DETECTION AND DESCRIPTION OF TRAFFIC EVENTS USING FLOATING CAR AND SOCIAL MEDIA DATA

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

AHMET DÜNDAR ÜNSAL

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF DOCTOR OF PHILOSOPHY
IN
GEODETIC AND GEOGRAPHIC INFORMATION TECHNOLOGIES

SEPTEMBER 2022

Approval of the thesis:

DETECTION AND DESCRIPTION OF TRAFFIC EVENTS USING FLOATING CAR AND SOCIAL MEDIA DATA

submitted by **AHMET DÜNDAR ÜNSAL** in partial fulfillment of the requirements for the degree of **Doctor of Philosophy** in **Geodetic and Geographic Information Technologies, Middle East Technical University** by,

Prof. Dr. Halil Kalıpçılar
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. Zuhal Akyürek
Head of the Department, **Geodetic and Geographic Information Technologies**

Prof. Dr. Hediye Tüydeş-Yaman
Supervisor, **Civil Engineering, METU**

Prof. Dr. Pınar Karagöz
Co-Supervisor, **Computer Engineering, METU**

Examining Committee Members:

Prof. Dr. Zuhal Akyürek
Civil Engineering, METU

Prof. Dr. Hediye Tüydeş-Yaman
Civil Engineering, METU

Prof. Dr. Ayşen Akkaya
Statistics, METU

Asst. Prof. Dr. Oruç Altıntaşı
Civil Engineering, İzmir Katip Çelebi University

Assoc. Prof. Dr. Berk Anbaroğlu
Geomatics Engineering, Hacettepe University

Date: 12.09.2022

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name Last name : Ahmet Dündar Ünsal

Signature :

ABSTRACT

DETECTION AND DESCRIPTION OF TRAFFIC EVENTS USING FLOATING CAR AND SOCIAL MEDIA DATA

Ünsal, Ahmet Dündar
Doctor of Philosophy, Geodetic and Geographic Information Technologies
Supervisor: Prof. Dr. Hediye Türdeş-Yaman
Co-Supervisor: Prof. Dr. Pınar Karagöz

September 2022, 150 pages

Detection and verification of traffic events, in traffic management, can be performed traditionally using roadside sensor data. More recently wide coverage travel time information obtained from floating car data (FCD) is also used, despite its limitations to describe the event characteristics and requires verification. Social media, widely adopted in our daily lives, hosts a sheer amount of data which can be analyzed to identify incidents and events using information retrieval methods. In this study, a framework is proposed to detect and describe traffic events in real-time using two independent data sources, FCD and Social Media Data (SMD). Traffic event related tweets in SMD are classified using a language model which is tailored to handle agglutinative nature of Turkish language. Detected traffic event tweets are geolocated using a custom named-entity recognition (NER) integrated, knowledge-based geocoding approach, which achieves a median positional error of 379.2 meters. In FCD, proposed detection tasks identified non-recurrent congestions (NRCs) with their spatiotemporal impact areas. Matching experiments using spatiotemporal information showed that 64.1% of traffic event reporting tweets can be verified by an NRC, whereas only 33% of the large-scale NRCs are verified by a tweet.

Keywords: Traffic Event Detection, Intelligent Transportation Systems, Floating Car Data, Social Media

ÖZ

HAREKETLİ ARAÇ VE SOSYAL MEDYA VERİSİ KULLANARAK TRAFİK OLAYLARININ ALGILANMASI VE TANIMLANMASI

Ünsal, Ahmet Dündar
Doktora, Jeodezi ve Coğrafi Bilgi Teknolojileri
Tez Yöneticisi: Prof. Dr. Hediye Tüydeş-Yaman
Ortak Tez Yöneticisi: Prof. Dr. Pınar Karagöz

Eylül 2022, 150 sayfa

Trafik yönetiminde, trafik olaylarının algılanması ve doğrulanması, karayollarına yerleştirilen sensör verileri kullanılarak gerçekleştirilmiştir. Daha yeni bir yaklaşım olarak, hareketli araç verilerinden (FCD) elde edilen geniş kapsamlı seyahat süresi bilgisi de aynı amaçla kullanılır, ancak bu veri de olay niteliklerini tanımlamada kısıtlıdır ve doğrulama gerektirir. Günlük hayatımızda yaygın olarak kullanılan sosyal medyanın ev sahipliği yaptığı büyük veri, olayların algılanmasında, bilgi getirimi yöntemleri kullanılarak analiz edilebilir. Bu çalışmada, iki bağımsız veri kaynağı olan FCD ve Sosyal Medya Verileri (SMD) kullanılarak, trafik olaylarını gerçek zamanlı olarak algılamak ve tanımlamak için bir çerçeve önerilmiştir. SMD'deki trafik olayları, Türkçe'nin sondan eklemeli yapısı dikkate alınarak özelleştirilmiş bir dil modeli kullanılarak sınıflandırılır. Algılanan trafik olayı tweet'lerinin konumları, özelleştirilmiş bir varlık adı tanıma (NER) sistemine entegre çalışan bilgi tabanlı coğrafi kodlama yaklaşımı ile, 379.2 metre medyan hata ile belirlenmiştir. FCD üzerinde önerilen yöntemler ile, tekrarlamayan sıkışıklıklar (NRC) algılanabilir ve mekan-zamansal etki alanları belirlenebilir Mekan-zamansal bilgi eşleştirme yöntemini kullanan eşleştirme deneylerinde, trafik olayı bildiren tweet'lerin %64,1'i bir NRC tarafından doğrulanabilmiştir, diğer taraftan büyük ölçekli NRC'lerin ancak %33’ü en az bir tweet ile doğrulanmıştır.

Anahtar Kelimeler: Trafik Olayı Algılanması, Akıllı Ulaşım Sistemleri, Hareketli Araç Verisi, Sosyal Medya

To my family.

ACKNOWLEDGMENTS

I would like to express my deepest gratitude to my supervisor Prof. Dr. Hediye Tüydeş-Yaman, for her guidance throughout the study and for being an excellent mentor and a great inspiration. I’m extremely grateful to my co-supervisor Prof. Dr. Pınar Karagöz, without her valuable insights, guidance and support I would not have been able to complete this research. I would also like to thank Prof. Dr. Zuhal Akyürek and Prof. Dr. Ayşen Akkaya for their suggestions, comments, and strong support.

Many thanks to companies, ISSD and Parabol, and their devoted staff including, but not limited to, Şule Yücel, Metin Barış, Fulya Özsan and Nurgazy Nazhimidinov for the access to FCD database and support during the study. I am also thankful to Mert Kürkçüoğlu for his friendship and support.

My special thanks go to Assoc. Prof. Dr. Nurunnisa Usul for encouraging me along this path.

TABLE OF CONTENTS

CHAPTERS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

AF	Anomaly Factor
AID	Automatic Incident Detection
API	Application Programming Interface
BiLSTM	Bidirectional Long Short-Term Memory
CRF	Conditional Random Fields
FCD	Floating Car Data
FN	False Negative
FP	False Positive
GESD	Generalized Extreme Studentized Deviate
GIS	Geographic Information Systems
GPS	Global Positioning System
HMM	Hidden Markov Model
HTTP	Hypertext Transfer Protocol
IOB	Inside-Outside-Beginning
IR	Information Retrieval
ITS	Intelligent Transportation Systems
LSTM	Long Short-Term Memory
MAD	Median Absolute Deviate
ML	Machine Learning
NB	Naïve Bayes
NER	Named Entity Recognition

NLP	Natural Language Processing
NLTK	Natural Language Toolkit
NRC	Non-recurrent Congestion
OSM	OpenStreetMap
POI	Point of Interest
REST	Representational State Transfer
RC	Recurrent Congestion
RNE	Road Network-based Estimator
SMD	Social Media Data
SND	Standard Normal Deviate
SVM	Support Vector Machine
TEER	Traffic Event Entity Recognizer
TEG	Traffic Event Geocoder
TF-IDF	Term Frequency – Inverse Document Frequency
TN	True Negative
TOD-DOW	Time of Day – Day of Week
TP	True Positive
XML	Extensible Markup Language

CHAPTER 1

INTRODUCTION

As a part of smart city framework, traffic management systems, which aim to provide measures for a sustainable and optimal network, require monitoring of the traffic flow. A traffic event is defined as a non-recurring incident which can lead to a reduction in road capacity or an increase in demand, such as traffic accidents, car breakdowns, etc. (Neudorff et al., 2003). Congestion and delays caused by such incidents cost commuters time and money, and disrupt the traffic flow, thus decreasing overall capacity of an urban road network. Detection and management of traffic incidents is an important topic in Intelligent Transportation Systems (ITS). Timely detection of a traffic events can help alleviate the burden of the consequent congestion on the network. Traffic flow on major highways is commonly monitored using sensors that are deployed on roads, such as inductive loops or camera systems, which employ computer vision technologies. Due to high deployment and maintenance cost of such sensor systems, it is impractical to cover urban traffic networks with adequate number of sensors.

Floating car data (FCD) obtained from Global Positioning System (GPS) equipped vehicles emerge as a high coverage and cost-effective data source alternative in traffic management applications. Increasing number of GPS equipped vehicles enables better monitoring of traffic networks in real-time. Commercial FCD is processed outcome of track data of fleet of GPS-equipped vehicles, providing average speed (or travel time) data for predefined road segments at given time intervals (i.e. 1-min, 5-min, etc.). Using highly granular road network segments in FCD makes them a useful data source for detecting traffic events. However, event detection using FCD beg confirmation, similarly to methods using other sensor data sources, due to limitations of the data content to describe the events.

Social media, which are Internet-based social interaction networks, allow sharing of information, experiences, and opinions in various forms. Social media platforms have high adaptation rates. 313 million monthly active Twitter users send an average of around 500 million tweets a day (Krikorian, 2013; Twitter, 2016). Social media hosts a large data including happenings and events, hence social media streams are useful resources to detect real-world events. Social media data (SMD) emerge as a new information source in smart city applications. The information shared by users or *human sensors* as commonly referred, can complement the data provided by other traffic sensors, such as FCD. Thus, SMD along with FCD can be useful data sources to detect and confirm traffic events.

1.1 Research Objectives

The main objective of this research is to study and present methods which enable real-time detection and confirmation of traffic events using FCD and Twitter streams. Detection methods for each data stream are evaluated independently.

- Traffic event reporting Twitter posts are detected using information retrieval techniques accompanied by natural language processing methods. A custom traffic event geocoder is evaluated for precise localization of traffic event related tweets.
- Non-recurring congestions, as possible indicators of traffic events, are inspected in FCD using statistical and data-driven methods.
- A spatiotemporal information matching method is applied to confirm events detected using two distinct data sources.

1.2 Contribution

The main contribution of this thesis is to present a framework to detect traffic events using low-cost and high coverage data sources, social media, and floating car data. Novel approaches in the framework can be listed as follows:

- Use of morphological analysis to extract *subtokens* to improve text classification and named entity recognition in a highly agglutinative language namely Turkish.
- A traffic event geocoder, which constitutes of a named entity recognition model customized for traffic event term detection accompanied with a rule-based geocoder for precise localization of traffic events in an urban road network.
- A novel metric for travel speed anomaly detection, which make use of wide coverage of an FCD.

1.3 Thesis Structure

The work in this study is presented in 6 chapters. Existing research in 1) event detection in social media, 2) localization of social media messages, and 3) non-recurrent congestion detection are presented in Chapter 2. In Chapter 3, the methods to detect traffic event related tweets are investigated. In Chapter 4, a custom traffic event geocoder to localize traffic event related tweets is described. In Chapter 5, data-driven methods to detect non-recurrent congestion in floating car data (FCD) are presented. In Chapter 6, the experiments and results of the proposed methods are presented on a case study in Ankara. In Chapter 7, a conclusion of the study with a summary of the findings is given.

CHAPTER 2

LITERATURE REVIEW

In this study, a traffic event detection framework is presented which consist of modules to detect and localize traffic events in social media data and identify non-recurrent congestions in floating car data. In this chapter a brief summary of the research in the correspoding domains is presented.

2.1 Traffic Event Detection in Social Media

On the literature, there are various studies that use social streams for detecting real-world incidents such as fires (Abel et al., 2012), earthquakes (Sakaki et al., 2010), and traffic accidents (Schulz et al., 2013b). As one of the basic approaches, sudden occurrence of terms in a short interval of time, which is described by term "burstiness", was examined to detect events in text streams (Fung et al., 2005). Abdelhaq et al. (2013) detected local events by clustering identified bursty words per their spatial similarity and time frames. Li et al. (2012a) proposed a segment-based event detection method based on burstiness and content similarity. Kleinberg (2003) modeled streams using an infinite-state automaton to detect bursty sequences.

Studies employing machine learning methods for text classification were able to detect incidents regardless of their scale (Gutierrez et al., 2015; Schulz et al., 2013b). Twitter stream was used for detecting incidents, which concern transportation (Chen et al., 2014; D'Andrea et al., 2015; Gutierrez et al., 2015; Schulz et al., 2015). Chen et al. (2014) developed a unified framework based on hinge loss Markov random fields to combine the models proposed for language ambiguity and location uncertainty in tweets. Kurkcu et al. (2015) developed a framework to provide a travel time data collection method as incidents occur. Jalaparthi and Kumar (2016) presented a real-time monitoring system for detection of traffic events from Twitter stream Nguyen et al. (2016) employed Conditional Random Fields (CRF) to analyze social media for traffic incident detection. Deep learning methods including deep belief network, convolutional neural network and recurrent neural network are used to detect traffic incidents using microblog posts from Twitter or Weibo (Chen et al., 2019; Dabiri and Heaslip, 2019; Zhang et al., 2018).

Studies for event detection in text streams are rare for Turkish language. Can et al. (2010) presented methodology to perform topic detection and tracking in Turkish language. Erdogan et al. (2017) analyzed tweets in Turkish language in order to monitor events. Methods including named entity recognition, multinomial naive Bayes and stochastic gradient descent are used to detect events. Ertugrul et al. (2017) used word embeddings to detect events in social media posts in Turkish. Genc and Yilmaz (2019).employed graph embeddings to extract key information from Turkish social media texts to detect events.

2.2 Localizating Traffic Events

Extracting locations from social media data has been an attractive research problem. With its large and publicly available content, Twitter has become a useful resource for studies concerning spatial aspects of events and incidents. Geolocation of tweets has been studied in various domains, including disaster monitoring, traffic management, health monitoring and marketing (Ozdikis et al., 2017).

Location information is commonly extracted using three fields of tweets: 1) a *geo-tag field*, is the geographical coordinates of the post location of a tweet, set optionally; 2) *location field* is an optional text field set by users manually; 3) text content of a tweet. Due to scarcity of geotagged tweets, *geo-tag field* is not directly used to localize events, rather used as an input for constructing models to locate tweets without geo-tags (Kinsella et al., 2011; Paraskevopoulos and Palpanas, 2015; Paule et al., 2019) or as a ground-truth for verification (Priedhorsky et al., 2014; Schulz et al., 2013a). *Location field* is an attribute of a user profile, thus could be used to describe where a user resides. However, most of the users do not provide a valid or a specific location name, hence it is not commonly used considered as a useful resource for location extraction (Davis et al., 2011). Some studies utilized *location field* as a complementary reference along with other fields (Li et al., 2012b; Sakaki et al., 2012).

Location extraction research can be classified into three groups by their localization focus: 1) user location, the place of residence of the user 2) event location*,* where the mentioned event or incident took place. User location is used in numerous applications, such as disease monitoring, marketing, recommendation systems (Ozdikis et al., 2017; Zheng et al., 2018). Given the scarcity of reliable location information in the tweet meta-data, message content of a tweet is used as a reference for locating users. Language models based on toponyms or terms implicitly refer to a geographic location, also referred as location indicative words, have been proposed to estimate user locations (Cheng et al., 2010; Han et al., 2014; Roller et al., 2012; Zheng et al., 2018). Some studies presented models to utilize multiple sources of location data for localization (Laylavi et al., 2016; Ozdikis et al., 2016; Schulz et al., 2013a). Social relationships in Twitter are also employed to detect or enrich locations (Bakerman et al., 2019; Li et al., 2012c; Rodrigues et al., 2016). Granularity of user localization varied among the studies from city level (Cheng et al., 2010; Davis et al., 2011; Han et al., 2014; Li et al., 2012c; Mahmud et al., 2014; Rodrigues et al., 2016) to fine-grained coordinates. (Ryoo and Moon, 2014)

Localization of events, such as traffic accidents, natural disasters, fires, through social media has been studied extensively. Geo-tag and location fields are used to determine location for large scale events (Earle et al., 2011; Sakaki et al., 2010). However, location elements in tweet meta-data are not adequate for localizing small scale events or individual tweets, due to scarcity and reliability problems. Geotagged tweets, however, are used to associate location indicative words with geographic references, such as administrative boundaries, uniform or adaptive grid cells, which enabled localizing of non-geotagged tweets (Kinsella et al., 2011; Paraskevopoulos and Palpanas, 2015; Paule et al., 2019). Fine grained localization of tweet content based on such data-driven models using geotagged tweets were presented in numerous studies (Flatow et al., 2015; Paraskevopoulos and Palpanas, 2015; Paule et al., 2019). Various other studies involve temporal dimension in their methods for fine-grained localization of tweets (Chong and Lim, 2017; Li and Sun, 2017). Another approach to localize tweets is to employ explicit location references in tweet content. Twitter users tend to provide a geographic reference to the events they report (Longueville et al., 2009). The most common approach taken to localize location references in the tweet content consist of the phases: 1) extraction of location terms in text content, 2) mapping detected terms onto geographic locations, commonly using a knowledgebase. There are basically two main approaches in the literature for extracting locations from text content: natural language processing (NLP) based methods and term search against gazetteer databases (Karagoz et al., 2016). Named entity recognition (NER), which is an NLP method, is commonly employed to detect location references in tweets (Gelernter and Balaji, 2013; Gelernter and Mushegian, 2011; Nguyen et al., 2016; Schulz et al., 2013b; Zhang and Gelernter, 2014). Location references are also identified by matching content with place names listed in a dictionary or a gazetteer (Gu et al., 2016; Middleton et al., 2014; Ribeiro et al., 2012). Detected location terms are localized using off-the-self geocoders such as Google Map Geocoding API, ArcGIS Geocoder, Yahoo's PlaceFinder API, Nominatim or custom geocoders.

Geocoders are commonly used in traffic event detection studies to localize traffic events on road networks. Sakaki et al. (2012) extracted location terms from traffic information tweets by building a web-based location name dictionary which is complemented also with a name entity recognition approach. Detected location terms are geocoded using Google Map Geocoding API. Schulz et al. (2013) used Standford NER to extract location terms in detected traffic events and used Nominatim to geocode the location terms. Gutierrez et al. (2015) evaluated various NER engines to extract locations from classified traffic events, and used Google Map Geocoding API, Nominatim and Geonames to geocode the events. Gu et al (2016), used regular expressions and a fuzzy language matching algorithm given in Gelernter and Balaji (2013) to geo-parse traffic incident tweets detected from Twitter. They use ArcGIS geocoder and a custom highway geocoder to locate the geo-parsed incidents. Khan et al. (2020) used NYC Geoclient and Geopy packages for geocoding locations from tweets parsed by NER of NLTK module. Vallejos et al. (2021) proposed a NER based on regular expressions and approximate string matching to address informal writing in tweets. Detected terms are geocoded using Google Geocoding API. Luan et al. (2021) used part of speech filtering based on a list of point of interest (POI) and street names to detect location terms in traffic events and geocoded the events using Amap geocoding engine. Suat-Rojas et al. (2022), used CRF, BiLSTM and Spacy methods to identify named entities. Extracted named entities are geocoded using Batch Geocode package, which combines the results from Google Maps, OpenStreetMap and GeoNames. Some studies proposed custom geocoding methods which employ place names in traffic event tweets, such as street names, landmarks, street crossings (Ribeiro et al., 2012; Wang et al., 2015).

Although precise localization is critical for most traffic management applications, spatial errors from localization based on named entity recognition accompanied by geocoders are rarely reported. Khan et al. (2020) reported that NYC Geoclient and Geopy packages were able to assign locations to NER detected location terms within an average of 7.3 miles of the geo-tagged location. Suat-Rojas et al. (2022) reported more than 1 km between the actual and estimated coordinates due to insufficient data in Google Maps and OpenStreetMap for cities outside the United States. They called for further analysis of estimation errors and city specific geocoder as a future study.

2.3 Non-recurrent Congestion Detection

Initial works on incident detection focus on incidents taking place in freeways. Early studies leveraged data acquired from roadside sensor stations, such as inductive loop detectors. California algorithm, which is based on a decision tree with states is among the most notable early methods to detect traffic incidents on freeways (Payne and Tignor, 1978). Algorithm developed further and variations are used as a baseline to other automatic incident detection (AID) algorithms (Masters et al., 1991). Dudek et al. (1974) presented an AID method using the standard normal deviate (SND) of energy or lane occupancy. Method was evaluated on an urban freeway with double-loop sensors. Ishak and Al-Deek (1999) applied two types of artificial neural networks to detect incidents on a freeway corridor using loop detector data. Karim and Adeli (2002) presented a wavelet analysis-based algorithm to separate patterns produced by incidents from those of recurrent congestions and compression waves. Yuan and Cheu (2003) used Support Vector Machines to classify incident and incident-free stretches using data collected from loop detectors on a freeway. With the advance of mobile technologies, position data collected from probe vehicles emerged as an alternative data source for incident detection studies. Balke et al. (1996) used probe vehicle data to detect incidents by applying standard normal deviate (SND) to travel times. Petty et al. (1997) presented probe-vehicle data-based incident detection algorithm, proposing probe-vehicle data as a viable data source for incident detection, addressing difficulties with loop detector-based systems, such as cost of expansion, and sparse coverage. Cheu (2002) detected incidents by comparing average section travel times from probe vehicle data before and during an incident. Li and McDonald (2005) used bivariate analysis model using the average travel times and the travel time differences between adjacent time intervals to detect incident in probe vehicle data in a motorway. Asakura et al. (2015) investigated the traffic flow dynamics during incidents and proposed incident detection methods using probe vehicle-data in an urban freeway.

Incident detection in urban arterials has its own challenges due to interrupted traffic flow, traffic controls and complexity of network (Zhang and Taylor, 2006). Hoose et al. (1992) proposed an image analysis algorithm to detect incident on congestions using video images from urban streets. Sethi et al. (1995) developed and evaluated incident detection algorithms using data collected from fixed detectors and probe vehicles in urban arterial streets. Ivan et al. (1995) detected incidents on arterial street network using data from: fixed detectors and probe vehicles using neural network-based approaches. Sermons and Koppelman (1996) investigated incident and non-incident traffic patterns using discriminant classification models on 1-second vehicle positioning data collected in urban arterial road segments. Thomas (1998), and Zhang and Taylor (2006) solved arterial incident detection problem with Bayesian based decision making approaches.

In traffic studies, congestions are commonly classified into recurrent and non-recurrent congestions. Recurrent congestions (RC) take place mostly in urban road due to daily demand patterns. Non-recurrent congestions (NRC) occur due to incidents, such as traffic accidents, car breakdowns, special events or weather conditions (Neudorff et al., 2003).

NRC can occur frequently whereas RC does not always repeat with the daily demand pattern, so identifying recurrent congestion by their frequency of recurrence is not possible (Dowling et al., 2004). Common inference indicates that non-recurrent congestions constitute more than half of the congestion delays (Hall, 1993). Recent studies on detection of incidents or events causing delays on urban arterial networks focus on identification of non-recurrent congestions. Anbaroglu et al. (2014) presented a clustering-based approach to identify NRCs on a large urban road network using journey times calculated using data from automatic number plate recognition cameras. Chen et al. (2016) proposed data-driven methods to determine spatiotemporal extents of NRCs. Luan et al. (2021) detected non-recurrent congestions using statistical methods on traffic speed data.

