysis/correction and loop fault detection," in *15th World Congress on Intelligent Transport Systems and ITS America's 2008 Annual Meeting*, 2008.

[6] N. Zygouras, N. Panagiotou, N. Zacheilas, I. Boutsis, V. Kalogeraki, I. Katakis, and D. Gunopulos, "Towards detection of faulty traffic sensors in real-time." in *MUD@ ICML*, 2015, pp. 53–62.

[7] A. Ghafouri, A. Laszka, A. Dubey, and X. Koutsoukos, "Optimal detection of faulty traffic sensors used in route planning," in *Proceedings of the 2nd International Workshop on Science of Smart City Operations and Platforms Engineering*. ACM, 2017, pp. 1–6.

[8] S. Kamijo, Y. Matsushita, K. Ikeuchi, and M. Sakauchi, "Traffic monitoring and accident detection at intersections," *IEEE transactions on Intelligent transportation systems*, vol. 1, no. 2, pp. 108–118, 2000.

[9] H. Veeraraghavan, P. Schrater, and N. Papanikolopoulos, "Switching kalman filter-based approach for tracking and event detection at traffic intersections," in *Intelligent Control, 2005. Proceedings of the 2005 IEEE International Symposium on, Mediterrean Conference on Control and Automation*. IEEE, 2005, pp. 1167–1172.

[10] S. Yang, K. Kalpakis, and A. Biem, "Detecting road traffic events by coupling multiple timeseries with a nonparametric bayesian method," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 5, pp. 1936–1946, 2014.

[11] X. Kong, X. Song, F. Xia, H. Guo, J. Wang, and A. Tolba, "Lotad: long-term traffic anomaly detection based on crowdsourced bus trajectory data," *World Wide Web*, pp. 1–23, 2017.

[12] W. Liu, Y. Zheng, S. Chawla, J. Yuan, and X. Xing, "Discovering spatio-temporal causal interactions in traffic data streams," in *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2011, pp. 1010–1018.

[13] L. Xu, Y. Yue, and Q. Li, "Identifying urban traffic congestion pattern from historical floating car data," *Procedia-Social and Behavioral Sciences*, vol. 96, pp. 2084–2095, 2013.

[14] A. H. Chow, A. Santacreu, I. Tsapakis, G. Tanasaranond, and T. Cheng, "Empirical assessment of urban traffic congestion," *Journal of advanced transportation*, vol. 48, no. 8, pp. 1000–1016, 2014.

[15] S. Kwoczek, S. Di Martino, and W. Nejdl, "Predicting and visualizing traffic congestion in the presence of planned special events," *Journal of Visual Languages & Computing*, vol. 25, no. 6, pp. 973–980, 2014.

[16] ——, "Stuck around the stadium? an approach to identify road segments affected by planned special events," in *Intelligent Transportation Systems (ITSC), 2015 IEEE 18th International Conference on*. IEEE, 2015, pp. 1255–1260.

[17] R. Al Mallah, A. Quintero, and B. Farooq, "Distributed classification of urban congestion using vanet," *IEEE Transactions on Intelligent Transportation Systems*, 2017.

[18] P. Widhalm, H. Koller, and W. Ponweiser, "Identifying faulty traffic detectors with floating car data," in *Integrated and Sustainable Transportation System (FISTS), 2011 IEEE Forum on.* IEEE, 2011, pp. 103–108.

[19] F. Terroso-Sáenz, M. Valdés-Vela, C. Sotomayor-Martínez, R. Toledo-Moreo, and A. F. Gómez-Skarmeta, "A cooperative approach to traffic congestion detection with complex event processing and vanet," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 2, pp. 914–929, 2012.

[20] H. Wang, H. Wen, F. Yi, H. Zhu, and L. Sun, "Road traffic anomaly detection via collaborative path inference from gps snippets," *Sensors*, vol. 17, no. 3, p. 550, 2017.

[21] J. Kwon, M. Mauch, and P. Varaiya, "Components of congestion: Delay from incidents, special events, lane closures, weather, potential ramp metering gain, and excess demand," *Transportation Research Record: Journal of the Transportation Research Board*, no. 1959, pp. 84–91, 2006.

[22] X. Ma, Z. Dai, Z. He, J. Ma, Y. Wang, and Y. Wang, "Learning traffic as images: a deep convolutional neural network for large-scale transportation network speed prediction," *Sensors*, vol. 17, no. 4, p. 818, 2017.

[23] "Here traffic api," https://developer.here.com/rest-apis/documentation/traffic/topics_v6.1/flow.html.

[24] . M. G. of Nashville and D. County, "Nashville fire department," 2017, [Online; accessed 30-September-2017]. [Online]. Available: http://www.nashville.gov/Fire-Department.aspx

[25] O. Wiki, "Tmc/location code list/location types," 2017, [Online; accessed 12-September-2017]. [Online]. Available: http://wiki.openstreetmap.org/wiki/TMC/Location_Code_List/Location_Types

[26] J. A. Hanley and B. J. McNeil, "The meaning and use of the area under a receiver operating characteristic (roc) curve." *Radiology*, vol. 143, no. 1, pp. 29–36, 1982.

[27] MathWorks, "Detector performance analysis using roc curves," 2017, [Online; accessed 30-September-2017]. [Online]. Available: https://www.mathworks.com/help/phased/examples/detector-performance-analysis-using-roc-curves.html

[28] Keras, "Keras: The python deep learning library," 2017, [Online; accessed 30-September-2017]. [Online]. Available: https://keras.io/

[29] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin *et al.*, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," *arXiv preprint arXiv:1603.04467*, 2016.