Although probe vehicle data and floating car data (FCD) are used interchangeably, FCD commonly describes a dataset which is collected from various mobile applications, such as probe vehicle fleets or mobile phones, with a broader spatiotemporal coverage due to large number of trajectories processed (Fabritiis et al., 2008; Zhao et al., 2010; Zhu et al., 2009). FCD can provide real-time travel speed data for road segments for even 1-min intervals (Altintasi et al., 2017). FCD is used as a data source in some relevant incident detection studies. Zhu et al. (2009) detected incidents in urban arterials as outliers in spatial and temporal features extracted from FCD collected from 13,000 taxies. Chakraborty (2019) used spatiotemporally denoised thresholds to detect incidents using traffic speed data sampled in 0.5 miles long road segments and 1-minute interval. Luan et al. (2021) used a high resolution, broad coverage, near real-time travel-speed dataset to detect NRCs in an urban road network.

Multiple data sources and data fusion methods are used to improve incident detection performance. Ivan (1997) used neural network-based approaches to detect incidents in urban arterials using data from two different sources: loop inductive detectors and probe vehicles. Thomas (1998) combined data from detector stations and probe vehicles in their decision making-based incident detection method. Social media data has been emerging as an alternative data source to detect traffic incidents. Zhang and He (2016) fused the data collected from loop detectors and Twitter to detect traffic accidents in real-time. Wang et al. (2017) analyzed congestions employing probe vehicle data, social media data and other supplementary information such as social events, road features, point of interests, and weather. Luan et al. (2021) used traffic speed data along with social media data from Weibo to detect NRCs.

CHAPTER 3

TRAFFIC EVENT DETECTION IN SOCIAL MEDIA

In this chapter the method for traffic event detection in social media is presented. In Section 3.1, an introduction of Twitter platform and a description of data structure of tweets are given. In Section 3.2, background on natural language processing (NLP) methods commonly used for information retrieval from text content of social media is given. In Section 3.3,supervised learning methods commonly used for information retrieval from text content is given. Proposed method to detect traffic events in social media is presented in Section 3.4

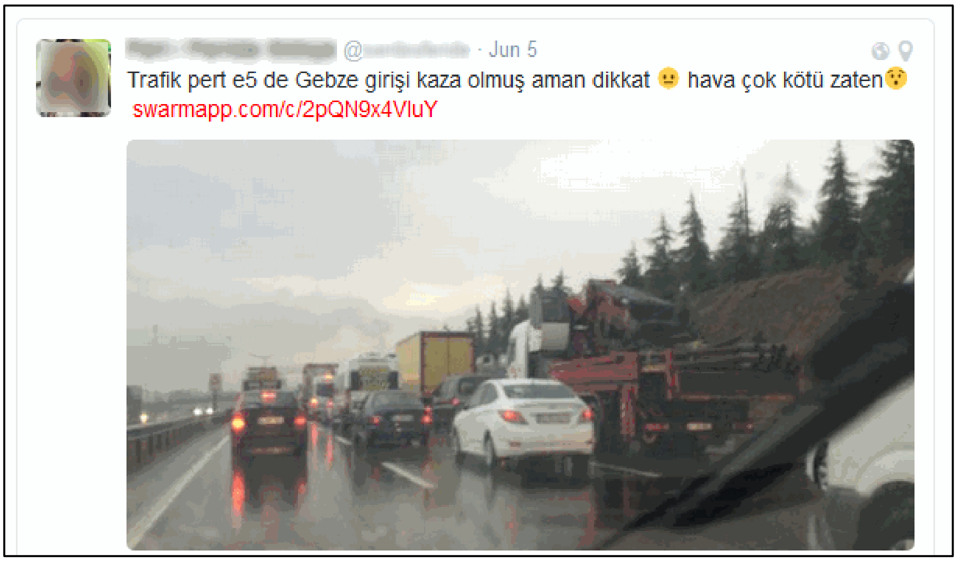
3.1 Twitter Stream

Twitter is a social media service, where users post up to 280-character long messages, called *tweets*. Twitter has around 429 million users sending more than 450 million tweets a day (*Twitter - Statistics & Facts*, 2022; *Twitter Usage Statistics*, 2022) sharing personal experiences, news, and happenings, including traffic events (Figure 3.1).

Twitter provides two ways to access its public data. Twitter Search API serves data through keyword-based search queries against the recent stream. Whereas Twitter Steaming API provides access to live streaming public Twitter data.

Twitter Search API is a representational state transfer (REST) based service, which provides methods to query the recent or popular tweets. Query operators allow search of terms and Twitter features, such as hashtags and mentions. Some additional parameters are provided to filter results by meta fields of tweets, such as geolocation, and language. Geolocation filter consist of *latitude*, *longitude* and *radius* parameters. A rate limit is applied per-user basis on search queries using Twitter Search API. Data structure of a tweet returned by Twitter Search API is given in Appendix A.

Twitter Steaming API provides access to live streaming Twitter data through a persistent Hypertext Transfer Protocol (HTTP) connection with a low latency. Twitter serves its public data through Streaming API in three access levels. *Spritzer*, which is the only freely accessible level, serves roughly 1% of all public tweets, while *gardenhose* level serves roughly 10% of all publicly available tweets. *Firehose* level serves almost all publicly available Twitter data (Boyd and Crawford, 2012). Streaming API can be queried with additional parameters to filter language, users, terms, locations and etc. Unlike the location filter used in Search API, the location query is run only against the tweets with a geotag.



[redacted] @[redacted] · Jun 5

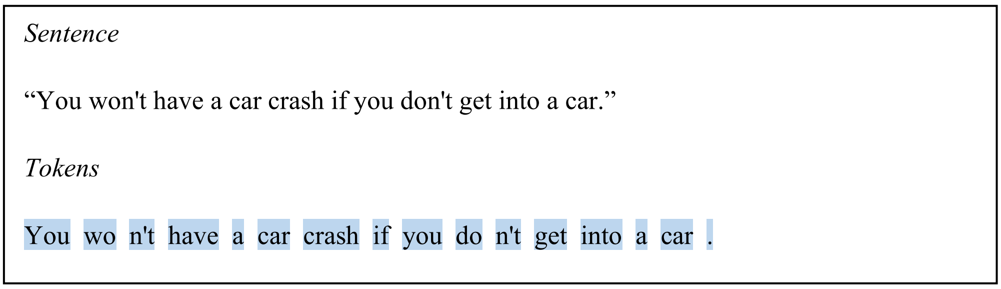
Trafik pert e5 de Gebze girişi kaza olmuş aman dikkat 😐 hava çok kötü zaten😯 swarmapp.com/c/2pQN9x4VluY

Figure 3.1 An incident reporting tweet as it appears on Twitter

3.2 Natural Language Processing Methods Used for Information Retrieval

3.2.1 Tokenization

Tokenization is the process of segmenting texts into its meaningful semantics units, such as words, numbers, punctuation, and other elements. It is a prerequisite step for the methods involved in fields such as natural language processing and information retrieval. Most simple approach is the white space tokenization method, which splits the words by its white spaces. However, whitespaces are not used as token boundaries in some languages such as Chinese and Japanese, in which tokenization is an ambiguous process which require utilization of machine learning methods. Exceptions also exist in the languages where white spaces can be used as token boundaries. Figure 3.2 presents a sentence in English and its tokens which are generated by Stanford NLP library. In this example, "Don't" is tokenized as "do" and "not", which would be kept together with a simple white space tokenizer or would be tokenized as "don" and "t", if punctuation is regarded as token boundaries. In Turkish, tokenization is commonly performed using white spaces and punctuations. Turkish language has a complex morphology. Depending on the problem, morphological features of the words might be treated as individual tokens. In this study proposed methodology includes a morphological analysis task, which is used to extract stem and morphological features of words to treat them as individual tokens.



Sentence

“You won't have a car crash if you don't get into a car.”

Tokens

You wo n't have a car crash if you do n't get into a car .

Figure 3.2 A sentence in English, tokenized by Stanford NLP library

3.2.2 Stemming/Lemmatization

In documents, words are used in various forms to express grammatical features. In English "drink", "drank", "drunk" represent infinitive, past simple and past participle forms of the same verb "drink". In Turkish, “git”, “gidiyorum”, “gittim” are imperative, present continuous and simple past forms of the verb “gitmek”, respectively. The task of stemming and lemmatization are used to reduce the inflected forms into stems or lemmas. While definition of stem might differ among different languages, by its most simple definition, it is the unchanged part among all forms of a word. Lemma is the dictionary form of words. A table comparing reduced forms of words to stems and lemmas can be seen in Table 3.1.

Table 3.1 Examples for stems and lemmas in English and in Turkish

Word	**Stem**	**Lemma**
English		
is	is	be
ran	ran	run
situation	situat	situation
analyzing	analyz	analyze
Turkish		
yönünde (in the direction)	yön (direction)	yön (direction)
hızlıydı (it was fast)	hız (speed)	hızlı (fast)
kitapçıdan (from bookstore)	kitap (book)	kitapçı (bookstore)

Stemming and lemmatization, can increase the performance of classification in precision and computation time by reducing the words into their stems or lemmas. Relevance of grammatical expressions, which are reduced by these processes, should be evaluated according to the characteristics of the problem.

3.2.3 Morphological Analysis

Morphological analysis (MA) is one of the core sub-tasks of natural language processing. Analysis extracts the morphological features and their boundaries from a word. It is important for morphologically complex languages where one word can contain multiple linguistic information which would be represented with several words in other languages (Coltekin, 2010). Morphological analysis can be employed in several natural language processing tasks, such as stemming, part-of-speech tagging, etc.

Methods used for morphological analysis depend on the characteristics of the language. Rule based methods, such as finite state transducers (Coltekin, 2010) and supervised learning methods, such as Hidden Markov Models (Takeuchi and Yuji, 1995) are employed in the task.

Turkish is an agglutinative language with a complex morphology. Available research on Turkish language uses rule-based methods to perform morphological analysis. Oflazer (1990) implemented a two-level morphological description on the PC-KIMMO environment. Another rule-based approach has been implemented using finite state transducers, which is available as an open-source toolset called TRMorph (Coltekin, 2014).

3.2.4 Stop Word Filtering

Stop words are described as the words which have little or no value in common natural language processing tasks. Selection of stop words depends on the language and the design of a task. Stop word filtering can significantly reduce the dimensionality of the features, improve the performance of tasks, such as text classification.

3.3 Supervised Learning Methods used for Information Retrieval

3.3.1 Named Entity Recognition

Named entity recognition (NER) is the task of detecting word or phrases which identify an entity, such as people, locations, companies, on a text. It is a commonly used sub-task in Information Retrieval (IR) and Natural Language Processing (NLP) applications, to retrieve entities from natural language documents. Examples of named entities identifying a company name and locations in a news headline are underlined below.

__Google__ on Wednesday inaugurated a free city-wide Wi-Fi system
in its home town of __Mountain View__, __California__

Rau (1991) presented one of the first works in the field, which proposed heuristic and rule-based methods to extract company names from text. The concept of *named entity* is first defined for the Sixth Message Understanding Conference (MUC-6), as the recognition of information units describing people, organization names, location names and numerical expressions. Task was identified as an essential part in Information Extraction (Nadeau, 2007). Initial rule-based systems are followed by supervised learning-based methods (Nadeau, 2007). Hidden Markov models (Bikel et al., 1997; Zhou and Su, 2002), Maximum Entropy Models (Borthwick, 1999; Chieu and Ng, 2002), Support Vector Machines (Isozaki and Kazawa, 2002), and Conditional Random Fields (CRF) (McCallum and Li, 2003) are applied to the problem.

There have been several studies proposing methods for NER tasks in Turkish texts. Cucerzan and Yarowsky (1997) proposed a language-independent bootstrapping algorithm for NER tasks, method was tested in several languages including Turkish. Tür et al. (2003) used HMM on n-gram language models for named entity extraction in Turkish texts. Bayraktar and Temizel (2008) used local grammar-based approach to extract person names from Turkish financial news texts. Küçük and Yazıcı (2009) presented a rule-based system for NER on Turkish news texts, using lexical resources which include a dictionary of person names in Turkish, well-known political people, locations, and organizations in Turkey and in the world. Supervised learning methods are used in named entity recognition in Turkish texts. Özkaya and Diri (2011) used CRF to extract person, location, and organization names in Turkish informal texts, such as emails. Some research focused on involving morphological features of Turkish language on NER tasks. Tatar and Cicekli (2011), proposed an automatic learning method to identify named entities in Turkish texts, and improved NER performance by using morphological features. Yeniterzi (2011), analyzed the effect of morphology for NER in Turkish, by involving morphological features as separate tokens. Şeker and Eryiğit (2012) included morphological features of Turkish words in CRF model, to detect person, location and organization entities in news texts, using basic and generative gazetteers. Küçük et al. (2014) performed comparative NER experiences with an adapted rule-based NER on Turkish corpora, including tweet corpora.

Entity recognition has been used to detect events in Twitter stream. Nguyen used CRF to identify traffic entities in Twitter text. A special tag set, which describe relevant features of a traffic incident used to annotate the learning data (Nguyen et al., 2016).

Conditional Random Fields, introduced as a sequence modeling framework, perform better than HMM and Maximum entropy Markov models (MEMM) for common cases in practice (Lafferty et al., 2001). It is commonly used for labeling purposes on sequential data, such as biological sequences, text documents and images. It is used in common natural language processing sub-tasks such as part-of-speech tagging, named entity recognition (McCallum and Li, 2003). CRF are used in named entity recognition in twitter text (Nguyen et al., 2016; Ritter et al., 2011).

3.3.2 Classification

Classification is a statistical method used to identify the classes in which an observation belongs. It is used in various applications, such as remote sensing, medical imaging, speech recognition, sentiment analysis and text classification.

Machine learning (ML) methods are frequently employed in classification (Yang, 1999). Choice of ML method depends on the characteristics of the classification problem.

ML methods differ in the way they address problems. The way they learn depends on the characteristics of the available data. ML methods which are based on supervised learning require a labelled input data set, commonly defined as "training data" (Mitchell, 1997). Supervised learning methods train and optimize their model with the training data in order to achieve the desired level of accuracy. Unsupervised learning methods are applied when there is no labelled dataset available. These methods involve algorithms to derive rules to organize data. Semi-supervised learning employs a set of labelled data along with the unlabeled samples with some features of supervised and unsupervised learning.

ML methods can also be grouped into different categories in terms of their function (Jiang et al., 2013). Generative methods try to learn the model which forms the data using assumptions, such as distribution functions. Bayesian algorithms are generative methods which apply Bayes theorem for classification problems. Discriminative methods try to create a model based on the observation data. Linear classifiers, decision tree algorithms are examples for discriminative methods. Instance based algorithms do not maintain explicit classes, creates hypotheses as it compares new instances with the ones seen in training. There are numerous types of ML algorithms to address various types of problems, which are not covered here.

3.3.3 Text Classification

Text classification is the task of assigning texts to categories or classes. Spam filtering, document organization, sentiment analysis, language identification, genre classification, news filtering etc. are some applications performed using text classification. Commonly used methods for text classification are Decision Trees, SVM, Neural Network Classifiers, Bayesian Classifiers (Aggarwal and Zhai, 2012).

3.3.4 Decision Trees

A decision tree is a graph to model possible outcomes of each decision as a hierarchical tree structure. In machine learning, decision trees are used to classify data (Quinlan, 1986).

A decision tree splits data items into sub-nodes according to a splitting condition determined for each node. Splitting condition is set to attain a level of purity on each split, which is defined by a metric called *impurity*. Splitting is performed recursively until a determined stop condition is reach for each node, thus all data is partitioned into a set of hierarchical nodes which represent data space in corresponding splits. Each node is labelled with majority class label. A sample is assigned to the most likely partition for the purposes of classification (Aggarwal and Zhai, 2012).

In text data classification, splitting conditions for the nodes are typically determined by regarding the terms in a document. A splitting condition on a node can be set depending on the existence of a term in the document (Aggarwal and Zhai, 2012).

Construction of a decision tree from observations, which is called a *decision tree learning*, performed using various algorithms. ID3 (Quinlan, 1986) and C4.5 (Quinlan, 2014) are among the most well-known decision tree learning algorithms (Li and Jain, 1998).

3.3.5 Support Vector Machines

Support vector machine (SVM) is a supervised algorithm for pattern classification (Vapnik, 1999). SVM classifiers use the decision boundaries on a hyperplane to identify the class of an input vector. Decision boundary is constructed based on statistical learning theory using structural risk minimization principle. SVM has been proposed as an effective method in text classification, albeit with an increased training cost compared to other commonly used classification methods, such as decision trees or naïve bayes-based classifiers (Colas and Brazdil, 2006; Liu et al., 2010). SVM has been used in various event detection studies on Twitter (D'Andrea et al., 2015; Sakaki et al., 2012; Schulz, Ristoski, et al., 2013).

3.3.6 Naïve Bayes Classifier

A Naïve Bayes classifier is a probabilistic supervised learning model for classification which is based on the Bayes theorem. Model assumes that the probability of each feature belonging to a class is independent of the others. Naïve Bayes classifiers are effective in text classification (Lewis, 1998; Manning, 2008; McCallum and Nigam, 1998; Zhang and Li, 2008). Multinomal model, based on integer word counts, and Bernoulli model, consisting of binary word features, are commonly employed in text classification tasks (McCallum and Nigam, 1998). Naïve Bayes classifiers are also used in event detection in social media (Agarwal et al., 2012; Becker and Gravano, 2011; Schulz et al., 2015).

3.3.7 Validation of Supervised Learning Methods

Evaluation of the performance of a supervised model is a crucial part of the modeling process. In a binary classification model, there are two possible outcomes: positive or negative. In a model, which classifies tweets according to whether they report a traffic accident or not, there are two classes. Positive class consists of the tweets reporting a traffic incident, while negative class consists of the tweets which does not report a traffic accident. To predict if a tweet reports a traffic accident, a model is trained using sample data and performance of the model is evaluated. The positive samples which are predicted correctly by the model are called true positives (TP) while the correctly predicted negative samples are called true negatives (TN). The negative samples which are predicted incorrectly as positive are called false positives (FP) and positive samples which are predicted incorrectly as negative are called false negatives (FN). These core metrics of binary classification performance are presented in a table called confusion matrix (Table 3.2a). The evaluation metrics commonly used for binary classification are accuracy, precision, recall and F1 Score (Table 3.2b).

Table 3.2 Metrics used in classification performance evaluation a) Confusion matrix b) Performance evaluation metrics

a)		**Predicted**	
		Positive	**Negative**
Actual	**Positive**	True Positive (TP)	False Negative (FN)
	Negative	False Positive (FP)	True Negative (TN)
b)			
Accuracy		$\frac{TP + TN}{TP + FP + FN + TN}$	
Precision		$\frac{TP}{FP + TP}$	
Recall		$\frac{TP}{FN + TP}$	

3.3.7.1 Cross Validation

In machine learning, prediction models learn the parameters from a dataset, which is commonly referred as a "training dataset". Testing the performance of a prediction model using the training dataset would fail to assess the performance of the prediction model. Repeating the samples used in learning stage in performance tests would result in high scores, a phenomenon called overfitting. To avoid overfitting, separate datasets are used to train and test the prediction model.

3.3.7.2 k-fold Cross Validation

In k-fold cross validation, all the samples are divided into k equal sized groups, which is referred as folds. The prediction model is learned using k-1 folds, while the sample in the remaining fold is reserved as test-dataset (Figure 3.3). The training and testing process is repeated k times until all k folds are used as test-datasets. The results from tests are summed up to get overall performance results.

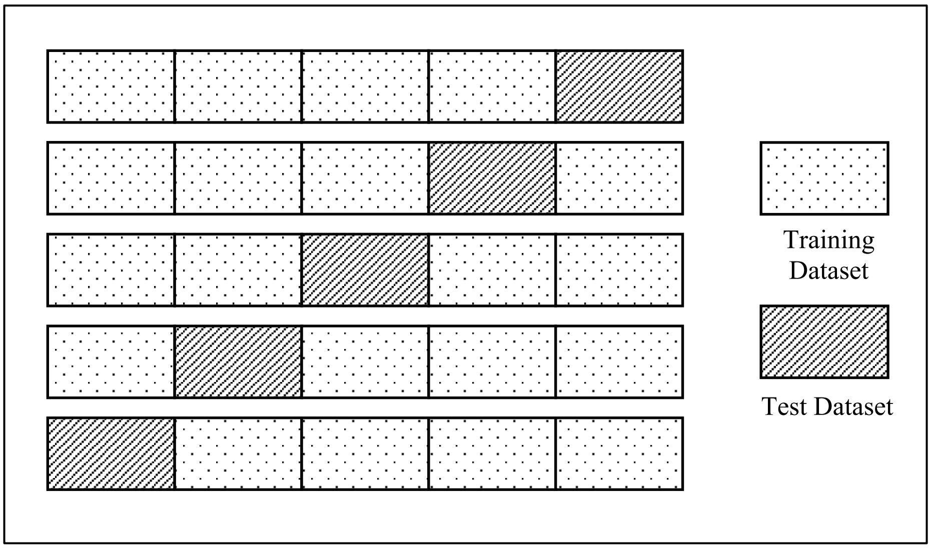


Figure 3.3 Segmentation of data set in k-fold cross validation tests

3.4 The Method

In this section, the proposed method for detecting posts related with traffic incidents and conditions from Twitter stream is presented. As given in Figure 3.4, the method consists of five main steps:

1. Collection of tweets using predetermined keywords
2. Preprocessing of text content to extract tokens
3. Morphological analysis of tokens

4. Recognition of traffic related named entities, a customized set of named entities that are commonly used in traffic related incident or condition descriptions
5. Classification of tweets to identify tweets which report an incident or condition which affect traffic flow.

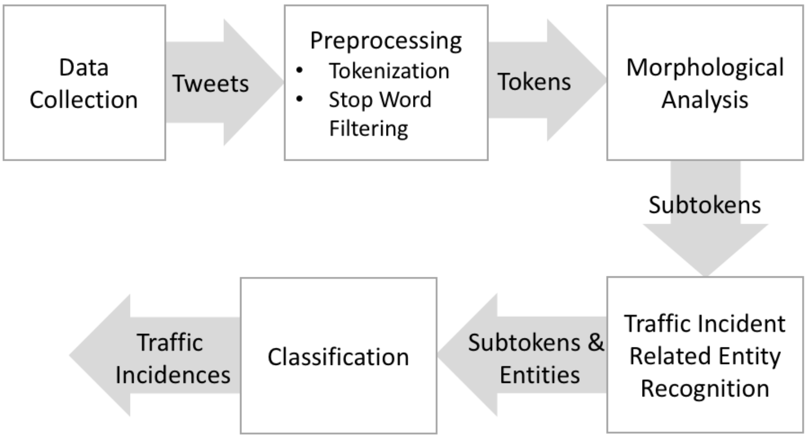


Figure 3.4 Basic Steps and the Flow of the Proposed Method

3.4.1 Data Collection

The proposed method aims to retrieve information from short social media texts. Short message services, such as Twitter, provides necessary API's[1] to access their publicly available data. During the manual scan of tweets, a set of search terms are determined which are directly related to traffic flow or to incidents that might have an impact on traffic flow. Twitter Search API performs simple term search, ignoring morphological features of the terms. Due to agglutinative structure of Turkish language, inflection forms of the terms are also included in search query strings.

[1] https://developer.twitter.com

Additionally, due to informal nature of tweets, Turkish specific letters are frequently replaced by their corresponding Latin versions. Alternative replacements for such letters, which are not handled by API, are also included in search terms. Search is performed using “OR” query strings to cover all possible related posts. Therefore, collected data consist of all the posts that include any of the keywords provided, hence it is not guaranteed that they are all traffic flow related posts.

3.4.2 Preprocessing

Preprocessing step includes the cleaning-up, tokenization and segmentation tasks, which are required before NER and classification steps. This step starts with filtering re-tweets, which are reposts of original tweets with the same content. Tweet texts contain Twitter-specific features, such as mentions, hashtags, “retweet” tags and links. Due to specific formatting of these features, a customized tokenizer is used for determining Twitter-specific features and tokenization. Preprocessing step is completed with stop-word filtering. Stop words are mostly conjunctions, prepositions or postpositions, which act to connect or support sentences or other words rather than having a word-sense of its own. A limited set of Turkish stop-words including "ve" (and), "için" (for), "ile" (with), "gibi" (as, like) is used.

3.4.3 Morphological Analysis

Agglutinative structure and complex morphology of Turkish imposes a high rate of inflection per word, resulting with high dimensionality in vector space of terms. Redundant dimensions can be reduced through stemming. However, stemming in Turkish words is a process including high ambiguity, requiring morphological analysis of the tokens. Therefore, within our method, each token is analyzed morphologically in order to extract roots and a set of inflection groups using the morphological analysis tool TRMorph (Coltekin, 2014). Since tweets are short texts and they provide a limited context information for morphological disambiguation, morphological analysis of tokens often results in ambiguous results. This Ambiguity is resolved using the zero-context disambiguation tool provided within TRMorph. After disambiguation, segmentation is performed over token according to the generated stems and inflection groups. Each segment, which is referred to as a *subtoken* in this study, consists of a surface and the morphological tags assigned by the analyzer. Morphological tags are part of speech tags for stems and inflection tags for inflection groups.

3.4.4 Traffic Event Related Entity Recognition

Terms that are used for describing an event or a condition related to traffic network may also be used for indicating other meanings in entirely different contexts. For instance, the Turkish word "kaza" means "accident", which is relevant in traffic context, and also "town". Furthermore, the same term can be used to indicate any type of accident taking place in different contexts, such as a "kitchen accident". Ambiguity of senses and contexts of these terms might affect the performance of classification tasks.

This study aims to focus on tweets that report traffic related incidents or conditions on road networks. Due to characteristics of such incidents, locations are reported with respect to the street on which the event took place. Unlike official address definitions, which include either house number or distance reference to define a reference location on a street, tweet reports commonly use landmarks, directions, or conjunction points to denote a location on a street. These cause another potential ambiguity among the location references considering their functions in the address definition. For example, a district name in an address definition can indicate the location of an event, alternatively it can be used for defining a particular direction of a street or a conjunction point on a street, on which the incident took place. Consider the following two tweets including the location reference *Eskişehir Yolu* (Eskişehir Road) with two different referencing functions:

Eskişehir yolu describing the district where event took place:

> *Eskişehir yolu tarım bakanlığından mesa kavşağına kadar yoğun akıcı*
> Eng: Heavy but stable flow on Eskişehir Road from ministry of agriculture to mesa intersection)

Eskişehir yolu describing the direction of a street on which event took place:

> *#YolDurumu Anadolu Bulvarı- Marşandiz sonrası Eskişehir yolu yönünde araç arızası!*
> Eng: #RoadCondition Vehicle break-down after Marşandiz on Anadolu Boulevard, in the direction of Eskişehir Road

In order to address the problem of referencing ambiguity, a traffic event related entity recognition model based on Conditional Random Fields (CRF) is developed. To this aim, a customized set of traffic related named entities are defined. These named entities correspond to common definitions used for reporting traffic related incidents or conditions. In Table 3.3, these named entities are grouped under the titles of *traffic*, *incidents*, *incident attributes*, and *location tag*s. A web-based annotation tool has been developed for annotating *subtokens* with part of speech or morphological tags (Table 3.4) and the traffic related entities to create the training data set to be used for constructing the CRF-based model.

As an example, consider the following tweet that reports an accident:

> *Keçiören Fatih köprüsü Keçiören yönünde 3 aracın karıştığı zincirleme kaza var. Köprü tıkalı.*
>
> Eng: A three-vehicle pile-up accident in Keçiören direction on Fatih overpass at Keçiören.

Table 3.3 Traffic event related named entities

Named Entity	Explanation
Traffic	
Flow	Flow Condition
Direction	Terms describing a flow direction on a street
DirectionIndicator	Morphemes, words or phrases indicating directionality
Connection	Terms describing connections on s street
ConnectionIndicator	Morphemes, words or phrases indicating directionality
Incidents	
Accident	Traffic accidents are tagged separately from other incident types
Maintenance	A road maintenance work
ExternalEvent	Non-traffic related events, such as concerts, sport events
RoadCondition	Conditions effecting the traffic flow, including weather conditions
Misc	Other incidents affecting traffic flow
Incidents Attributes	
Damage	Terms describing the damage of an incident
Lane	Lanes which are affected by reported incidents or conditions
Time	Time of incidents or conditions
Vehicle	Vehicles involved in incidents
People	People involved in incidents
RoadFeatures	Road features involved in incidents or conditions
Location Tags	
Street	Street names
Region	Toponyms describing a region or administrative unit
Landmark	Location references describing a landmark or a well-known location
LocationIndicator	Morphemes assisting a toponym

Table 3.4 Part of speech and morphological tags used in TEER model

Tag	Type	Definition
abl	Morpheme	Ablative case ("-den" suffix)
Adj	POS	Adjective
Adv	POS	Adverb
Cnj	POS	Conjunction
dat	Morpheme	Locative case ("-e" suffix)
Det	POS	Determiner
Exist	POS	"var" (exists) or "yok" (not exists)
loc	Morpheme	Locative case ("-de" suffix)
N	POS	Noun
Not	POS	"değil" (not)
Num	POS	Number
Postp	POS	Postposition
Prn	POS	Pronoun
Q	POS	Question Particle
V	POS	Verb

The traffic event related named entities recognized by the model in this tweet are as follows:

- *Location:* Keçiören (a district)
- *Location:* Fatih köprüsü (Fatih bridge)
- *Direction:* Keçiören
- *DirectionIndicator:* yönünde (in the direction of)
- *Damage*: zincirleme (pile-up)
- *Incident*: kaza (accident)

Recognized entities by the model are used as features in classification step in order to improve the accuracy performance. Entities retrieved using the developed model can also be employed in geolocation tasks.

3.4.5 Traffic Event Related Post Classification

Keyword based search by using traffic related keywords is not an accurate method to determine traffic event related tweets. Search API retrieves postings including any of the search keywords, and hence the result set may contain irrelevant tweets with unrelated contexts. Therefore, a classification model, using a manually annotated set of tweets as the training data set, is constructed. Each tweet is represented a feature vector such that each feature is either a stemmed word from the tweet content, or a traffic event related named entity extracted from the tweet.

A representation model is proposed in order to classify tweets by relevancy using machine learning methods. Stems retrieved in morphological analysis are used to represent tweets in a bag-of-words model. Terms are weighted with their TF-IDF scores in the model. The entities detected in Traffic Event Related Entity Recognition step are also included in feature representation to improve classification performance.

Support Vector Machines (SVM), Naïve Bayes (NB) and Decision Trees based classifiers, which are commonly applied in incident detection problems, are used to classify relevant tweets. SVM and Decision Trees are discriminative classifiers while Naive Bayes classifier is based on a generative mode (Boser et al., 1992; Quinlan, 1986). A linear SVM model is trained using Sequential Minimal Optimization algorithm. For the Naïve Bayes-based classifier, a univariate discrete distribution is used. Minimum allowed probability in the frequency tables is set as $1e^{-10}$. No regularization is used. Decision Tree model is modeled by using C4.5 algorithm (Quinlan, 2014).

CHAPTER 4

TRAFFIC EVENT GEOCODER

In this chapter, a geocoding method which is customized to localize traffic event tweets is presented. Given tweets that report traffic-related incidents or conditions which occurred on a road network, in this work, a method is proposed to generate coordinates of the events or conditions mentioned in tweets. The proposed geocoding method consists of two steps. The first one includes a customized named entity recognition model called Traffic Event-related Entity Recognizer (TEER). The second step includes a rule-based road geocoder, which makes use of the customized named entities detected by the TEER module. The overall architecture of the proposed solution is given in Figure 4.1.

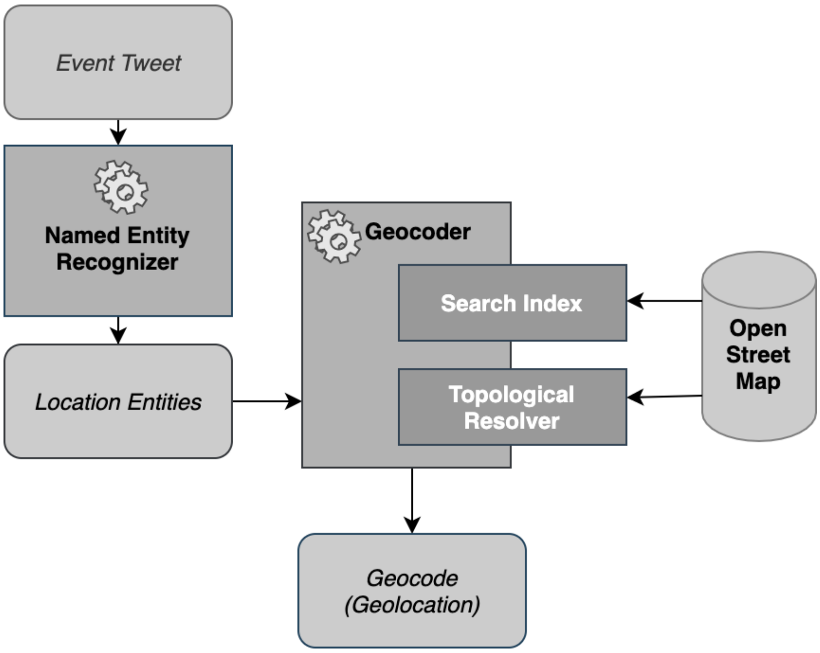


Figure 4.1 Traffic Event Geocoder

In the rest of this chapter, the steps of the solution given as modules in Figure 4.1 is described.

4.1 TEER Module

TEER is a Conditional Random Fields (CRF) based custom named entity recognition module, which is developed to recognize a custom set of traffic event-related named entity tags (Section 3.4.4). Traffic event-related tags include terms that describe the attributes of an event, including its location. Unlike formal address definitions, which include either house number or distance reference to define a location on a street, event tweets commonly use landmarks, directions, or conjunction points to define locations on roads. This might arise ambiguity among the location terms regarding their function in the location definition, such as a district name in an address definition can define the location of an event or can indicate a particular flow direction of a street on which the event took place. TEER also aims to resolve location terms into their granular functions in the location definition. The set of traffic event related named entity types recognized by TEER are as follows: The set of traffic event related named entity types recognized by TEER are as follows:

- Road Term: Road names, such as *Eskişehir Yolu* (eng. Eskişehir Road, a street).
- Location Term: Location references describing a landmark, district, point of interest or a well-known location, such as *Armada* (a mall), *Ümitköy Köprüsü* (eng. Ümitköy overpass), *Bağlıca* (a district).
- Direction Term: Terms describing a flow direction on a street, such as *merkez* (eng. inbound), *Kızılay yönü* (eng. Kızılay direction).
- Connection Term: Terms describing connections on a street.
- Indicators: Auxiliary terms or suffixes assisting the aforementioned tags, which are annotated separately.

4.2 Geocoder Database

Geocoding step involves methods to locate detected location entities in TEER step on the road network. The proposed geocoding method includes a knowledgebase in order to perform geocoding tasks. To this aim, OpenStreetMap (OSM)[2] data is employed. OSM data is organized as elements and their tags. For instance, stretches of roads are stored in *way* elements. The name, class, and speed information of the streets are stored as the tags of *way* elements.

Geocoder extracts and converts OSM elements into its internal data object types, landmarks, roads, network topology, and regions, which are described as follows:

- *Landmark object* defines a coordinate that represents a point of interest, such as amenities, public buildings, schools, or other named geographic entities. It is used for geolocating toponyms defining a particular location on a road network.
- *Road object* defines all stretches that belong to a distinct road in a network. Geocoder geocodes locations on the road objects.
- *Network topology* is a model representing road network as a graph structure of nodes and links to perform analysis based on the connectivity of streets' stretches.
- *Region objects* define an area that can represent a neighborhood, district, or any other administrative area. Since many of the event tweets refer to a district name as a destination describing a flow direction, they are frequently detected as a *Direction Term* and employed to determine the flow direction of a street on which an incident took place. They are also used to assist street or landmark matching when detected as a *Location Term*.

[2] https://www.openstreetmap.org

Landmark objects are retrieved from OSM *node* and *way* elements with tags *amenity*, *shop*, *office*, *public_transport*, *station*, *highway*, and *building*. *Way* elements are represented by their centroids.

OSM data include administrative boundaries in hierarchical levels. Levels start from province boundaries and reaches down until local neighborhood-level administrative boundaries. However, official administrative boundaries fall short in defining the exact regions for place names which are used in informal location descriptions. OSM is a rich dataset including landmarks and points of interest, which could be expanded to extract further levels of information. A region detection method is proposed, which determines the border of *region objects* through the geometric distribution of the elements that include the region name in the name of the element. For instance, *Sincan Kapalı Spor Salonu* (eng. Sincan Indoor Sports Hall), *Opet Sincan* (a gas station in Sincan), etc., represent Sincan region. Similarly, *Ümitköy İtfaiyesi* (eng. Ümitköy Fire Station), *Pet Hospital Ümitköy* (a veterinary hospital in Ümitköy) and other elements having Ümitköy in their names represent Ümitköy region. Detected region names are further checked against a list of reference regions, populated using elements such as administrative units and public transport station names, to filter out false detections.

Road and *network topology objects* are extracted from OSM *way* elements with *highway* tag. In this study, only major roads and arterials are used for geocoding the events. Values for *highway* tag that are used for retrieving the roads are "motorway", "trunk", "primary" and "secondary". Definitions in *addr:street* tag of nearby landmarks are used to extract informal names for the roads. A *network topology object* consists of unidirectional polyline links that are extracted from *way* elements through their nodes shared by other elements. *Roads objects* are identified using search based on topological connectivity and name matching. A fuzzy name matching is used for handling the name variations of the same road.

4.2.1 Use of the search index

The names of the extracted landmarks, streets, and regions are indexed using Apache Lucene[3], a software library that provide indexing and search features, as well as spellchecking and tokenization capabilities. Location expressions in tweets are informal, often represented in the shortest possible form. Typographical errors such as misspellings, transpositions, omissions, splitting errors, concatenation errors, wrong key errors are also common. The n-gram analyzer of Apache Lucene is customized to fix such errors.

Results solely ranked by text similarity score do not provide the best candidates for location term searches. To address this, in landmark queries, existence of spatial clustering of search results is checked using Nearest Neighbor Analysis and scores of matches in clusters are boosted to favor over the matches scattered spatially. Text-similarity scores accompanied by a spatial clustering index based on nearest neighbor distances improved the search results.

4.3 Geocoding

In this work, a rule-based geocoder is proposed to generate the coordinates for location terms detected by TEER on a road network. Input for the geocoder are traffic event-related entities, *Location Term*, *Road Term*, *Connection Term* and *Direction Term*. All possible mappings of *Locations Terms* and *Connection Terms* on the road network are generated. These mappings are further refined by the *topological resolver*, and they are finally geocoded into a location. The flowchart of the rule-based geocoder is presented in Figure 4.2.

[3] https://lucene.apache.org

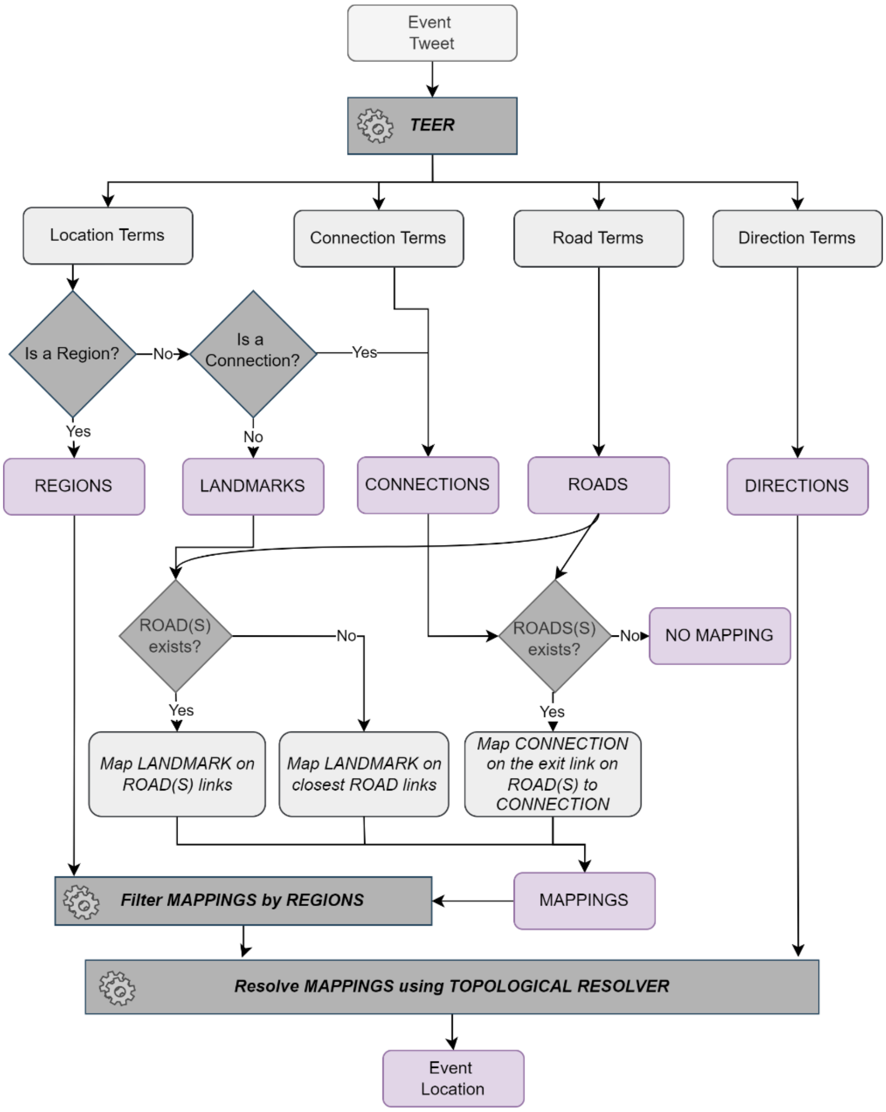


Figure 4.2 Geocoding Flow

4.3.1 Road Network Mapping

In road network mapping step, set of rules given in Figure 4.2 are used to map the location entities into candidate objects on the network. The process starts with a refinement of location entities using data extracted from OSM. Due to limitations of TEER model in distinguishing whether a place name is for a landmark, neighborhood, or a district, all such terms are tagged as *Location Terms*. *Location*

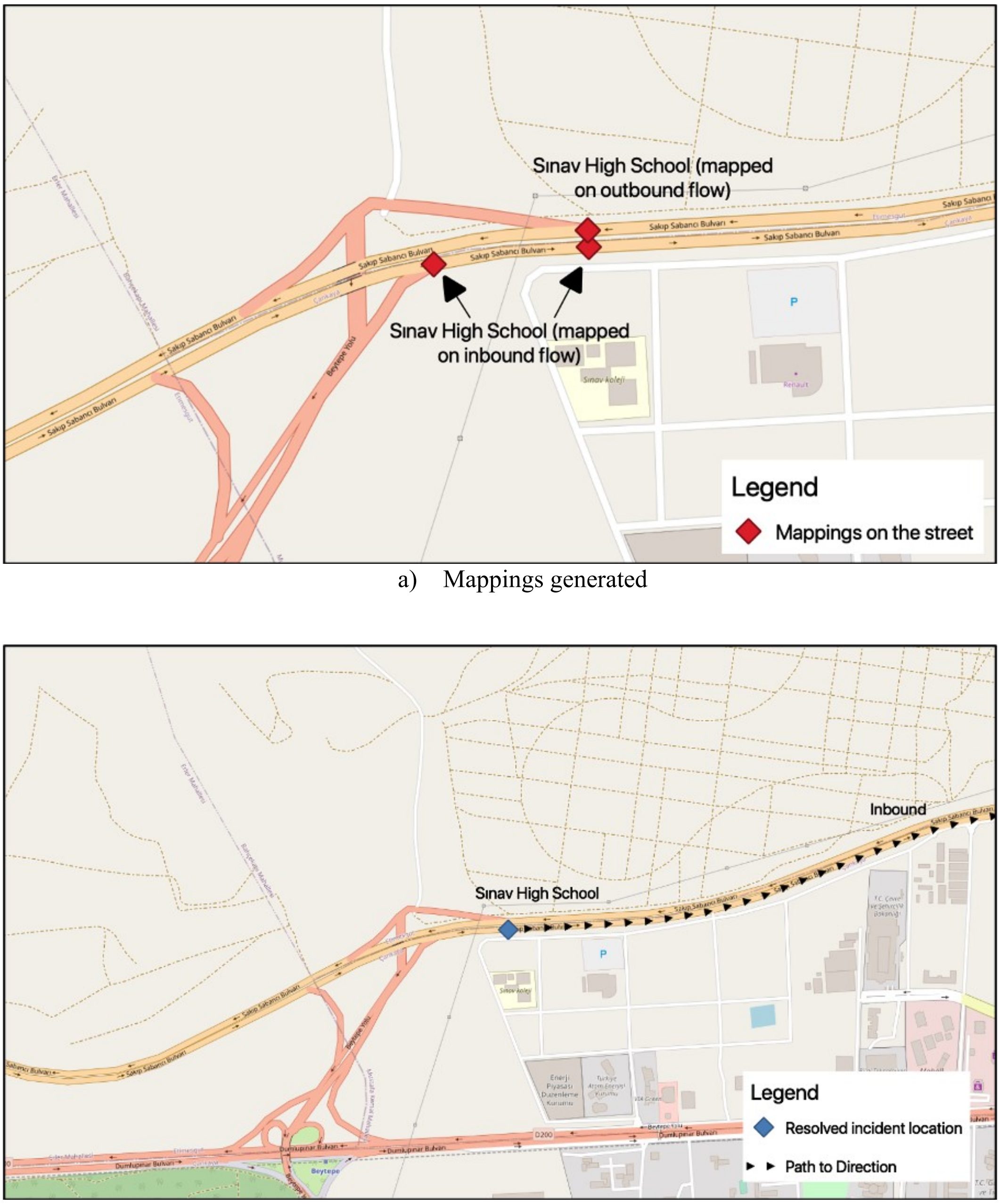
Terms are refined using a dictionary-based approach. *Location Terms* defining a region is identified by matching against the *Region Objects* in the geocoder database. One common informal use in location definitions is to name intersections by the name of the district they connect, rather than its official name, such as *Hacettepe Kavşağı* (eng. Hacettepe Intersection) connecting Eskisehir Road to Hacettepe University. Such informal names are mostly missing in OSM. To alleviate this problem, intersection names are identified by matching *Location Terms* against a dictionary of intersection terms such as *kavşağı* (eng. intersection), *üst geçidi* (eng. overpass). Detected intersection terms are treated as *Connection Terms* for a correct localization.

Geocoder maps each *Location Term* on the road network using *Landmark* and *Road Objects*. A *Location Term* will be mapped to the detected road, otherwise, they are mapped onto the closest road on the network. Similarly, *Connection Terms* are also mapped to the roads by the proposed geocoder. Unlike *Location Term*, *Connection Term* can indicate a road, a landmark, or a region, therefore they are matched on all the corresponding object types. Mapping is performed by employing an A* path-finding algorithm from the matched *Road Objects* to the matching connection objects in the geocoder database, on the network topology. Connections are mapped on the exit links of the matching streets.

Mapping task may produce multiple candidates for matching landmarks, roads, and links on the roads, including stretches of opposite flow directions. Candidate mappings are resolved by Topological Resolver into a single geocoding result.

4.3.2 Topological Resolver

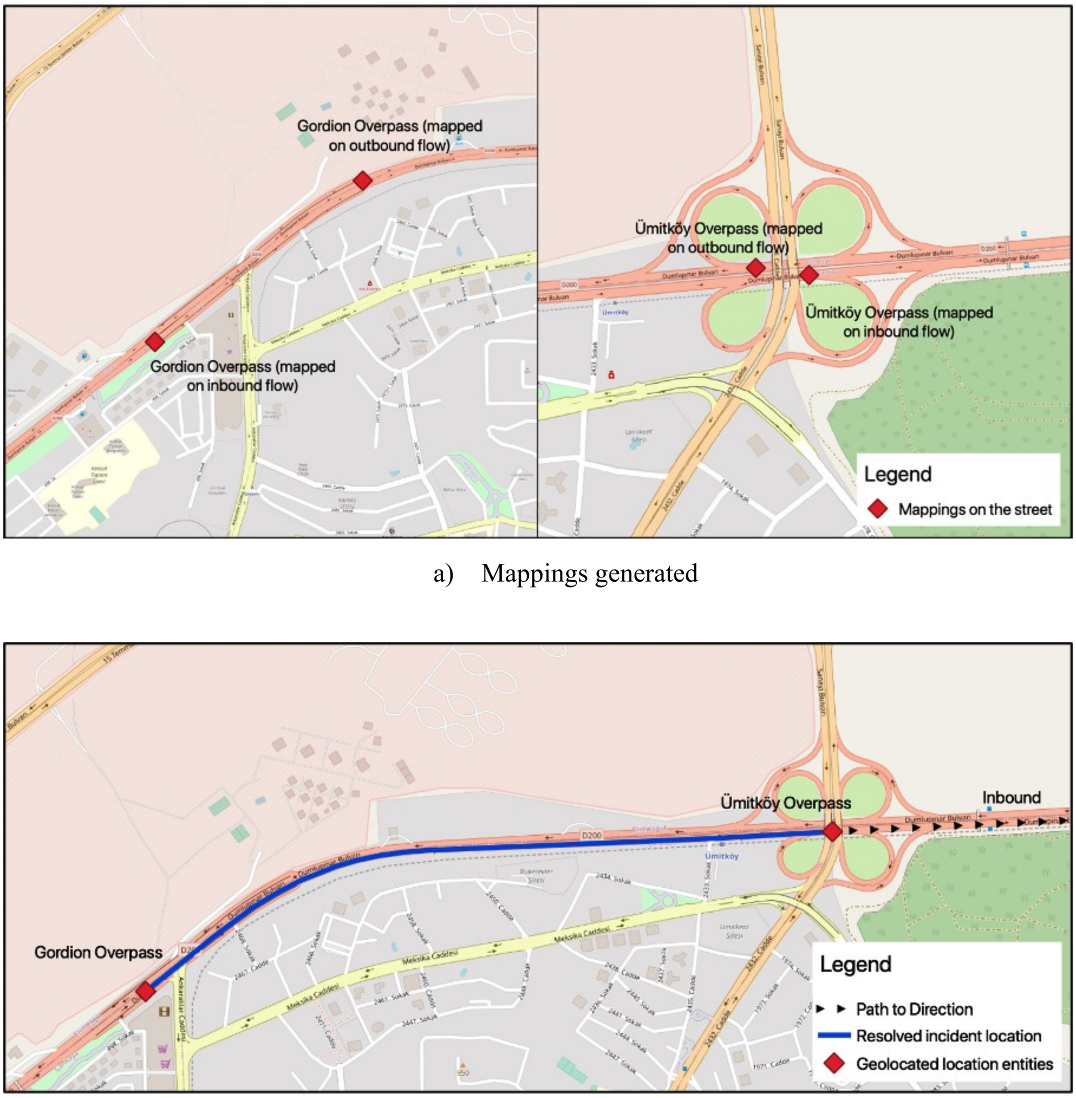
Topology resolver is the last step of geocoding process, which resolves, and maps candidate road mappings found for location entities on the road network. Topological resolver finds the shortest path among the mappings of the location entities. It solves a traveling salesman problem among location entities mapped on a road, involving all possible mappings. If there is only one location entity mapped on a road, the mapping with the closest link distance is accepted as the geocoding result, without further analyzing the connecting paths. An example is given in Figure 4.3. In this figure, due to the existence of *Direction Term*, the flow is marked on the map starting from the closest mapping for the *Location Term*. If there exists more than one location term mapped on roads, the path connecting the location entities is selected as the location of the event. If a *Direction Term* is also detected by TEER, it is appended at the end of points to determine the direction of the flow. An example case is given in Figure 4.4.



a) Mappings generated

b) Geocoded location

Figure 4.3 Mappings and geocoded location for location entity Sınav Koleji (eng. Sınav High School) on street Sabancı Bulvarı (eng. Sabancı Boulevard)


Gordion Overpass (mapped on outbound flow)
Gordion Overpass (mapped on inbound flow)
Ümitköy Overpass (mapped on outbound flow)
Ümitköy Overpass (mapped on inbound flow)
Legend
◆ Mappings on the street
Ümitköy Overpass
Inbound
Gordion Overpass
Legend
▸ ▸ Path to Direction
Resolved incident location
◆ Geolocated location entities

a) Mappings generated

b) Geocoded location

Figure 4.4 Mappings and geocoded location for location entities Gordion Köprüsü (eng. Gordion Overpass) and Ümitköy Köprüsü (eng. Ümitköy Overpass) on road Eskişehir Yolu (eng. Eskişehir Road)

CHAPTER 5

NON-RECURRENT CONGESTION DETECTION IN FCD

In this study traffic incidents are detected using two independent data streams, social media data (SMD) and floating car data (FCD). In previous chapters, methods to detect and geolocate traffic events detected in Twitter are presented. In this section, proposed methods to detect non-recurrent congestions (NRC) from Floating Car Data (FCD) is presented. Method to detect NRCs consist of two phases: 1) anomaly detection and 2) spatiotemporal congestion identification. In anomaly detection step, traffic flow speeds observed on links are compared with the historical data using statistical methods to detect anomalous travel speeds. In spatiotemporal congestion identification step, links with anomalous travel speeds are merged into spatiotemporally continuous clusters, which will define the spatial and temporal extend of an NRC.

5.1 Anomaly Detection

First phase of the proposed NRC detection method aims to detect anomalous drops in flow speeds in FCD, which are assumed to be a sign of a non-recurrent congestion. In this study, FCD frame denotes to a spatiotemporal data analysis window consisting of a link description representing the road stretch and a time-window on which FCD records are sampled to. In this phase an anomaly factor (AF) is calculated for each FCD frame, which will quantify the anomalous speed observed in a segment during an epoch, by comparing observed flow speed with respect to the estimated flow speed. Also, a threshold AF (T_{AF}) which separate normal and anomalous frames will be calculated.

Flow speed data has multiple seasonality patterns in daily, weekly, and annual cycles, demand is also affected from national days, religious holidays, and frequent changes in the road network. New roads opening to service will take off same demand from other roads. All these factors in a dynamically changing road network makes traffic flow speed estimation, thus anomaly detection a challenge. Several statistics, such as standard normal deviate (SND), median absolute deviate (MAD) or inter-quartile deviate (IQD), have been used to recognize unexpected changes in traffic flow data to detect incidents or congestions (Balke et al., 1996; Chakraborty et al., 2019; Dudek et al., 1974; Li et al., 2013; Luan et al., 2021). Statistics are applied on time-series data collected from a detector or a probe vehicle. Time series data can be a sequence of a data collected from a sensor, or a series of data organized in time-of-day day-of-week (TOD-DOW) slots to handle weekly traffic flow patterns.

In this study commonly used methods for anomaly detection on time-series data are evaluated along with a proposed spatiotemporal anomaly detection approach, Road Network-based Estimator (RNE), which estimates current flow speed on a segment combining both segment's historic data and speeds observed in the whole road network.

5.1.1 Statistical Methods

Standard Normal Deviate, Mean Absolute Deviation and Generalized Extreme Studentized Deviate, which are used in anomaly detection methods in traffic networks data, are evaluated as baseline statistical methods. In these evaluations, FCD data is sampled in 5-minute intervals and weekly seasonality is removed by dividing data set into weekly time windows. A sliding analysis-window is used to define the date range of historic data to be used in the analysis. For instance, to analyze a segment in a 6-week analysis-window, a dataset consisting of traffic speeds observed in the segment during a weekly time-window (same time-of-day and day-of-week) over the last 6-weeks is used.

5.1.1.1 Standard Normal Deviate

Standard Normal Deviate (SND) is the difference of a random variate with respect to a normal distribution. SND is a metric used to check whether a value is significantly different from what is expected under normal distribution. SND has been used to detect traffics incidents in several studies (Dudek et al., 1974; Luan et al., 2021). Formula of SND is given in Equation 5.1.

$$SND = \frac{x - \mu}{\sigma} \tag{5.1}$$

where; x = given value, μ = mean of data set, σ = standard deviation of data set.

5.1.1.2 Median Absolute Deviate

Median Absolute Deviate (MAD) is the distance of a value and the median of the univariate dataset. MAD has been a choice of metric due its insensitivity to outliers and robustness in anomaly detection studies (Hochenbaum et al., 2017; Luan et al., 2021). Formula of MAD is given in Equation 5.2.:

$$MAD = median(|X_i - \tilde{X}|) \tag{5.2}$$

where $X_{1,}X_2, \dots X_n$ is the data set, and $\tilde{X} = median(X)$.

5.1.1.3 Generalized Extreme Studentized Deviate

Generalized Extreme Studentized Deviate (GESD) is a statistical test used to detect outliers in a univariate data set which has an approximately normal distribution (Rosner, 1983). GESD is able to detect up to a given number (r) of outliers in a data set. GESD performs r separate tests, resulting in r test statistic result (R_i), shown in Equation 5.3:

$$R_i = \frac{max_i|x_i - \bar{x}|}{\sigma} \tag{5.3}$$

where; $x_1, x_2, \dots, x_n$ is a univariate data set, $\bar{x}$ is the mean and σ is the standard deviation of the data set. In each test the data point maximizing $|x_i - \bar{x}|$ is removed from the data set and next test is run with the remaining data points. For each test a critical value (λ_i) is calculated, as shown in Equation 5.4:

$$\lambda_i = \frac{(n-i)t_{p_i,n-i-1}}{\sqrt{(n-i-1+t^2_{p_i,n-1-1})(n-i+1)}} \quad i = 1,2,\dots,r \tag{5.4}$$

$$p = 1 - \frac{\alpha}{2(n-i+1)} \tag{5.5}$$

where; r is the number of outliers to be tested and $t_{p,v}$ is percentage point from t-distribution, v is degrees of freedom and α is the significance level. Outliers are determined by data points satisfying R_i>λ_i.

5.1.2 Long Short-term Memory Model

A recurrent neural network (RNN) is a type of artificial neural network, where nodes are connected in a way to feed a layer with the output of the previous process along a sequence. RNN's are widely used for applications such as language modelling, machine translation, speech recognition and time series prediction.

Long short-term memory (LSTM) is a recurrent neural network designed to LSTMs were developed to alleviate the vanishing gradient problem which might arise when training particularly long sequences. LSTM is commonly used for applications including road traffic flow prediction and congestion detection, web traffic anomaly detection and intrusion detection.

5.1.3 Road Network-based Estimator

Methods used in anomaly detection studies are mostly based on univariate time-series datasets operating solely on time dimension. FCD is a spatiotemporal data, which provide insight on the road network state along time and over space. In order to utilize network state knowledge in space, a link traffic speed estimator Road Network-based Estimator (RNE) is proposed based on both links historic data and link speeds observed over the road network. As in other models, daily and weekly seasonality is removed from FCD data by dividing data set into weekly time windows. A sliding analysis-window is used to define the date range of historic data to be used in the estimation. Since the inbound and outbound links produce different daily patterns, links are annotated with *inbound* or *outbound* according to their flow direction with respect to the central business district *(Kızılay)* of Ankara. Links which are not generating flow towards or from Kızılay assigned a *none* flow direction. A network state ratio $r_{d,\tau}$ is calculated for flow direction d, and time-window τ, comparing corresponding n-week historic speed averages observed in links with the speeds observed on the same sat of links, as shown in Equation 5.6:

$$r_{d,\tau} = \frac{\bar{x}_{d,\tau}}{\frac{1}{w}\sum_{i=0}^{w} \bar{x}_{d,\tau_i}} \tag{5.6}$$

where; $r_{d,\tau}$ is the network state ratio for flow direction d at time-window τ, $\bar{x}_{d,\tau}$ is the average speed observation of links with flow direction d at time-window τ, w is analysis window size in weeks and τ_i denotes to corresponding time-window τ in week $- i$. Flow speed of a link l with a flow direction of d at time-window τ is estimated by multiplying corresponding n-week historic observations in l with $r_{d,\tau}$. (Equation 5.7)

$$s_l, \tau = r_{d,\tau} \left(\frac{1}{w}\sum_{i=0}^{w} x_{l,\tau_i}\right) \tag{5.7}$$

where x_{l,τ_i} denotes to speed observation of link l at the time-window τ in week $-\ i$.

5.2 Spatiotemporal Congestion Identification

In anomaly detection phase, methods to quantify the unexpectedly low travel speeds over FCD frames are presented and an anomaly threshold is determined using a manually annotated dataset of concurrent and non-recurrent congestions. Detection of anomalous speed drops in FCD frames are not adequate for incident detection in an urban network due to several reasons. Non-recurrent congestions may occur due to external reasons impacting the wider network, such as inclement weather affecting the capacity (Chin et al., 2002) or events such as school terms and national days which breaks the general weekly flow patterns. In order to identify local incidents in non-recurrently congested links, a congestion front detection method based on supervised learning is presented. Proposed classifier detects local incident related congestion fronts in each epoch, and their upstream using a graph search. Total spatiotemporal impact area of an incident related NRC is determined by merging of the related congestion fronts and their upstreams. Method is presented in three phases:

- Detection of congestion fronts
- Detection of upstreams of detected congestion fronts in each epoch. Spatiotemporal extend of congestion in each epoch is denoted as a *Congestion Stretch*
- A rule-based merging of *Congestion Stretches* to detect a local incident related NRC and its impact area

5.2.1 Data Preparation

A road (R_i) is represented as a tuple of (*name*, *segments*), where *name* denotes the name of the road, *segments* represent all road stretches constituting the road. A road segment (S_J) represents a stretch of a road, boundaries of which is determined by intersections in the topology or by data sampling limits. A road segment is represented as a tuple of (*id, direction, roadclass, start, end, from, to*), where *id* is a unique identifier of the segment, *direction* is the flow direction of the segment, such as “inbound”. *Roadclass* is the enumeration of functional road class based on the capacity of the road. *Start* and *end* denote the geographic coordinates of start and end node of the segment, respecting the flow direction, in WGS84 coordinate system. *From* and *to* refers to adjacent segments which are connecting in or out of the corresponding segment. Segments in a section of road network is given in Figure 5.1a. The road network is modelled as a directed graph of $G = (V, E)$, where vertices (V) represent segments and edges (E) represent connecting nodes between the segments. Nodes are constructed using *start* and *end* coordinates of segments, while adjacency relationship is retrieved using connectivity attributes of *from* and *to*.

5.2.2 Congestion Front Detection

Comparative methods using measurements along a road are commonly employed in incident detection studies. California algorithm compares occupancy data obtained from adjacent loop detectors along with downstream occupancy data to detect incidents (Payne and Tignor, 1978). Rapid declines in speed measurements along a road is commonly observed as an indicator of an incident (Li and McDonald, 2005; Sun et al., 2010; Zhao et al., 2010)

In this study, a supervised-learning based method is proposed to detect congestion fronts based on the traffic flow speed changes observed during an incident over adjacent segments. Differences between upstream and downstream traffic flow speeds has been used to detect incidents. Incident related congestion front detection is carried out on the links, that are classified as non-recurrently congested in anomaly detection phase (Section 5.1). Incident related congestion fronts are classified using a supervised learning-based methods. A binary classification method is developed using following classes:

- Incident related congestion fronts (ICF)
- Non-incident related congestion fronts and congestion upstream for all NRCs (Other)

Observed travel speeds and calculated anomaly factors (Section 5.1) in upstream and downstream of segment is used as input for binary classification. Input vector consists of:

- Speed Difference: Nominal difference between average speeds observed on adjacent upstream and downstream segments of a segment.

$$\Delta V_{s,t}^{u,d} = \frac{1}{n(\mathrm{Up}_{s,u})} \sum_{i \in \mathrm{Up}_{s,u}} V_{i,t} - \frac{1}{n(\mathrm{Down}_{s,u})} \sum_{i \in \mathrm{Down}_{s,d}} V_{i,t} \qquad (5.8)$$

where $\mathrm{Up}_{s,u}$ is the set of upstream segments of segment s with a subgraph depth of u.
$\mathrm{Down}_{s,d}$ is the set of downstream segments of segment s. d denotes number of downstream segments.
$V_{s,t}$ is the travel speed observed on segment s on epoch t

- Speed Difference Deviate: Standard deviate of the current *Speed Difference* on a time-series consisting of historic data of the corresponding upstream and downstream segments.

$$SND_{s,t}^{u,d,h} = \frac{\Delta V_{s,t}^{u,d} - \frac{1}{n}\sum_{n=0}^{h} \Delta V_{s,t-h}^{u,d}}{S} \qquad (5.9)$$

where $\Delta V_{s,t}^{u,d}$ is the average travel speed difference between upstream and downstream segments segment s on epoch t (See Equation 5.8).

u is the subgraph depth of upstream segments.

d is the number of downstream segments.

h is the length of historic datapoints on time-series. Datapoints correspond to the same time-of-day.

S is the standard deviation of the datapoints.

- Downstream Anomaly Factor: Anomaly factors calculated for the adjacent downstream segments of a segment.

$$AF_{s,t}^{d} = \frac{1}{n(\mathrm{Down}_{s,u})} \sum\nolimits_{i \in \mathrm{Down}_{s,d}} AF_{i,t} \tag{5.10}$$

where $\mathrm{Down}_{s,d}$ is the set of downstream segments of segment s. d denotes number of downstream segments. $AF_{s,t}$ is the anomaly factor calculated for segment s on epoch t (See Section 5.1).

In analysis, set of upstream segments consist of all adjacent upstream segments and the link itself, whereas set of downstream segments of segment includes only the links which are on the same flow direction of the same road (Figure 5.1b). Classification model is using decision tree and support vector machine-based supervised learning models using annotations of *ICF* and *Other* segments (Figure 5.1c). For each epoch, all congested segments are classified as an incident related congestion front or not.

Congestion front identification phase is followed by congestion stretch identification, in which upstream boundaries of detected congestion fronts are determined.

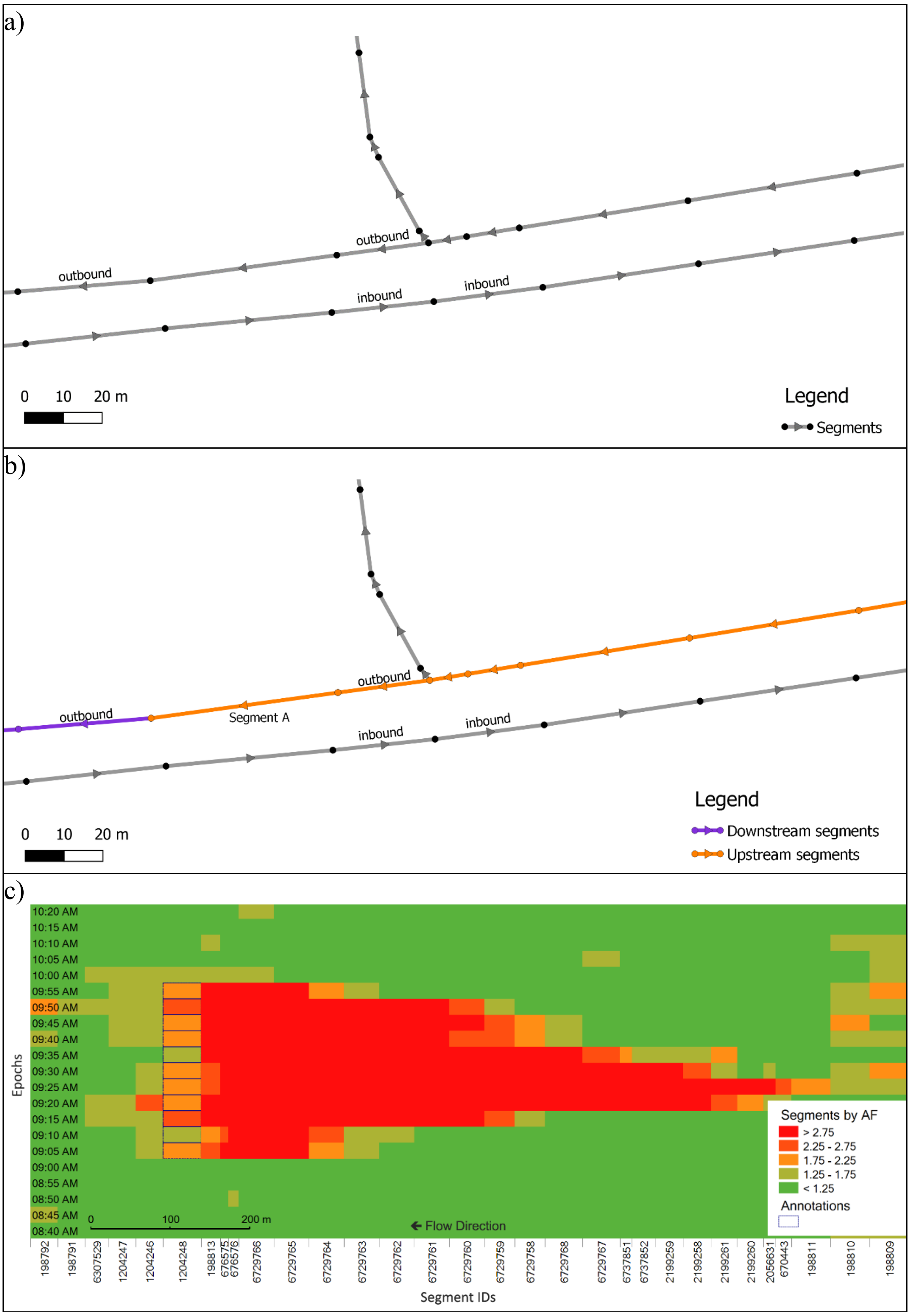


Figure 5.1 a) Segments b) Upstream and downstream segments of Segment A c) An incident related congestion front annotated on Segment A

5.2.3 Congestion Spillback Identification

A congestion spillback denotes the set of links affected by an NRC in epoch t (Figure 5.2). Upstream links of a congestion front is scanned, and congested links are determined by a threshold-based qualification. To perform necessary topological search along the segments, a network topology graph is built using FCD segment features and created links are annotated with the length, road name and flow direction of the corresponding segment (Section 5.2.1). In search for congestion downstream dissolution, the anomaly factor threshold determined in Section 5.1 is used. All adjacent upstream segments are traversed and included in the spillback until the links with below threshold AF ($AF < T_{AF}$) are reached. Due to temporal changes in AF along the downstream links, a search tolerance span is implemented to avoid under-detection of congested links due to links with under threshold AF values.

5.2.4 Congestion Impact Area Construction

A congestion impact area is defined as the continuous set of links with epochs which describe the spatiotemporal impact area of a congestion. In order to identify spatiotemporal extent of congested regions, detected congestion spillbacks in each epoch are merged using their level of relatedness in time and space. Construction is performed in each sequential epoch to be able to use the method for real-time congestion detection. For each epoch, detected congestion spillbacks are either merged into a set of spillbacks grouped as a congestion or a new set is initialized. Merging is carrying out using a rule-based approach. Front of the spillbacks are assumed to be the location of a possible incident or bottleneck causing the congestion, therefore used as a reference location to calculate relatedness among spillbacks in consequent epochs. A congestion spillback is merged into an existing congestion based on the spatial distance of their fronts and time-distance. Spatial distance threshold for matching is presented as a constant value, while time-distance threshold is proposed as a function of spatial extent of the previous congestion to merge. Time-distance threshold gets higher to provide more tolerant merging as the detected congestions covers a larger area. A pseudo-code describing the rule-based merging is given in Figure 5.3.

5.3 Spatiotemporal Information Matching

In this study, methods to detect traffic incidents using two independent data-sources, social media data (SMD) and floating car-data (FCD), are presented. Traffic related events reported on Twitter are detected using information retrieval methods and geocoded using Traffic Event Geocoder framework. FCD is used to detect local incident related non-recurrent congestions (NRC) in the road network.

A model based on spatial temporal relatedness is proposed to match the information which are detected from independently from FCD and SMD. A matching score based on a normalized spatial distance and time distance between event tweets and NRCs is used for matching. The results are augmented with contribution of two other parameters: locations being on the same flow direction of a street and the spatiotemporal impact area of the NRC. Individual scores which constitute the matching score are:

Spatial Distance Score: Spatial distance is measured as the network distance between geocoded location of an event tweet to the closest link of an NRC. Distance score values are normalized using a threshold distance to fit in a scale from 0 to 1, e.g., distance at threshold gets a score of 0, and gets a score of 1 when distance is 0 meter. Pairs which are farther than threshold are not evaluated as a possible match.

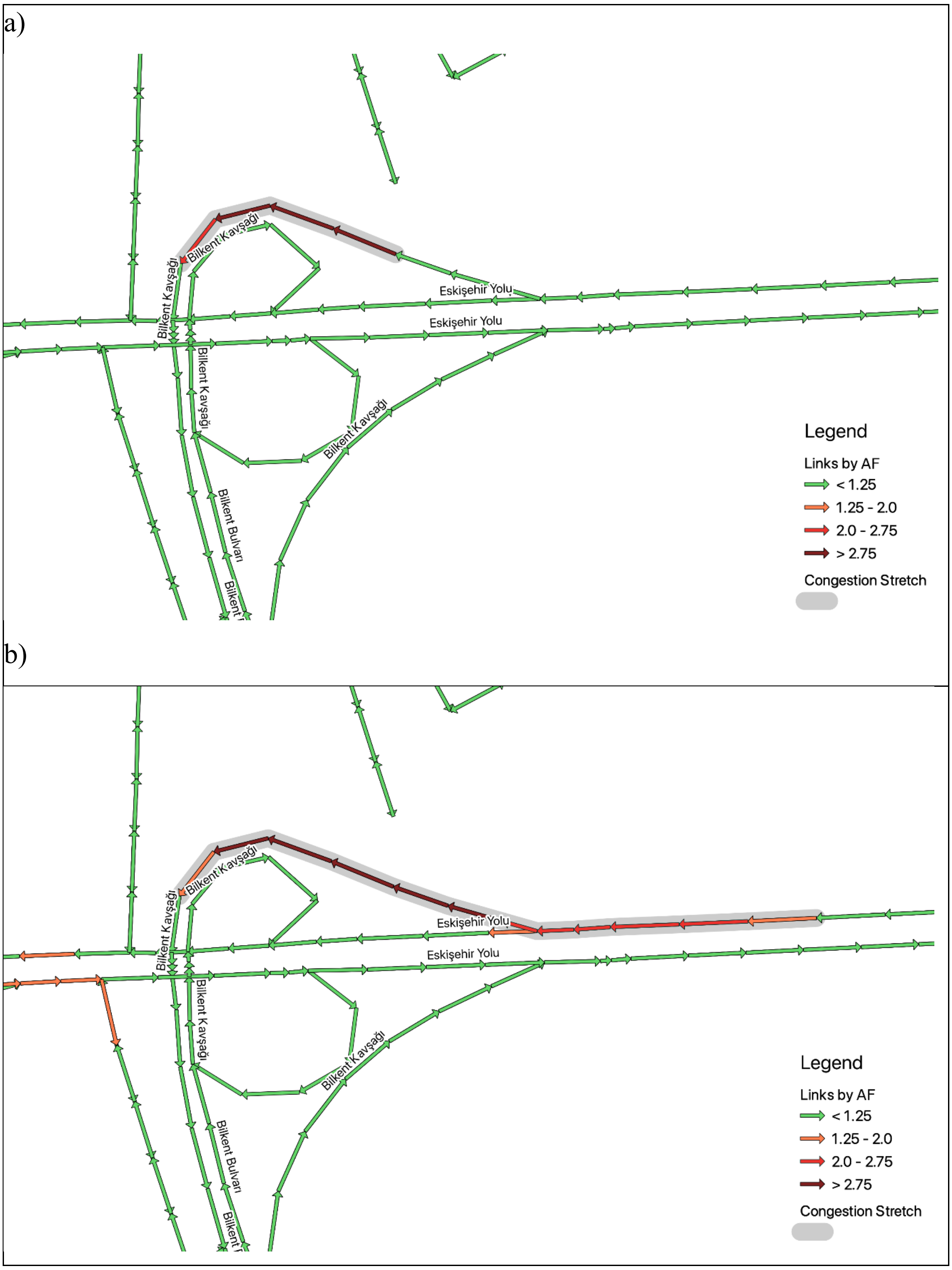


Figure 5.2 Example congestion spillbacks detected for an NRC in a) epoch t and b) in epoch t+3

```
$spillbacks_t$: links impacted by a congestion in epoch t
$congestions_t$: identified congestions which extends to epoch t
$dist^S_{a,b}$: number of links constituting shortest path connecting link a to link b
$dist^T_{a,b}$: time distance between congestion c and spillback
$T^T_e$ : time distance threshold based on congestion spatial extend e for merging
$T^S$ : spatial distance threshold for merging
$front_{c,t}$: link which describe the front of congestion c at epoch t
$front_s$: link which describe the front of spillback s
$extent_c$: spatial extent of congestion c
$t^{LAST}_c$: last epoch in a congestion timespan
$b$: epochs to check in past
For $t$ = l in epochs
    For each $s$ in $spillbacks_t$
        For each $c$ in $congestions_{t-b \ldots t-1}$
            If $dist^S_{front_s,,front_{c,t^{LAST}_c}} \leq T^S$ & $dist^T_{s,c} \leq T^T_{extent_c}$
                merge $s$ to $c$
        If $s$ is not merged
            create new congestion from $s$
```

Figure 5.3 Congestion Identification Algorithm

Time Distance Score: Time distance is the minimum absolute time difference between the post time of an event tweet and the impacted time span of the NRC. Time distance value is normalized to interval (0-1) using a maximum threshold value, closer pairs are getting a higher score. Values above threshold are excluded from evaluation.

Impact Score: Spatiotemporal impact area of NRCs is included in the score in order to give a higher priority to NRCs with a bigger impact area in contesting situations. Impact score is an approximation of total delay, calculated using the sum of differences of observed and estimated travel times along the segments impacted by congestion, multiplied by segments lane count estimated using *roadclass* of the segment.

Street Match Score: To improve matching accuracy, an extra score is added, when both geocoded location of the event and location NRC are on the same flow direction of a street. Overall matching score is a composite score, consisting of spatial distance score, time distance score, impact size score and street match score.

5.3.1 Match Confidence Estimate

Urban road network has a complex topology, thus producing a noisy travel time data impacted by features such as signals and bottlenecks. Such noise can mimic non-recurrent congestion patterns, causing false alarms and random information matches. In order to quantify reliability of information matching a measure based on a delta-score is introduced. The delta-score is calculated as the difference between the matching scores of top and the second-best matches. Since the underlying distribution of delta-score is unknown, a distribution can be created using a simulation. A simulation is developed by assigning random incidents in road network in active hours and matched with the congestion detected in Section 5.2. Delta-scores of the simulated matches are calculated as a reference distribution for the variable. A match confidence estimate (C) is calculated as the p-value of a match in the distribution of the simulated delta-scores. Matches with a match confidence estimate (C) below a determined threshold are referred as confident matches. The match confidence estimate is used to minimize number of random matches and filter congestion detection results to avoid false detections.

CHAPTER 6

CASE STUDIES

In this chapter experiments using methods presented in previous chapters are conducted on a case study in Ankara. In Section 6.1, traffic event detection method given in Chapter 3 is evaluated on a tweet data set covering a region surrounding Ankara city. In Section 6.2, experiments of traffic event localization method given in Chapter 4 are conducted on a set of traffic events reported on Twitter. In Section 6.3, non-recurrent congestion detection method given in Chapter 5 is evaluated on a commercial FCD data covering a major arterial, Eskişehir Road. Finally, in Section 6.4, the traffic event tweets and non-recurrent congestions detected in Section 6.1 and Section 6.3 respectively are matched using of spatial information matching method given in Section 5.3.

6.1 Traffic Event Detection in Social Media

In this section, the data set, conducted experiments and the results are presented. The codes are implemented in C# language on ASP.NET platform. Machine learning models are developed by using Accord framework (Souza et al., 2014). All experiments are carried out on a Windows-based PC running on a 6-core Intel Core i7 3.2 GHz processor with 32 GB of memory.

6.1.1 Dataset

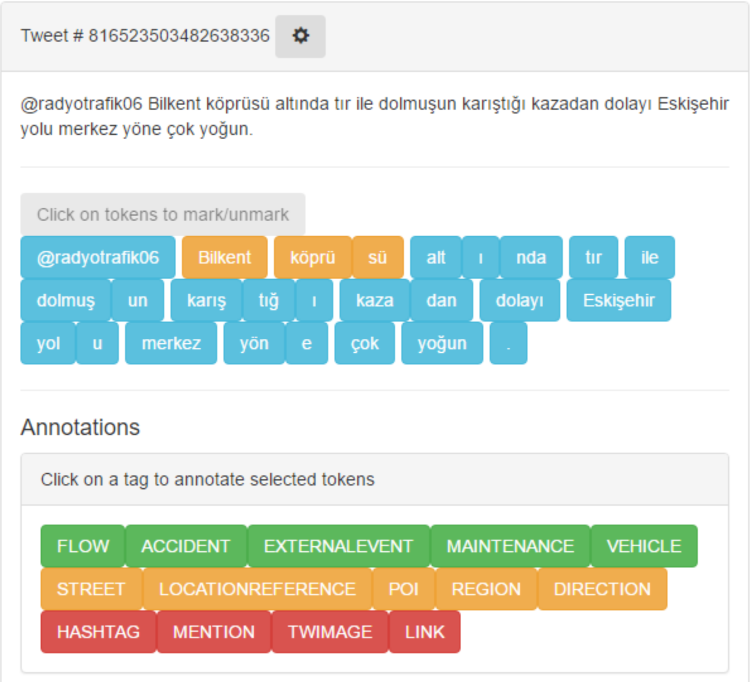
Our data set is a tweet collection that is retrieved by using Twitter Search API with a predefined set of keywords under location filtering defining 50mi radius around the city center of Ankara. The collection consists of 21,077 tweets, posted from January 1st to January 31st, 2017. For ground truth construction, tweets are manually annotated for the traffic entity recognition and classification by using a web-based application. In Figure 6.3, locations of the some of the traffic events reported in tweets are shown on map of Ankara around Eskişehir Road. As shown on the map, the reported sample events typically correspond to locations on the city road network.

The method involves supervised learning methods for named entity recognition and classification. A training data set covering 21,077 tweets, posted between January 1st to January 31st, 2017, are manually annotated for traffic event entity recognition (Section 3.4.4) and classification tasks (Section 3.4.5). To this aim, a web based custom tool, Entity Annotation Tool, is developed.

6.1.1.1 Entity Annotation Tool

Although there exist numerous tools for text annotation, these provide limited support for morphologically rich languages. A web-based tool which provide functions to handle complex Turkish morphology has been developed for annotation purposes for supervised learning models.

Tool uses TRMorph toolset (Coltekin, 2014) to perform morphological analysis for each token and segment tokens into *subtokens*. Each *subtoken* is identified with a unique identifier for annotation purposes. Annotators can assign tokens with the tags which are customized for traffic event related tweets. (Figure 6.1 and Figure 6.2). Segmented *subtokens* and assigned tags are stored in MongoDB as json formatted documents.



Tweet # 816523503482638336

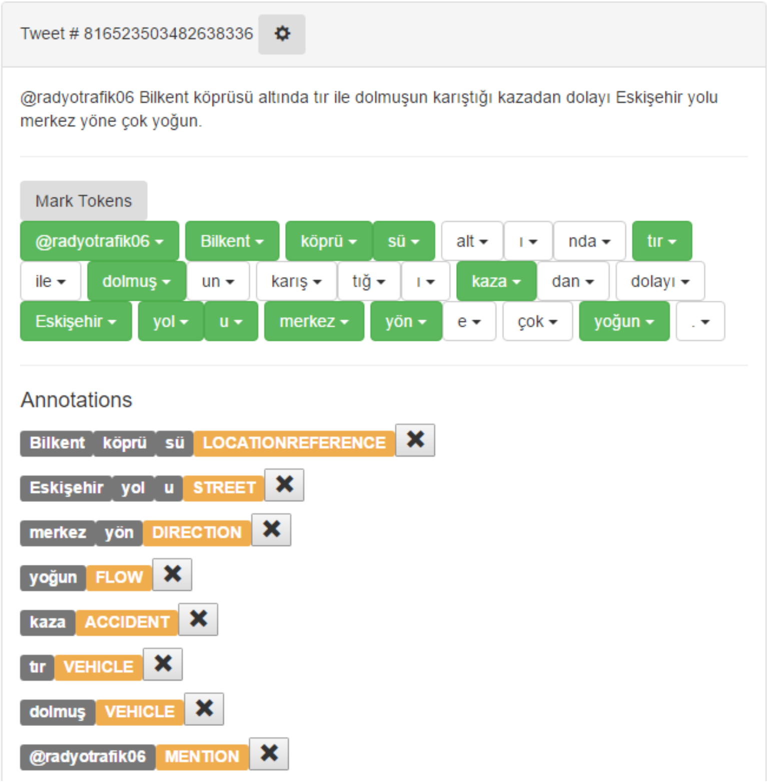
@radyotrafik06 Bilkent köprüsü altında tır ile dolmuşun karıştığı kazadan dolayı Eskişehir yolu merkez yöne çok yoğun.

Click on tokens to mark/unmark

Annotations

Click on a tag to annotate selected tokens

Figure 6.1 Annotating text with entity tags



Tweet # 816523503482638336

@radyotrafik06 Bilkent köprüsü altında tır ile dolmuşun karıştığı kazadan dolayı Eskişehir yolu merkez yöne çok yoğun.

Mark Tokens

Annotations

Bilkent köprü sü LOCATIONREFERENCE ✖

Eskişehir yol u STREET ✖

merkez yön DIRECTION ✖

yoğun FLOW ✖

kaza ACCIDENT ✖

tır VEHICLE ✖

dolmuş VEHICLE ✖

@radyotrafik06 MENTION ✖

Figure 6.2 A tweet annotated with tags using Entity Annotation Tool

6.1.1.2 Class Labelling

The focus of the study is to detect and describe incidents or conditions which affect the traffic flow using social media stream and floating car data in real-time. Relevant posts in social media stream are detected using supervised machine learning methods. A training data set is labelled manually to define the ground truth for the classification tasks. Two classes are defined to categorize the stream, *Direct Traffic Report (DTR),* and *Other*. *DTR* denotes that the posting is a direct and immediate report of an incident or road condition which might affect the traffic flow. On the other hand, *other* denotes that the post does not meet the criteria of *DTR*.

More specifically, *DTR* denotes any of the following cases:

- Traffic accident reported by individuals
- Weather condition reported by individuals
- Road condition and warnings reported by individuals or institutes
- Ongoing road maintenance work reported by individuals or institutes if the report is recent or presumably effective by the time the tweet is posted and includes a location reference.

DTR excludes the following cases:

- News article (due to uncertainty in the recency of the article)
- Condition that is not clearly affecting traffic flow, such as posts reporting conditions affecting pedestrians without a reference to a road or traffic flow condition
- Report without a location reference
- Indirect report of individuals referring another information source, such as media or social media
- Speed radar location warnings

As the result of the annotation, 649 tweets are labeled as *DTR*.

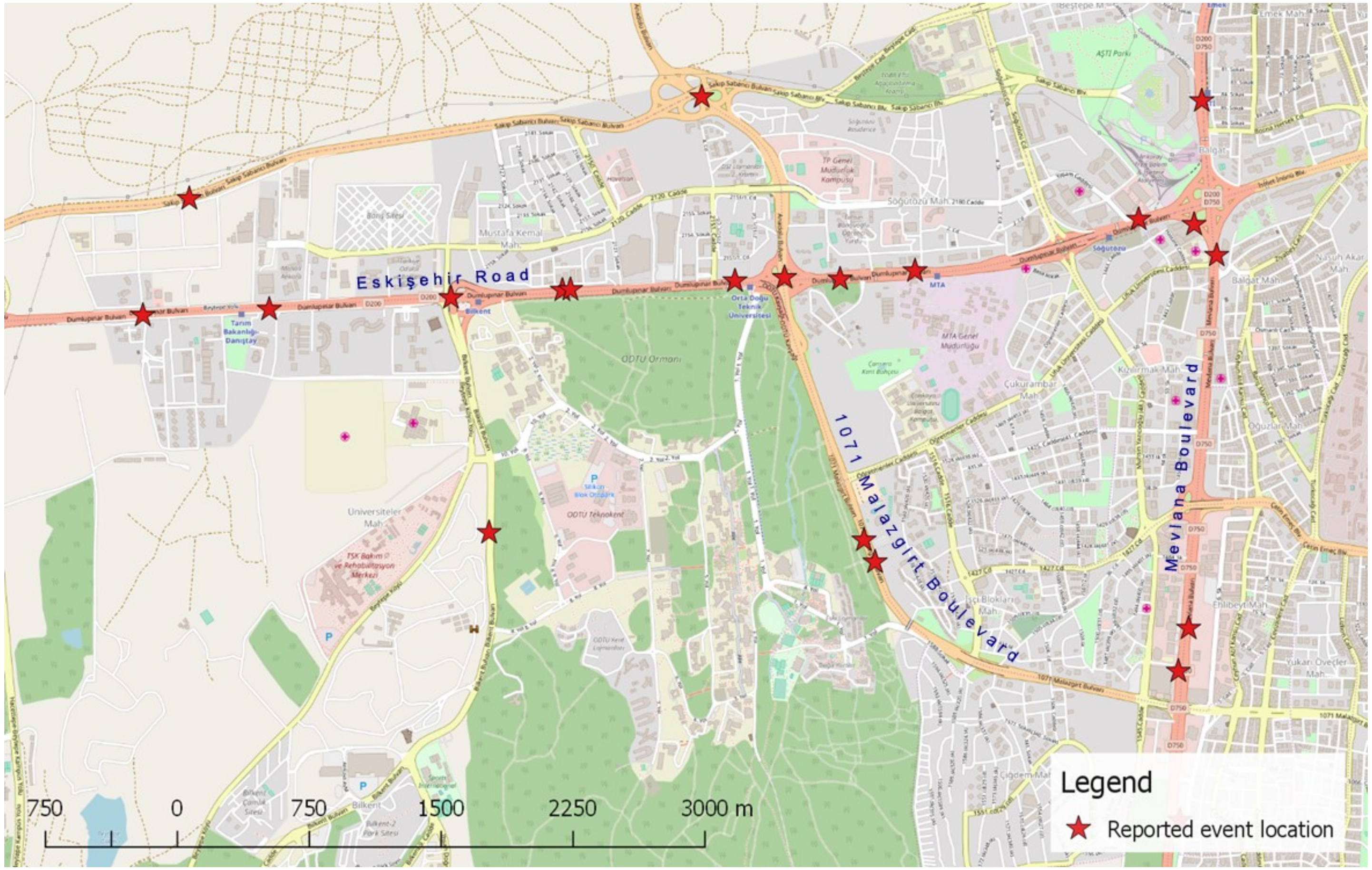


Figure 6.3 Locations of DTR tweets around Eskisehir Road

6.1.2 Traffic Event Entity Recognition

Traffic event related named entity recognition task is performed by using Conditional Random Fields (CRF) technique. Annotated tags are transformed into Inside-Outside-Beginning (IOB) format, which is a standard annotation format for tagging tokens in linguistic tasks. Each annotated token is marked with a suffix, I for inside, O for outside and B for begin, to determine the chunks of corresponding annotations. More concretely, IOB2 format, which is an extension over IOB by annotating the beginning of each chunk as B, is used for entity annotation. Tagging is performed on *subtokens*. *Subtokens* are represented with their surface text, morphological tags that are assigned during morphological analysis, and tag annotations in the training set. *Subtokens* that are not annotated with a tag are represented with an *S* tag. Annotation is performed only on tweets that are labeled as *DTR*. Training data set is created by including annotated tweets with *DTR* label and equal number of tweets with *Other* label. Tweets with *Other* label are not annotated with tags and each token is represented with an *S* tag.

Performance results for the traffic event recognition model are given in Table 6.1. In order to improve the entity recognition accuracy, tags with low recognition rate in validation experiments are merged into groups. All tags under incident category *Incident*, *ExternalEvent*, *Maintenance*, *RoadCondition* are modeled as *Incident*, *People* and *Vehicle* as *Entity*, *Location* and *Region* as *Location*. Resulting performance metrics for the merged traffic event recognition model are summarized in Table 6.2. For each named entity group, correctly recognized named entities are given as true positives (TP) and incorrectly recognized ones are given as false negatives (FN). Tokens that are incorrectly recognized to be in the given named entity group are given as false positives (FP). Precision is the ratio of the successfully recognized named entities to all entities recognized to be in the given group. Recall is the ratio of the successfully recognized named entities to all actual named entities in the given group. F1-Score (F1) is the harmonic average of Precision and Recall, which provides a single score considering two metrics. The results in the table are ranked with respect to F1-Score. As seen in the results, several entity groups including *Maintenance* and *Direction* have high precision and F1 values, whereas some groups are much harder to recognize, such as *Incident* or *Connection*. This is possibly due to the variety of the tokens in the entity group, such as *Maintenance* is generally denoted by phrases including the word *bakım* (Eng. maintenance). Constructed classification model is tested under 10-fold cross-validation. Test is performed on test datasets which consist of tweets from both *DTR* and *Others*.

Table 6.1 Precision, Recall and F1-score for recognition of merged traffic related named entities

Tag	TP	FP	FN	Recall (%)	Precision (%)	F1 (%)
Direction	176	31	68	72.1	85.0	78.0
Maintenance	207	110	25	89.2	65.3	75.4
DirectionIndicator	131	34	69	65.5	79.4	71.8
Damage	54	41	17	76.1	56.8	65.1
Flow	194	90	154	55.7	68.3	61.4
Lane	36	10	42	46.2	78.3	58.1
LocationIndicator	276	210	276	50.0	56.8	53.2
Street	697	1,040	237	74.6	40.1	52.2
ConnectionIndicator	12	0	23	34.3	100.0	51.1
RoadFeatures	61	62	82	42.7	49.6	45.9
Location	271	130	625	30.2	67.6	41.8
Connection	13	1	38	25.5	92.9	40.0
Incident	17	1	541	3.0	94.4	5.9
Accident	0	362	0	N/A	0.0	N/A
RoadCondition	0	592	0	N/A	0.0	N/A
Region	0	473	0	N/A	0.0	N/A
Vehicle	0	57	0	N/A	0.0	N/A
ExternalEvent	0	0	14	0.0	N/A	N/A
Entity	0	0	101	0.0	N/A	N/A
All	2,145	3,244	2,312	48.1	39.8	43.6

Table 6.2 Precision, Recall and F1-score for recognition of merged traffic related named entities

Named Entity	TP	FP	FN	Prec. (%)	Recall (%)	F1 (%)
Maintenance	295	1	68	99.7	81.3	89.5
Direction	177	10	71	94.7	71.4	81.4
Damage	55	8	21	87.3	72.4	79.1
DirectionInd.	131	12	74	91.6	63.9	75.3
Flow	201	56	164	78.2	55.1	64.6
Lane	36	10	42	78.3	46.2	58.1
Street	781	861	272	47.6	74.2	58.0
LocationInd.	320	168	306	65.6	51.1	57.5
Location	652	559	417	53.8	61.0	57.2
RoadFeatures	64	48	87	57.1	42.4	48.7
ConnectionInd.	10	0	26	100.0	27.8	43.5
Incident	289	804	300	26.4	49.1	34.4
Entity	33	58	74	36.3	30.8	33.3
Connection	10	0	43	100.0	18.9	31.7
ExternalEvent	0	0	20	0.0	N/A	N/A
Time	0	0	5	0.0	N/A	N/A
All	3054	2595	1999	54.1	60.4	57.1

6.1.3 Classification Performance

Classification is the step to detect traffic event related tweets. In this study, SVM, Decision Tree classifier with C4.5 algorithm and Naïve Bayes classifier are used for classifying tweets as *DTR* or *Other*. Feature vector corresponding to a tweet consists of three types of features: *Stems* under bag of words model, *Top Named Entities*, and *All Named Entities*.

Stems are the stemmed forms of the words that are generated in morphological analysis step. Due to the high number of stemmed words extracted from tweet contents, feature selection is applied such that the terms below a term frequency threshold are filtered out. Each classifier is run under several term frequency thresholds (TFT), which indicate the percentage of top terms included in the selected set with respect to all terms in the training corpus. For instance, feature selection under TFT value of 5 consist of only the top 5% of the terms ordered by their term frequencies in the corresponding training corpus.

Top Named Entities are a subset of traffic related named entities, which performed best in the Traffic Related Named Entity Recognition model. *Top Named Entities* consist of *Direction*, *Flow*, *Maintenance*, *Lane* and *Damage*. *All Named Entities* consist of all entities included in the model. Both *Top Named Entities* and *All Named Entities* exclude *DirectionIndicator*, *LocationIndicator*, *ConnectionIndicator*, which are auxiliary named entities that are used together with other named entities, such as *in* Ümitköy or *to* 12th Street.

Classification performance experiments are conducted under 10-fold cross validation. This set of experiments aim to analyze the effect of various feature groupings for SVM, Naive Bayes and Decision Tree (C4.5) classifiers. The results are summarized in Table 6.3.

Table 6.3 Classification performance with different feature sets and classifiers

Feature set	Classifier	Tokens[1]				Subtokens[2]			
		TFT	Recall	Prec.	F1	TFT	Recall	Prec.	F1
Stems	SVM	33	29,9	81,2	43,7	33	55,5	93,3	69,6
	C4.5	5	61,9	41,7	49,8	10	54,0	60,0	56,8
	NB	10	51,3	62,6	56,4	10	55,5	62,7	58,9
Stems Top Named Entities	SVM	10	56,4	82,1	66,8	10	53,3	92,0	67,5
	C4.5	5	68,1	59,4	63,5	5	66,6	61,3	63,9
	NB	5	54,7	71,7	62,1	5	60,2	71,6	65,4
Stems All Named Entities	SVM	2	70,7	65,7	68,1	25	67,0	73,7	70,2
	C4.5	2	73,3	50,7	60,0	10	72,4	54,8	62,4
	NB	2	74,6	62,1	67,7	2	70,7	65,7	68,1

[1] Models developed disregarding morphological features

[2] Using *subtokens* generated in morphological analysis

As given in the previous section, *DTR* tweets, which are correctly classified as *DTR* are given as true positives (TP) and incorrectly classified as *Other* are given as false negatives (FN). Tweets that belong to *Other*, which are correctly classified as *Other* are given as true negatives (TN), whereas the ones that are incorrectly classified as *DTR* are given as false positives (FP). Precision is the ratio of the number successfully detected *DTR* to the number of all *DTR* labeled tweets. Recall is the ratio of the number of actual *DTR* labeled tweets to the number of tweets detected as *DTR*. A detailed break-down of the analyzed feature sets and their performance values for the SVM classifier are given in Table 6.4.

In order to observe the contribution of the extracted morphological features on the classification task, the experiments are conducted separately using models based on tokens and *subtokens*. In Table 6.3, results for several classifiers comparing the models built with tokens and *subtokens* are presented.

SVM classifier performed best with the *subtoken*-based feature set including *stems* and *AllEntities* with a F1-Score of 70.2%. SVM classifier reached this score by using the top 25% of all terms sorted by their frequency in all documents. Classifier was able to detect 67% of all traffic related events, while 73.7% of the events detected as traffic related events were actually traffic related.

Highest F1-Score of 63.9% was achieved by C4.5 algorithm-based classifier using the *subtoken*-based feature set consisting of *stems* and *TopEntities*. This score was reached with a TFT level set at 5%. Classifier was able to detect 66.6% of all traffic related events, while 61.3% of the events detected as traffic related events were actually traffic related. Naive-Bayes classifier reached highest F1-Score of 68.1% with *subtoken*-based models employing *stems* and *AllEntities*. Highest score was reached at a TFT level of only 2%.

Table 6.4 Classification performance of SVM in precision, recall and F1-score

Feature Types: Stems

TFT	TP	FP	FN	TN	Recall (%)	Precision (%)	F1 (%)
1	0	0	650	18288	0	NA	NA
2	187	37	463	18281	28.8	83.5	42.8
5	322	48	328	18274	49.5	87.0	63.1
10	337	26	312	18293	51.9	92.8	66.6
25	357	23	292	18297	55.0	93.9	69.4

Feature Types: Stems and Top Named Entities

TFT	TP	FP	FN	TN	Recall (%)	Precision (%)	F1 (%)
1	53	12	596	18262	8.2	81.5	14.8
2	288	33	362	18278	44.3	89.7	59.3
5	329	30	321	18286	50.6	91.6	65.2
10	346	30	303	18284	53.3	92.0	67.5
25	344	27	305	18289	53.0	92.7	67.5
33	341	27	308	18289	52.5	92.7	67.1

Feature Types: Stems and All Named Entities

TFT	TP	FP	FN	TN	Recall (%)	Precision (%)	F1 (%)
1	460	355	189	17927	70.9	56.4	62.8
2	453	226	196	18088	69.8	66.7	68.2
5	430	160	219	18157	66.3	72.9	69.4
10	432	157	217	18161	66.6	73.3	69.8
25	435	155	214	18165	67.0	73.7	70.2
33	434	155	215	18165	66.9	73.7	70.1

Models employing *subtoken*-based feature sets scored better in all classification tests. Highest F1-Scores achieved among *subtoken* and token-based features sets are; 70.2% and 68.1% for SVM classifier, 63.9% and 63.5% for C4.5 classifier and 68.1% and 67.7% for NB classifier. Some example tweets which are correctly classified as DTR class are given in Table 6.5. Example set consist of tweets reporting traffic related incidents, maintenance works and other road conditions. Several challenges have been observed due to the strict inclusion criteria for *DTR* class. *DTR* label is used for only immediate and direct reports. Any tweet not matching these criteria are marked as *Others*. Indirect reports of incidents that refer other sources or news articles, which do not reflect a recent incident, are labeled as *Others*. Such tweets affected the performance of classification task negatively, due to lack of features to model subtle semantic differences. A set of misclassified tweets are given in Table 6.6.

The difficult cases that lead to incorrect labeling for false positive instances can be listed as follows:

- Non-DTR tweets using traffic terms (FP Example 1 in Table 6.6)
- Tweets that do not report an immediate condition (FP Example 2 in Table 6.6)
- Tweets reporting conditions that do not clearly affect the traffic flow. (FP Examples 3, 4 and 5 in Table 6.6)
- Tweets including a question on traffic flow rather than a report (FP Example 6 in Table 6.6)

Similarly, the cases that lead to false negative labeling can be summarized as follows:

- Tweets in which location descriptions are not successfully identified by named entity recognizer (FN Examples 2 and 6 in Table 6.6)
- Tweets that describe events using irony or in unconventional ways (FN Example 3 in Table 6.6)
- Tweets that do not refer to a particular location in the city (FN Examples 1 and 4 in Table 6.6)

- Unusual event types that are reported rarely (FN Example 5 in Table 6.6)

Table 6.5 TP Examples from classification experiment results

1	*Çiftlik Kavşağı Cumhurbaşkanlığı yolu üzerinde sol şerit de araç arızası var bölgeyi olumsuz etkiliyor* Vehicle break-down at Çiftlik Intersection, in the left lane on Presidential Road, is affecting region negatively
2	*Elvankent 12.cadde büyük kaza var 10-12 araç birbirine girmiş o caddenin alternatiflerini kullanın* A major incident on Elvankent 12 th Ave involving 10-12 vehicles, use alternative routes
3	*Gölbaşı Taşpınar Mahallesi 2855. Cadde'de yol bakım çalışmamız devam ediyor* Ongoing road maintenance on 2855th street in Taşpınar neighbourhood of Gölbaşı
4	*1071 Malazgirt Bulvarı kilit. 20 dk oldu böyle bekliyoruz kaza falan mı var?* Malazgirt Boulevard is gridlocked. We have been waiting here for 20 minutes, is there and accident or so?
5	*#YolDurumu İncek Bulvarı, İncek yönünde hasarlı kaza!* #RoadCondition Accident with damages on İncek Boulevard in İncek direction
6	*Eskişehir Yoluna dönüşler kapatıldığı için Konya Yolu trafiği Balgat Demirköprü' ye kadar kilit.* Due to blocked ramps to Eskişehir Road, traffic on Konya Road is gridlocked until Balgat Demirköprü
7	*Eskişehir Yolu Konutkent kavşağı... çok ciddi kaza can kaybı olabilir 4 araba birbirine girmiş durumda* Konutkent intersection on Eskişehir Road, a serious accident with possible causalities, 4-vehicle pile-up crash
8	*Pursaklar İlçesi, Saray - Cumhuriyet Mahallesi, As Sokak içerisindeki yağmursuyu ızgaraları temizlenmektedir.* Rainwater grates clean-up is underway on As Sokak, in Saray-Cumhuriyet districts in Pursaklar subprovince
9	*Güdül İlçesi muhtelif cadde ve mahallelerinde karla mücadele çalışmamız devam ediyor* Snow removal is underway in several streets and districts of Güdül subprovince.
10	*Öveçler 2. ve Öveçler 4. cadde arasında kalan yollar buzla kaplı.Kaldırımlarda da yürümek oldukça zor.* Ice on all the roads between Öveçler 2nd and Öveçler 4th avenues.Very difficult to walk on sidewalks
11	*yasamkent 3267 cadde ve atabilge okul yolu tamamen kapalı acil müdahale* 3267th street in Yaşamkent and the road to Atabilge school are blocked completely. Urgent intervention needed.
12	*Anadolu bulvarı merkez yöne marsandiz köprüde sol şeritte 2 araçlık kaza* 2-vehicle accident on inbound Anadolu Boulevard at Marsandiz Bridge
13	*Anadolu bulvarı kilit, kaza var 3 araba var* Anadolu Boulevard is gridlocked, there is an accident with 3 cars
14	*Elmadağdan Kırıkkale'ye gidiş kaza var trafik kilit* Accident on the way from Elmadağ to Kırıkkale, traffic is gridlocked

Table 6.6 FP and FN examples from classification experiment results

False Positives	
1	*Hayallerime giden yolda trafik var* There is traffic on the roads to my dreams
2	*Etimesgut'taki yeni Gimsanın önündeki ışıklar 4 gündür yanmıyor. Kaza ihtimali artıyor..* The traffic lights in front of the new Gimsa in Etimesgut are not working for the last 4 days. Increased accident risk..
3	*Burası Demetevler 406.cadde YapıKredi ve Ziraat bankasının önündeki kaldırım.buz pisti gibi.Acil yardm bekliyoruz…* This is the sidewalk in front of Yapı Kredi and Ziraat Bank on 406th Street in Demetevler. Covered in ice. Urgent help needed.
4	*Ankara Bahçelievler 7.cadde 1966. Son Durak Eser Sitesi İnşaatı devam ediyor. Eser Sitesi* is under construction at last stop on 7th Street, (1966th) in Bahçelievler, Ankara
5	*Malazgirt mahallesi 1009 sokak ta karlar gerçekten temizlenmiş. Teşekkürler.* Snow has indeed been removed from 1009th street in Malazgirt neighborhood. Thank you.
6	*Havaalanı yolu ve Oran ile ilgili yol-kar durumu bilgisi olan var mi* Has anyone information about road-snow conditions for Airport Road and Oran?
False Negatives	
1	*Kuvvetli Buzlanma ve Don Olayının etkili olduğu Ankara Kent Merkezinde yol tuzlama çalışmalarımız devam ediyor…* Road-salting operation is underway in frost-hit Ankara city center
2	*Zırhlı birlikler çıkışında 4 araç 1 otobüsün karıştığı zincirleme kaza var. Trafik kilit ötesi.* There is a pile-up accident of 4 vehicles and a bus at the exit of Zırhlı Birlikler
3	*Ankaraya göktaşı falan düştü de insanlar kaçıyor mu anlamadım. Konya yolundaki trafik neye hacet?* Has Ankara been hit by a meteor and are people running from it? Why so much traffic on Konya road?
4	*Ankaranın heryeri buz pisti çok kaza var dikkat edin...* Everywhere in Ankara seems like an ice rink, many accidents, be careful...
5	*Yaşamkent kavşağı trafik ışıkları yanmıyor curcuna yaşanıyor* Malfunctioning traffic lights are causing confusion at Yaşamkent intersection
6	*Eskisehir yolu Ümitköy dönüsu.dolmus bilinmeyen nedenle* Eskişehir Road, Ümitköy exit. Minibus on fire for unknown reason

6.2 Traffic Event Localization

In this section, the performance of Traffic Event Geocoder is evaluated independently and along with other geocoder packages commonly used in localizing traffic events. The analysis is conducted on a case study covering Ankara city proper. Traffic event related tweets are collected for Ankara using Twitter Search API[4]. For the analysis, the following steps are carried out:

1. Training data set is obtained and annotated to construct TEER model.
2. OSM data is downloaded, and objects used for geocoding are extracted from the OSM data.
3. For geocoding performance evaluation, test data set of 136 tweets are geolocated manually using the location description in their text content.
4. A custom data set of landmarks are mapped to alleviate the missing data problem in OSM data, creating the Enriched-OSM data set.

TEER model is implemented using the CRF method, a supervised learning model that is commonly used in named entity recognition tasks (Section 6.1.2). Performance results for location entities in precision, recall, and F1-score under 10-fold cross-validation are given in Table 6.7 (see Table 6.2 for all named entity types).

Map data covering the extent of the area of interest is downloaded using Overpass API[5]. The data, consisting of 1,371,700 elements in a 261-MB XML file, is parsed using OsmSharp[6]. Some of the landmarks that are frequently used in event reports were missing in the OSM data. Such missing landmarks are mapped to create a custom dataset, *Enriched-OSM.* Enriched dataset is used as an alternative to observe the effect of enrichment of map data on geocoding. These extensions consist of (1)

[4] https://developer.twitter.com/en/docs/twitter-api/v1/tweets/search/api-reference/get-search-tweets

[5] https://wiki.openstreetmap.org/wiki/Overpass_API

[6] http://www.osmsharp.com

Ankara Kapıları (eng. Ankara Gates, entrance gate structures of Ankara), which are located on main streets at the entrances of Ankara, (2) fixed speed cameras, which are referred as *82 Radar* due to their speed warning label of 82 km/h on them, (3) and some well-known intersections, including *Ölüm Kavşağı* (eng. Intersection of Death) informally named as such after fatal accidents. Locations of such frequently used landmarks mapped to construct the *Enriched-OSM* data are given in Figure 6.4.

Table 6.7 Named entity recognition performance results for the TEER model

Tag	TP	FP	FN	Recall (%)	Precision (%)	F1 (%)
Direction	177	10	71	71.4	94.7	81.4
Direction Indicator	131	12	74	63.9	91.6	75.3
Street	781	861	272	74.2	47.6	58.0
Location Indicator	320	168	306	51.1	65.6	57.5
Location	652	559	417	61.0	53.8	57.2
Connection Indicator	10	0	26	27.8	100.0	43.5
Connection	10	0	43	18.9	100.0	31.7

Traffic Event Geocoder is implemented as a .Net application. A test data set consisting of 136 tweets is selected among the traffic event tweets with a clear definition of location. Sample tweets from the test data set are given in Table 6.8. In the tweets, the tag (IMAGE) denotes that the tweet includes an image. However, during processing and annotation, such attachments to the tweets were not considered and not processed. Event locations in the test data set are manually geocoded by domain expert operators. The number of operators participating in the ground truth label annotation was limited to 2, due to the limited availability of field experts. Operators, both of which were Geographic Information Systems (GIS) specialists, were provided with a map of road network geometry along with the following set of instructions to standardize the markings:

- Markings should be performed on GIS software.

- Event locations should be marked on the road network geometry provided.
- Any reliable sources can be employed to choose the best location for the mentioned terms.
- Locations should be implied only from the terms in the tweet content. Other meta information, including embedded images or videos, should be ignored.
- Event locations can be represented as a point or a polyline if needed for continuous location definitions.
- Directions terms should be considered to mark the event on the stretch with the matching flow direction on the divided roadways.

Table 6.8 Sample tweets from the collected test data set

@radyotrafik06 eskişehir yolu merkez yönde gordion köprüsünden itibaren dur kalk ilerliyor. Ümitköy köprüsüne kadar… (IMAGE) #YolDurumu Stop and go traffic on inbound Eskişehir Road starting from Gordion Bridge. Until Ümitköy Bridge...
Yozgat bulvarı forum Ankara kavşağı kilit acil trafik Ekibi gerek. (IMAGE) Gridlock at Forum Ankara intersection on Yozgat Boulevard(,) traffic patrol needed urgently.
Eskisehir yolu ümitköy dönüsu.dolmus bilinmeyen nedenle yanıyor (IMAGE) Eskişehir Road Ümitköy Exit. Minibus in fire for unknown reason
Konya Yolu Taurus önünde kalabalık trafik var, buzlanmadan dolayı araçlar çukurambar tarafina dönemiyorlar. Heavy traffic in front of Taurus on Konya Road, vehicles cannot turn to Çukurambar direction due to icing.
Eskişehir yolu konutkent kavşağı... çok ciddi kaza can kaybı olabilir 4 araba birbirine girmiş durumda (IMAGE) Konutkent intersection on Eskişehir Road, a serious accident with possible causalities, 4-vehicle pile-up crash
Kuzey Çevre yolu Batıkent yönü Forum Ankara civarı. Yolda araç yangını var. İtfaiye ve ambulans geldi. Trafik duruyor! Batıkent direction on Northern Ring Road around Forum Ankara. Vehicle fire on the road. A fire truck and ambulance arrived. Traffic at standstill!
Mevlana Bulvarı, Malazgirt Bulvarı bağlantısı Gimat yönünde kaza!!! ... Accident at Malazgirt Boulevard exit on Mevlana Boulevard Gimat direction!!! ...

Agreement levels among operators are measured using network distances between markings. Event locations which are marked within 1000 meters by both operators are accepted as agreed locations. Operators agreed on the locations of 118 events, while network distances of markings were higher than the threshold in 18 events. An agreed location is set by the geometric center of two markings on the road network.

Evaluations of the geocoding results are performed on 118 events with the agreed locations. The error of geocoding is measured using the distance of the geocoding result with the agreed location. Distance is defined as the total length of links connecting two locations, geocoded, and agreed location, respecting the link flow directions. Traffic Event Geocoder using OSM (TEG) and Traffic Event Geocoder using *Enriched-OSM* (TEGE) are evaluated separately. Although most of the terms are mapped correctly on the OSM elements, some incorrect mappings yielded outlier errors. Hence median positional error which is more resistant to outliers is used as the main metric to evaluate the results. The average and median positional errors of geocoding are 757.0 and 232.4 meters for TEGE and 1488.5 and 379.2 meters for TEG. TEG geocoded 56 events within 750-meter error, and 7 events within 750 to 1500-meter error. Whereas TEGE geocoded 68 events within 750-meter error, and 6 events within 750 to 1500-meter error. The number of events which could not be geocoded is 36 in both TEG and TEGE. Geocoding performance of TEG and TEGE are summarized in Table 6.9. Median and average positional errors grouped by the type and the number of location terms located on the map per event in TEG experiments are given in Table 6.10. Geocoder rules are able to locate the events, if a single *Location* term or a *Road* term along with a *Location* or a *Connection* term are matched on map data. Median positional error decreases with the increasing number of terms geocoded, reaching 183-meter error when geocoding is performed using 1 *Road* term with 2 *Location* terms. Geocoder fails to locate 23 events in which only a single *Road* term is located on map. No location term is detected in 13 events.

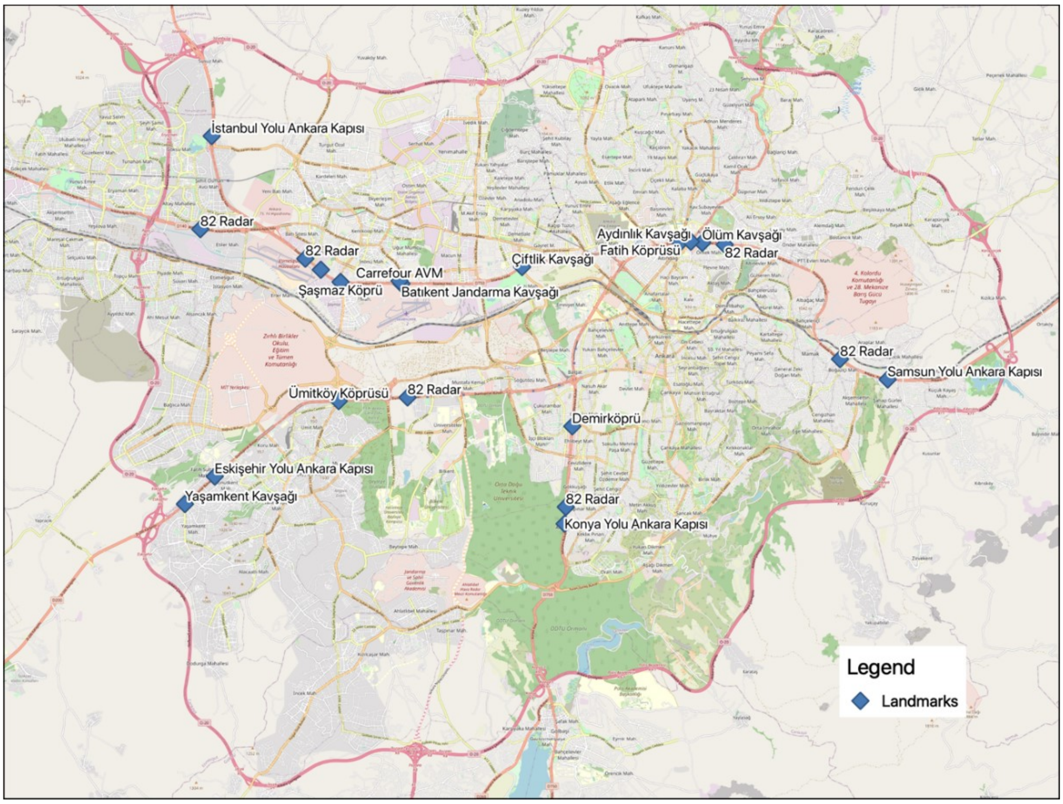


Figure 6.4 Frequently referenced landmarks appended to create Enriched-OSM

Table 6.9 Positional errors in the experiments of overall Traffic Event Geocoder method

Positional Error	Number of events (#) and percentages (%)			
	TEG		TEGE	
(meters)	#	%	#	%
0-750	56	47.5	68	57.6
750-1500	7	5.9	6	5.1
1500-5000	12	10.2	5	4.2
5000+	7	5.9	3	2.5
N/A	36	30.5	36	30.5

Table 6.10 Average positional errors by location terms located on map in TEG

Location terms geocoded	Median positional error (meters)	Average positional error (meters)	Number of events	Number of events geocoded
1 Location	420	1383.8	27	27
1 Road 1 Connection	211.5	295.0	4	4
1 Road 1 Location	373	1828.7	32	32
1 Road 2 Location	183	1758.3	7	7
2 Locations	486	2574.7	3	3
2 Road	405	566.6	9	9
1 Road	N/A	N/A	0	23
None	N/A	N/A	0	13

Performance of TEG is evaluated against the most commonly used geocoders in traffic event detection research. To present the results of rule-based road geocoder independently from TEER results, performances of TEG, TEGE, Google Maps Geocoding API (GM), ArcGIS World Geocoding Service (AG) and Nominatim (NOM) and Nominatim using *Enriched-OSM* (NOME) are evaluated against the set of events with successfully detected location terms. For 82 events out of 118, TEER task detected all the location terms successfully. Rule-based geocoder processed location entities along with their detected functions, whilst reference geocoders are provided with an address string consisting of location entities complemented with a bounding box of the study area. Terms indicating direction could not be interpreted by GM and AG explicitly and increased errors, therefore removed from corresponding queries.

Median positional errors of geocoding by TEG, GM, AG and NOM are 279.3, 739.1, 3687.6, 3953.8, respectively. Whereas average positional errors of geocoding by TEG, GM, AG, and NOM are 1300.6, 2051.2, 8416.5, 5505.6 meters, respectively. TEG, interpreting location entities with their detected functions, located more events than GM, AG and NOM (65.9% vs. 50.0%, 28.0% and 14.6%) with an under 750-meter positional error. Use of *Enriched-OSM* dataset improved results for both

Traffic Event Geocoder and Nominatim. TEGE performed better than TEG in terms of median position error (217.4 vs 279.3 meters) and located more events within 750-meter error (80.5% vs. 65.9%), similarly NOME achieved smaller median positional errors than NOM (1464.8 vs 3953.8 meters), locating more events within 750-meter error (23.2% vs 14.6%). The geocoding performance results for TEG, TEGE, GM, AG, NOM, and. NOME under successfully detected location terms are presented in Table 6.11. TEG's lead over the reference geocoding services in the number of events geocoded with under 750-m error can be seen in Figure 6.5. GM and AG geocoded all the events, though 3 events geocoded outside of the study area despite a provided bounding box. Events geocoded outside of the network are excluded from averages.

In the TEGE experiments, 6 over 82 events could not be geocoded. Among these, 2 cases failed due to missing or inadequate map data. Geolocating of 2 cases failed due to incorrect location tags assigned by TEER. Geocoder rules failed to handle detected entities in the remaining 2 cases. Being a crowd-sourced data, which is open for editing, OSM can further be improved by defining more landmarks which would assist geocoding in a region of interest. The proposed ruleset can also be extended to improve the geocoding results per need.

Table 6.11 Positional errors in the experiments of Traffic Event Geocoder (TEG), Traffic Event Geocoder with Enriched-OSM (TEGE), Google Maps Geocoding API (GM), ArcGIS World Geocoding Service (AG), Nominatim (NOM) and Nominatim with Enriched-OSM (NOME)

Positional errors	Number of events (#) and percentages (%)											
	TEG		TEGE		GM		AG		NOM		NOME	
(meters)	#	%	#	%	#	%	#	%	#	%	#	%
0-750	54	65.9	66	80.5	41	50.0	23	28.0	12	14.6	19	23.2
750-1500	6	7.3	5	6.1	6	7.3	7	8.5	2	2.4	4	4.9
1500-5000	9	11.0	4	4.9	25	30.5	14	17.1	6	7.3	6	7.3
5000+	6	7.3	1	1.2	9	11.0	36	43.9	15	18.3	16	19.5
N/A	7	8.5	6	7.3	1	1.2	2	2.4	47	57.3	37	45.1

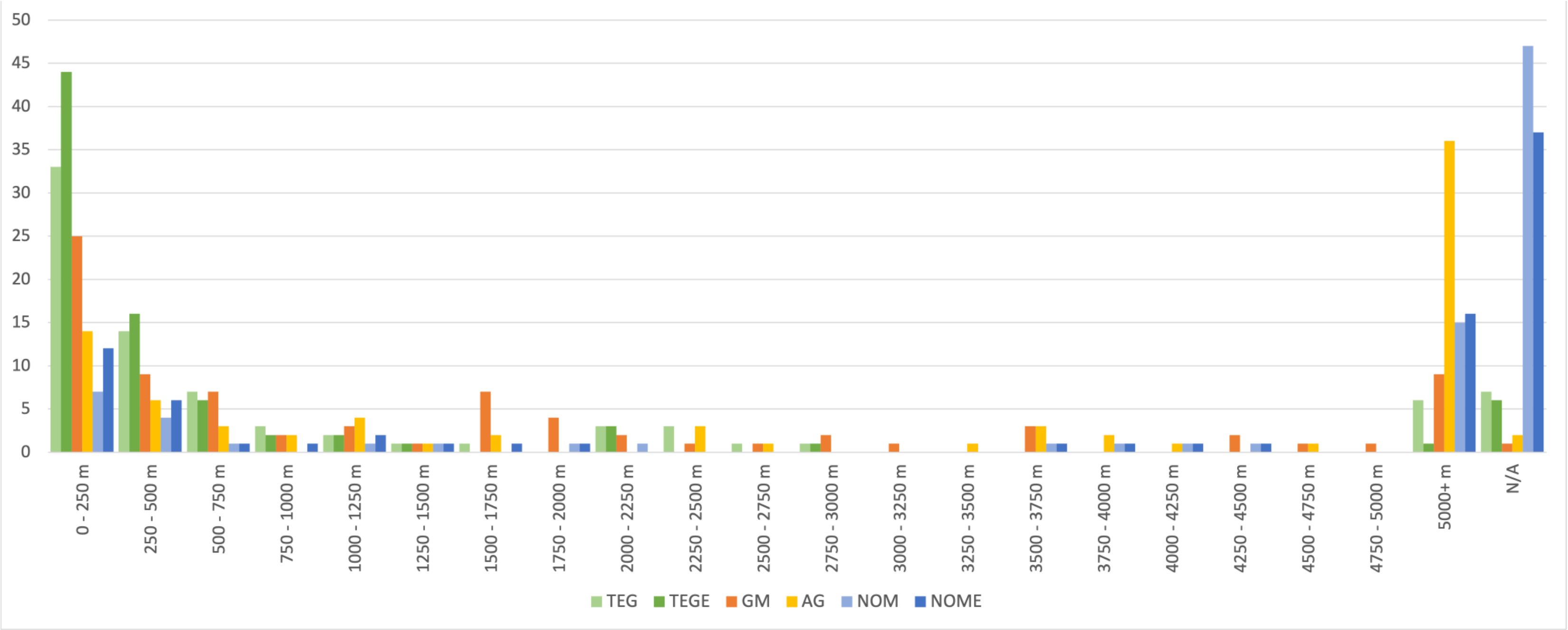


Figure 6.5 Distribution of positional errors in the experiments of Traffic Event Geocoder (TEG), Traffic Event Geocoder with Enriched-OSM (TEGE), Google Maps Geocoding API (GM), ArcGIS World Geocoding Service (AG), Nominatim (NOM) and Nominatim with Enriched-OSM (NOME)
]]>

6.3 Non-recurrent Congestion Detection in FCD

6.3.1 Floating Car Dataset

FCD used for this study is a commercial travel speed data provided by Be-Mobile. FCD data for Turkey is collected from around 600,000 GPS equipped vehicles (Altintasi et al., 2019). Available data covers Eskisehir Road and connecting arterials in Ankara. Data provides 1-minute resolution travel speed data for the road network which is split into segments with a maximum length of 50 meters (Figure 6.6). FCD for the study area consist of 5699 segments covering a total length of 207 kilometers.

In this study, incidents through the year 2017 is investigated. Obtained FCD for the study covers 319 days of 2017, the time periods with missing data are excluded from the analysis. Data consists of over 1,3 billion FCD records taking 22.1 GB disk space. FCD data covers time from 06:00 to 24:00 with 1080 records per day. An FCD record consists of an average travel time and an average travel speed associated with a segment and time-epoch (Table 6.12). Daily distribution of FCD records for an example segment on a major arterial is given in Figure 6.7. Availability of data varies for each segment, while the major arterials has more than 95% daily coverage, some minor arterials has almost no FCD records associated (Figure 6.8). In FCD, road network is represented as unidirectional connected segments, which is converted into an adjacency matrix for further analysis required by the study. In order to handle large FCD data efficiently, data is aggregated into 5-minute time windows. Throughout this section each time window is denoted as an *epoch* and segment-epoch tuples which contain aggregated FCD records of a segment during an epoch is referred as an *FCD frame*. Data is visualized in a 3D environment using Cesium 3D. Time dimension is projected on z-axis and displayed as extruded blocks over link geographical coordinates.

Table 6.12 Data stored in an FCD record

Field	Description
Segment ID	ID of the segment record belongs to
Update Time	Measurement time
Travel Time	Average time taken to traverse the segment
Speed	Average speed observed on the segment

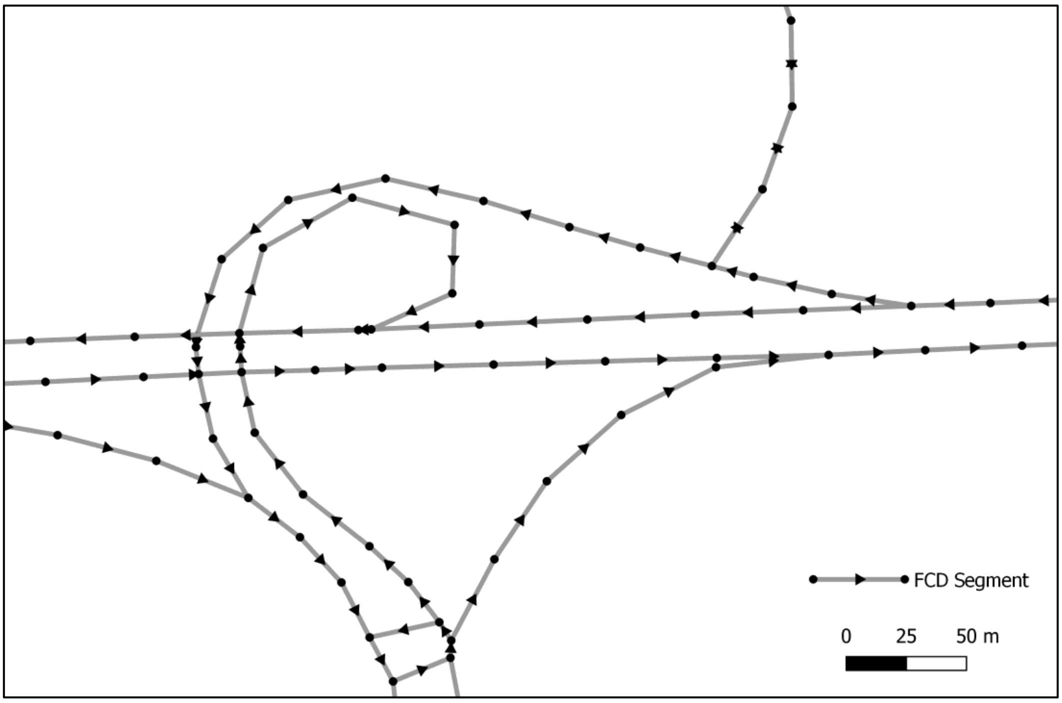


Figure 6.6 Segments of FCD

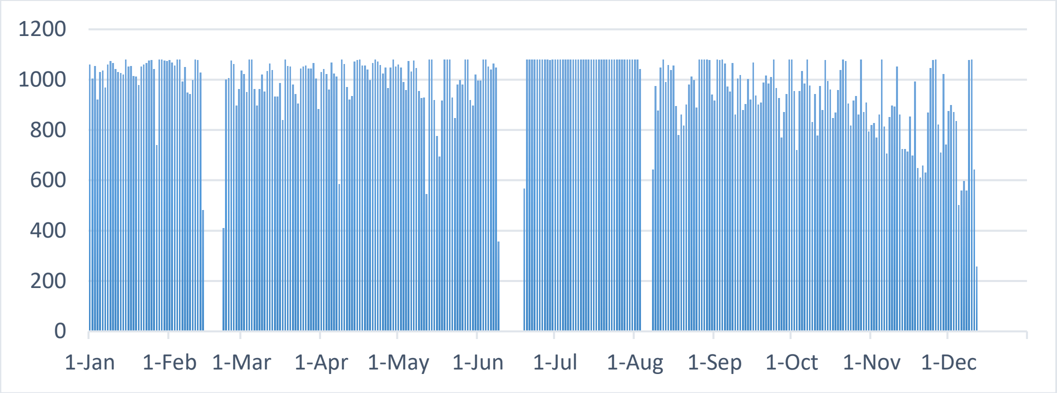


Figure 6.7 Daily number of FCD records for an example segment on Eskisehir Road

6.3.2 Anomaly Detection

In order to evaluate performance of anomaly detection methods, a test data set of recurrent and nonrecurrent congestions are annotated manually (Figure 6.9 and Figure 6.10). Annotations are conducted on a spatiotemporal profile displaying a single flow direction of a road stretch, with segment and epoch axes. FCD frames are displayed with a scaled color code representing observed average speeds. Although there have been some studies to quantify level of congestion in a network using speed thresholds (Long et al., 2008; Sun et al., 2014; Xing et al., 2019), travel speeds in a non-recurrent congestion depend on various variables, such as free flow speed, weather conditions, time and location of the event (Li and Chen, 2013; Zhao et al., 2019). In this study, non-recurrent congestions are identified and annotated using incident tweets with a clear location and impact confirmed in the tweet content. In annotations, no predetermined threshold speed value is used. Incidents without a visually distinguishable spatiotemporal impact area on the speed-based color-coded profile are excluded. In order to confirm non-recurrence, the travel speeds observed on the corresponding segments for same time window of week in previous weeks are checked on corresponding spatiotemporal profiles. Test data set of concurrent congestions are annotated via manual inspection of bottleneck locations on the network by confirming low speeds on a set of segments through same time window of week in the previous weeks. A total of 31 congestions with 10,567 FCD frames are annotated for the tests (Table 6.13).

In the test data set, travel speeds in the FCD frames annotated in a non-recurrent congestion are assumed as anomalous, whilst speeds of those annotated in recurrent congestions are treated as expected. The anomaly detection methods classified observed speeds on FCD frames in the test set as anomalous or not, using various threshold values. A confusion matrix is generated using the comparison of manual annotations and classification results. Performance of the anomaly detection methods are represented using various measures; accuracy, precision, recall and F1-Score.

Table 6.13 Number of recurrent and non-recurrent congestions and number of FCD frames annotated

	Recurrent	Non-recurrent
Congestions	10	21
Number of FCD frames	1880	8687

Table 6.14 Interpretation of classification results with respect to annotations in a confusion matrix

	Annotations	
Classification	Non-recurrently congested	Recurrently congested
Anomalous	True Positive	False Positive
Expected	False Negative	True Negative

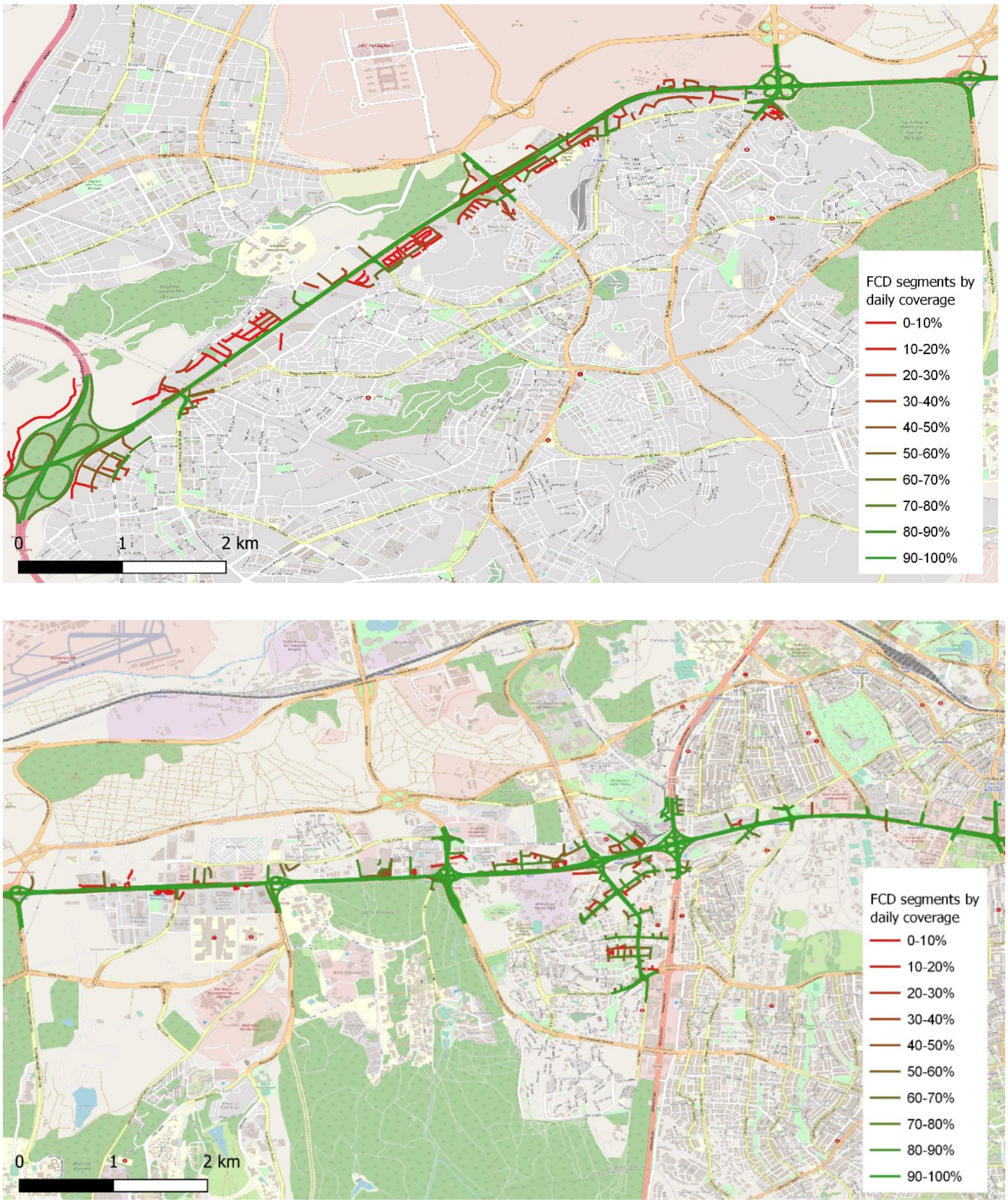


Figure 6.8 Daily average coverage of 1-minute resolution FCD records

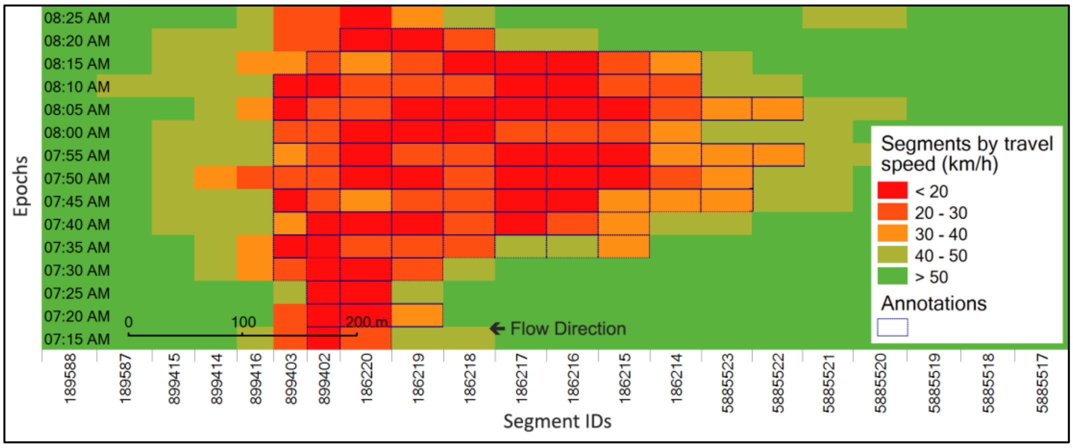


Figure 6.9 An annotated recurrent congestion due to signals at Konutkent Intersection on Eskişehir Road

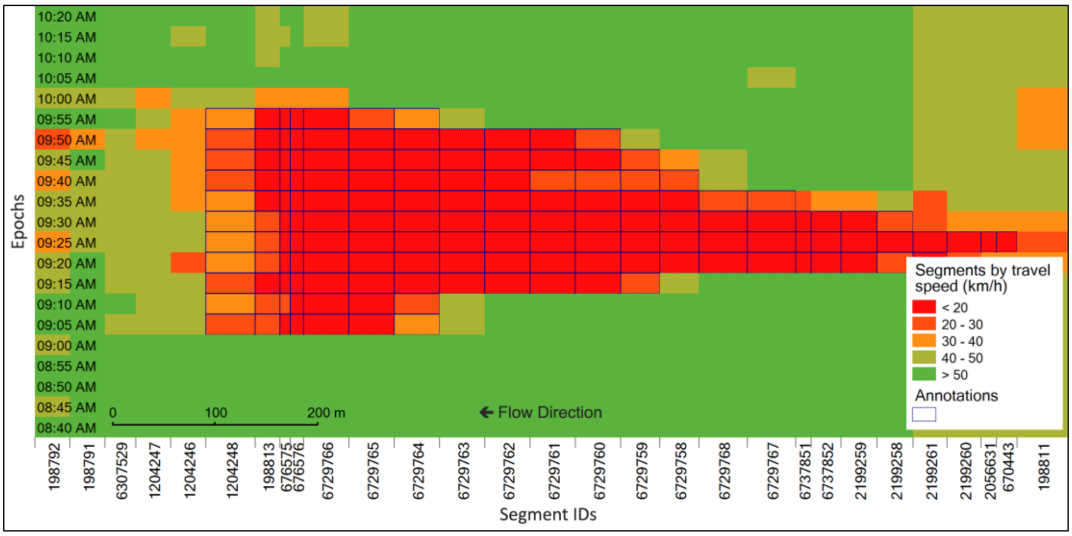


Figure 6.10 An annotated nonrecurrent congestion due to an accident on Eskişehir Road

6.3.2.1 Statistical Methods

6.3.2.1.1 Standard Normal Deviate

Standard Normal Deviate metric is used to detect anomalies over historic link speed data. A time series data consisting of link speed data observed through same window-of-week over a previous number (n) of week is used to detect anomalies. SND is used to classify the speed observed on a link, as anomalous or not by determining a threshold (T) value. Several values of T with various range of length of historic data (n) is tested to maximize F1-Score.

Test results achieved with varying historic data length with optimal T is given in Table 6.15. Tests with 6-week (n=6) historic data yielded highest F1-Scores. Effect of threshold on tests with 6-week historic data is given in Table 6.16.

Table 6.15 SND test results with varying n with the optimal T values.

n	$T\ (\sigma)$	TP	FP	TN	FN	Precision	Recall	F1
4	0.6	8520	673	1207	167	0.927	0.981	0.953
5	0.7	8468	512	1368	219	0.943	0.975	0.959
6	0.7	8509	524	1356	178	0.942	0.98	0.961*
7	0.8	8411	468	1412	276	0.947	0.968	0.957
8	0.8	8404	521	1359	283	0.942	0.967	0.954

Table 6.16 SND test results with by different thresholds (T) with 6-week historic data ($n = 6$)

$T\ (\sigma)$	TP	FP	TN	FN	Precision	Recall	F1
1.0	7964	300	1580	723	0.964	0.917	0.94
0.9	8255	360	1520	432	0.958	0.95	0.954
0.8	8421	436	1444	266	0.951	0.969	0.96
0.7	8509	524	1356	178	0.942	0.98	0.961*
0.6	8560	640	1240	127	0.93	0.985	0.957
0.5	8609	780	1100	78	0.917	0.991	0.953
0.4	8624	883	997	63	0.907	0.993	0.948
0.3	8641	991	889	46	0.897	0.995	0.943

6.3.2.1.2 Median Absolute Deviate

Median Absolute Deviate (MAD) is used to detect anomalous speeds in time-series consisting of historic link speed data in links. MAD is utilized to classify the observed speeds (X) as anomalous or not, by comparing its difference from median normalized by the MAD of the time-series dataset with a threshold (T) value (Equation 6.1). Several values of T with various range of length of historic data (n) is tested to maximize F1-Score.

$$T > \frac{X^c \ - median(X)}{\mathrm{MAD}(X^c)} \tag{6.1}$$

where T is the tested threshold, $X_{1,}X_2, \ldots X_n$ are the observed speeds, X^c is the tested observation.

Tests are carried out using different data lengths and presented with their optimal T's in Table 6.17. Tests with 6-week (n=6) historic data yielded highest F1-Scores. Effect of threshold on tests with 6-week historic data is given in Table 6.18.

Table 6.17 MAD test results with varying n with the optimal T values.

n	T (MAD)	TP	FP	TN	FN	Precision	Recall	F1
4	0.5	7578	653	1227	1109	0.921	0.872	0.896
5	0.5	7728	749	1131	959	0.912	0.89	0.901
6	0.5	8468	671	1209	219	0.927	0.975	0.95
7	0.5	8489	741	1139	198	0.92	0.977	0.948
8	0.6	8374	634	1246	313	0.93	0.964	0.947

Table 6.18 MAD test results with by different thresholds (T) with 6-week historic data ($n = 6$)

$T\ (MAD)$	TP	FP	TN	FN	Precision	Recall	F1
1	6904	276	1604	1783	0.962	0.795	0.871
0.9	7001	303	1577	1686	0.959	0.806	0.876
0.8	7089	337	1543	1598	0.955	0.816	0.88
0.7	7202	385	1495	1485	0.949	0.829	0.885
0.6	8412	632	1248	275	0.93	0.968	0.949
0.5	8468	671	1209	219	0.927	0.975	0.95*
0.4	8498	722	1158	189	0.922	0.978	0.949
0.3	8522	771	1109	165	0.917	0.981	0.948
0.2	8538	807	1073	149	0.914	0.983	0.947
0.1	8561	845	1035	126	0.91	0.985	0.946

6.3.2.1.3 Generalized Extreme Studentized Deviate

Generalized Extreme Studentized Deviate (GESD) is used to detect anomalies using a time-series dataset consisting of a n-week history and a current observation. GESD-based anomaly detection implementation as described in (Luan et al., 2021) is used for tests (Table 6.11). GESD-based classification achieved highest F1-Score using 6-week history data (Table 6.19) with a threshold (c) of 0.5 (Table 6.20).

```
**Inputs**: dataset $x_v^{T,D_n}$; number of iterations $r$, $r < n$; threshold coefficient $c$
**Outputs:** outliers; corresponding thresholds
**For** I in (1:r)
    Minimum index: $minind = which(x = \min(X))$
    Set the parameters: $p = 1 - \frac{c}{(n-i-1)}$
    Calculate the quantile from T distribution: $q = qt(p, ddf = n - i - 1)$
    Calculate the statistics: $T_s = \frac{(n-i)*q}{\sqrt{(n-i-1+q^2)(n-i+1)}}$
    Calculate the threshold: $\tau_{v,GESD}^{T,d} = Mean(X) - T_s \times SD(X)$
    **If** $(X[minind] < threshold)$:
        return $X[minind]$
    Update dataset: $X = X[-minind]$
```

Figure 6.11 GESD-based anomaly detection algorithm (Luan et al., 2021)

Table 6.19 Test results for GESD-based classification over historic data durations (n)

n	Best-c	TP	FP	TN	FN	Precision	Recall	F1
4	0.7	8555	1018	862	132	0.894	0.985	0.937
5	0.5	8414	829	1051	273	0.91	0.969	0.939
6	0.5	8388	750	1130	299	0.918	0.966	0.941
7	0.7	8472	938	942	215	0.9	0.975	0.936
8	1	8496	959	921	191	0.899	0.978	0.937

Table 6.20 Test results for GESD-based classification over different thresholds ($n = 6$)

c	n	TP	FP	TN	FN	Precision	Recall	F1
0.05	6	4957	17	1863	3730	0.997	0.571	0.726
0.1	6	5937	55	1825	2750	0.991	0.683	0.809
0.2	6	7065	246	1634	1622	0.966	0.813	0.883
0.3	6	8020	436	1444	667	0.948	0.923	0.935
0.4	6	8229	600	1280	458	0.932	0.947	0.939
0.5	6	8388	750	1130	299	0.918	0.966	0.941*
0.6	6	8532	931	949	155	0.902	0.982	0.94
0.7	6	8589	1029	851	98	0.893	0.989	0.939
0.8	6	8613	1085	795	74	0.888	0.991	0.937
0.9	6	8646	1189	691	41	0.879	0.995	0.933
1	6	8669	1239	641	18	0.875	0.998	0.932

6.3.2.2 LSTM

A LSTM model is developed for link travel time estimation. FCD dataset covers a date range from November 2016 until May 2018. Data belong to each link is treated as a time series data and split accordingly for model evaluation (Figure 6.12). Due to computational complexity of the models, a 15-minute time-window is used for analysis. Travel time data is aggregated for each link and time-window.

Models are trained using Microsoft Cognitive Toolkit and run on a Nvidia GeForce GTX1660S using CUDA. A dropout rate of %5 is used in the model. Each evaluation is run for 100 iterations using batch sizes of 100. Root mean squared is used as the error function for loss-function. Learning is carried out using a stochastic gradient descent optimizer with a learning rate and momentum. Learning rate is assigned as 0.005 value per sample. Model is trained using data starting from Nov 1St, 2016 to May 30th, 2018, in monthly time series splits. Each model is trained using a 2-month and a 4-month long training dataset (Figure 6.12 and Figure 6.13).

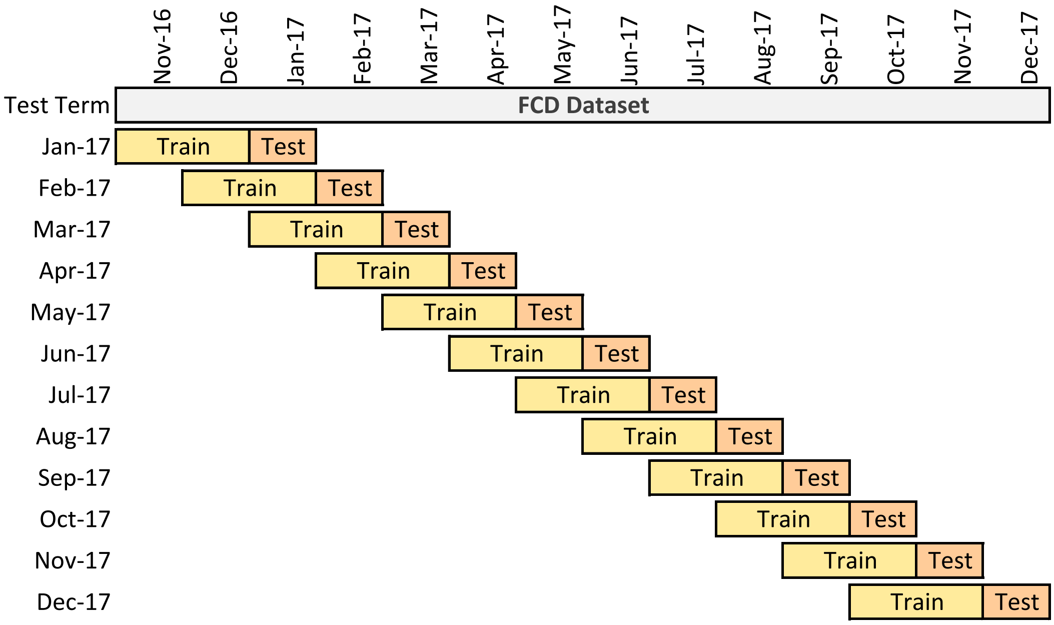


Figure 6.12 Splits when 2-month training data is used for monthly LSTM models

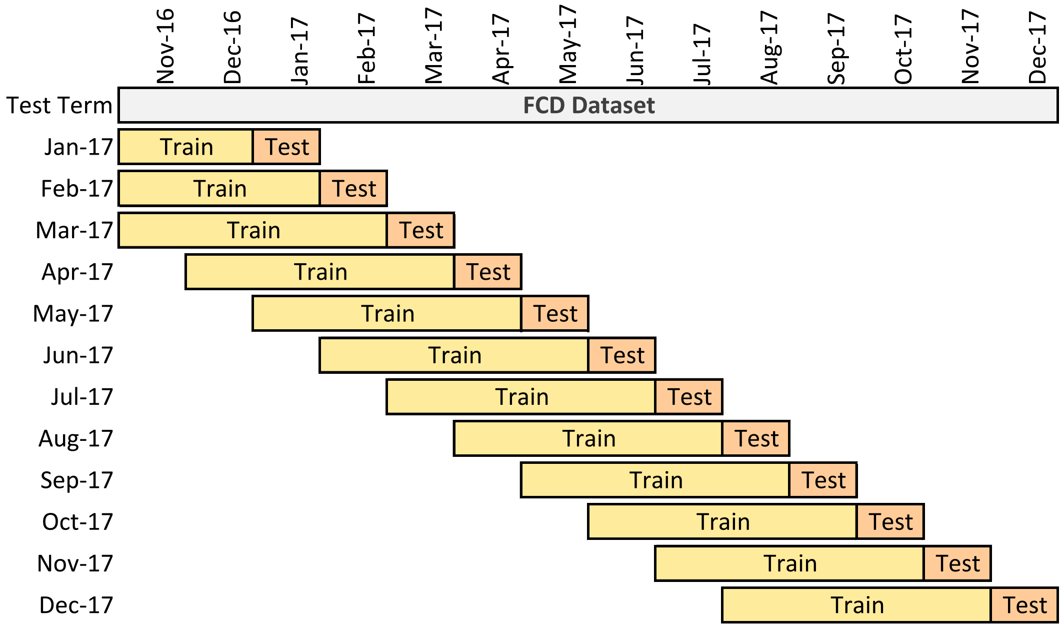


Figure 6.13 Splits when 4-month training data is used for monthly LSTM models

LSTM Models are trained for each link and for various input vectors. Expected traffic speeds for links are estimated using models and compared with the observed values. Estimated and observed speeds are converted into link travel times for comparison. For classification a threshold ratio of observed link travel time with respect to expected link travel time (O/E) is used (Equation 6.2). In classification, O/E values over the threshold are assumed as anomalous.

$$O/E \ = \frac{Observed\ Link\ Travel\ Time}{Estimated\ Link\ Travel\ Time} \tag{6.2}$$

Classification results are presented as a confusion matrix along with precision, recall and F1-Scores. Various O/E thresholds are tested to maximize the F1-Score. Model input types and achieved results are given in their corresponding sections below.

6.3.2.2.1 Time-window of Week

Link travel times have a strong daily and weekly seasonality. In order to model expected flow speeds for time-windows of a week, corresponding time-window index is used as the input. In the LSTM model using the time-window index (LSTM-TWI), a one-hot-encoded vector representing the time-of-day and day-of-week (τ) is used. Since pattern does not differ on midweek days, Tuesday, Wednesday, and Thursday were merged and represented as one day to decrease complexity, decreasing the number of days per week to 5. The day dimension is modelled hourly starting from 6 AM until 23 PM. Size of the input vector, which is the one-hot encoding representation of 18 hours in 5 days, is 90.

Test result metrics for the classification under various threshold O/E values are given in Table 6.21 and Table 6.22, for models trained with 2-month and 4-month data respectively. Model achieved highest F1-Score with a threshold O/E value of 1.6. Increasing the amount of training data to 4-months did not improve the classification performance.

Table 6.21 Test results for classification with LSTM-TWI (2-month training data)

Threshold O/E	TP	FP	TN	FN	Precision	Recall	F1
1.1	8662	1470	410	25	0.855	0.997	0.921
1.2	8652	1362	518	35	0.864	0.996	0.925
1.3	8633	1259	621	54	0.873	0.994	0.93
1.4	8606	1144	736	81	0.883	0.991	0.934
1.5	8553	1021	859	134	0.893	0.985	0.937
1.6	8478	902	978	209	0.904	0.976	0.939*
1.7	8356	804	1076	331	0.912	0.962	0.936

Table 6.22 Test results for classification with LSTM-TWI (4-month training data)

Threshold O/E	TP	FP	TN	FN	Precision	Recall	F1
1.1	8661	1470	410	26	0.855	0.997	0.921
1.2	8650	1367	513	37	0.864	0.996	0.925
1.3	8633	1262	618	54	0.872	0.994	0.929
1.4	8604	1154	726	83	0.882	0.99	0.933
1.5	8553	1048	832	134	0.891	0.985	0.936
1.6	8488	938	942	199	0.9	0.977	0.937*
1.7	8371	859	1021	316	0.907	0.964	0.935

6.3.2.2.2 Network State

In the LSTM model based on network state (LSTM-NS) average flow speeds observed on the road network at t-1 is used as the input for the model. For a better representation, inbound and outbound segments of the network, which produce different demand patterns through the day, are modelled separately with average speeds observed on the segments which have the same flow direction. To represent the network average, a sample link set is created by selecting every third link though each street flow direction.

Test results for classification under various threshold values are given in Table 6.23 and Table 6.24, for models trained with 2-month and 4-month data respectively. Model achieved highest F1-Score with a threshold O/E value where observed travel speed is 50% higher than the estimated travel speed. Increasing the amount of training data from 2-months to 4-months did not change the performance of the classification.

Table 6.23 Test results for classification with LSTM-NS (2-month training data)

Threshold O/E	TP	FP	TN	FN	Precision	Recall	F1
1.1	8651	1390	490	36	0.862	0.996	0.924
1.2	8636	1233	647	51	0.875	0.994	0.931
1.3	8593	1063	817	94	0.89	0.989	0.937
1.4	8529	915	965	158	0.903	0.982	0.941
1.5	8428	771	1109	259	0.916	0.97	0.942*
1.6	8308	642	1238	379	0.928	0.956	0.942*
1.7	8160	513	1367	527	0.941	0.939	0.94

Table 6.24 Test results for classification with LSTM-NS (4-month training data)

Threshold O/E	TP	FP	TN	FN	Precision	Recall	F1
1.1	8645	1408	472	42	0.86	0.995	0.923
1.2	8627	1263	617	60	0.872	0.993	0.929
1.3	8585	1099	781	102	0.887	0.988	0.935
1.4	8518	937	943	169	0.901	0.981	0.939
1.5	8418	793	1087	269	0.914	0.969	0.941*
1.6	8287	672	1208	400	0.925	0.954	0.939
1.7	8147	544	1336	540	0.937	0.938	0.937

6.3.2.2.3 Previous Link State

FCD dataset aggregated for LSTM model consist of 15-minute epochs on each link. As a baseline to LSTM-TWI and LSTM-NS, an LSTM model using flow speed observed in t-1 (LSTM-LS) is developed.

Classifier based on model estimation is tested using the test data set. Results are given in Table 6.25 and Table 6.26 for models trained with 2-month and 4-month data respectively. Model achieved highest F1-Score with 1.7 O/E threshold value. Increasing the amount of training data from 2-months to 4-months did not change the performance of the classification.

Table 6.25 Test results for classification with LSTM using Previous Link State input (2-month training data)

Threshold O/E	TP	FP	TN	FN	Precision	Recall	F1
1.1	8668	1568	312	19	0.847	0.998	0.9163
1.2	8659	1497	383	28	0.853	0.997	0.9194
1.3	8647	1420	460	40	0.859	0.995	0.922
1.4	8629	1349	531	58	0.865	0.993	0.9246
1.5	8608	1274	606	79	0.871	0.991	0.9271
1.6	8572	1207	673	115	0.877	0.987	0.9288
1.7	8523	1145	735	164	0.882	0.981	0.9289*
1.8	8455	1068	812	232	0.888	0.973	0.9286
1.9	8361	1004	876	326	0.893	0.962	0.9262

Table 6.26 Test results for classification with LSTM using Previous Link State input (4-month training data)

Threshold O/E	TP	FP	TN	FN	Precision	Recall	F1
1.1	8668	1552	328	19	0.848	0.998	0.9169
1.2	8657	1462	418	30	0.856	0.997	0.9211
1.3	8647	1393	487	40	0.861	0.995	0.9232
1.4	8628	1318	562	59	0.867	0.993	0.9257
1.5	8605	1245	635	82	0.874	0.991	0.9288
1.6	8571	1168	712	116	0.88	0.987	0.9304
1.7	8521	1097	783	166	0.886	0.981	0.9311*
1.8	8453	1017	863	234	0.893	0.973	0.9313
1.9	8361	1004	876	326	0.893	0.962	0.9262

6.3.2.3 Road Network-based Model

A classifier has been developed using the Road Network-based Model given in Section 5.1.3. Links are annotated with the direction information (d), *inbound*, *outbound* or *none*, to calculate network state ratio ($r_{d,\tau}$) for a time-window of interest (τ). Classification is carried out using threshold ratio of observed link travel time with respect to expected link travel time (O/E). Performance of classification is tested using different historic data lengths (n) and various O/E threshold levels. Best F1-Score is achieved when estimation is done using a 4-week link travel speed history is used (Table 6.27). Classifier performed best with a threshold O/E value of 1.5 (Table 6.28).

Table 6.27 Test results for classification using Road Network-based Model with respect to n

n	Best Thr. O/E	TP	FP	TN	FN	Precision	Recall	F1
3	1.4	8386	318	1562	301	0.963	0.965	0.964
4	1.5	8400	244	1636	287	0.972	0.967	0.969*
5	1.5	8402	408	1472	285	0.954	0.967	0.960
6	1.5	8455	424	1456	232	0.952	0.973	0.962
7	1.5	8456	490	1390	231	0.945	0.973	0.958
8	1.5	8456	551	1329	231	0.939	0.973	0.955

Table 6.28 Test results for classification using Road Network-based Model, $n = 4$

Threshold O/E	TP	FP	TN	FN	Precision	Recall	F1
1.1	8620	812	1068	67	0.914	0.992	0.9514
1.2	8583	604	1276	104	0.934	0.988	0.9602
1.3	8543	450	1430	144	0.95	0.983	0.9662
1.4	8483	332	1548	204	0.962	0.977	0.9694
1.5	8400	244	1636	287	0.972	0.967	0.9695*
1.6	8283	185	1695	404	0.978	0.953	0.9653
1.7	8163	146	1734	524	0.982	0.94	0.9605
1.8	8008	116	1764	679	0.986	0.922	0.9529
1.9	7829	84	1796	858	0.989	0.901	0.943
2.0	7679	68	1812	1008	0.991	0.884	0.9344

6.3.2.4 Discussion of Anomaly Detection Results

Detecting anomalous flow speeds on the links is the first step of the proposed non-recurrent congestion detection methods. Various anomaly detection methods which are commonly used in the literature is evaluated along with a new method, Road Network-based Model, based on the spatiotemporal nature of the FCD data. Performance of the methods are evaluated using a manually annotated data set of recurrent and non-recurrent congestions. Results are compared using F1-Scores calculated using a confusion matrix (Table 6.29). Road Network-based Model achieved highest F1-Score of 0.969, closely followed by SND-based anomaly detection method. Road Network-based Model achieved highest score using the past 4-week historic data, whereas SND used 6-week historic data for best results. MAD-based anomaly detection method has a 0.95 F1-Score, outperforming GESD-based method which achieved an F1-Score of 0.941. Performance of LSTM-based models are inferior to SND and MAD-based methods and Road Network-based Model. Even F1-Scores of LSTM models are on par with the evaluated statistical methods, high computational costs of LSTM models make statistical methods and the proposed Road Network-based Model less costly and robust choices to detect anomalies in an FCD dataset with a weekly repeating pattern. Anomaly Factor metric which quantifies the level of anomaly is calculated for each FCD frame using O/E parameter of Road Network-based Model. The threshold AF value (T_{AF}) which is used to classify FCD frames as anomalous or not is determined as the threshold O/E value with achieved highest F1-Score in the tests.

6.3.3 Incident Related Congestion Identification

Proposed congestion front detection method is based on supervised learning. In this step non-recurrently congested FCD frames, detected using Network-based Model (see Section 6.3.2.3), are classified into two classes (see Section 5.2.2):

- Incident related congestion fronts (ICF)

- Non-incident related congestion fronts and congestion upstream segments for all NRCs (Other)

A set of accidents with a clearly identifiable impact region on FCD are selected from the accident log and incident tweets. FCD frames which form the congestion fronts of the accidents are annotated manually as *ICF* (Figure 6.14). The days on which network links observe lower average travel speeds are identified as network-wide congested days. Congestion fronts from bottleneck locations on network-wide congested days are identified as non-incident related congestion front and annotated as *Other* (Figure 6.15). Some sample upstream links of both incident and non-incident related congestion FCD frames are also annotated as *Other*. For training set, 783 non-recurrently congested segments are annotated manually with the corresponding class (Table 6.30).

Table 6.29 Overall comparison of performances of anomaly detection methods

Model Name	Parameters	TP	FP	TN	FN	Precision	Recall	F1
SND	$n = 6$ $\sigma =$ 0.7	8509	524	1356	178	0.942	0.98	0.961
GESD	$n = 6$ c = 0.5	8388	750	1130	299	0.918	0.966	0.941
MAD	$n = 6$ MAD = 0.5	8468	671	1209	219	0.927	0.975	0.95
LSTM-TWI	O/E = 1.6	8478	902	978	209	0.904	0.976	0.939
LSTM-LS	O/E=1.5	8428	771	1109	259	0.916	0.97	0.942
LSTM-NS	O/E=1.8	8453	1017	863	234	0.893	0.973	0.9313
Road Network-based Model	$n = 4$ O/E=1.5	8400	244	1636	287	0.972	0.967	0.9695*

Table 6.30 FCD Segment annotations for congestion front detection

Class	Description	Number of annotations
ICF	Incident related congestion fronts	169
Other	Non-incident related congestion fronts	383
	Congestion upstream	231

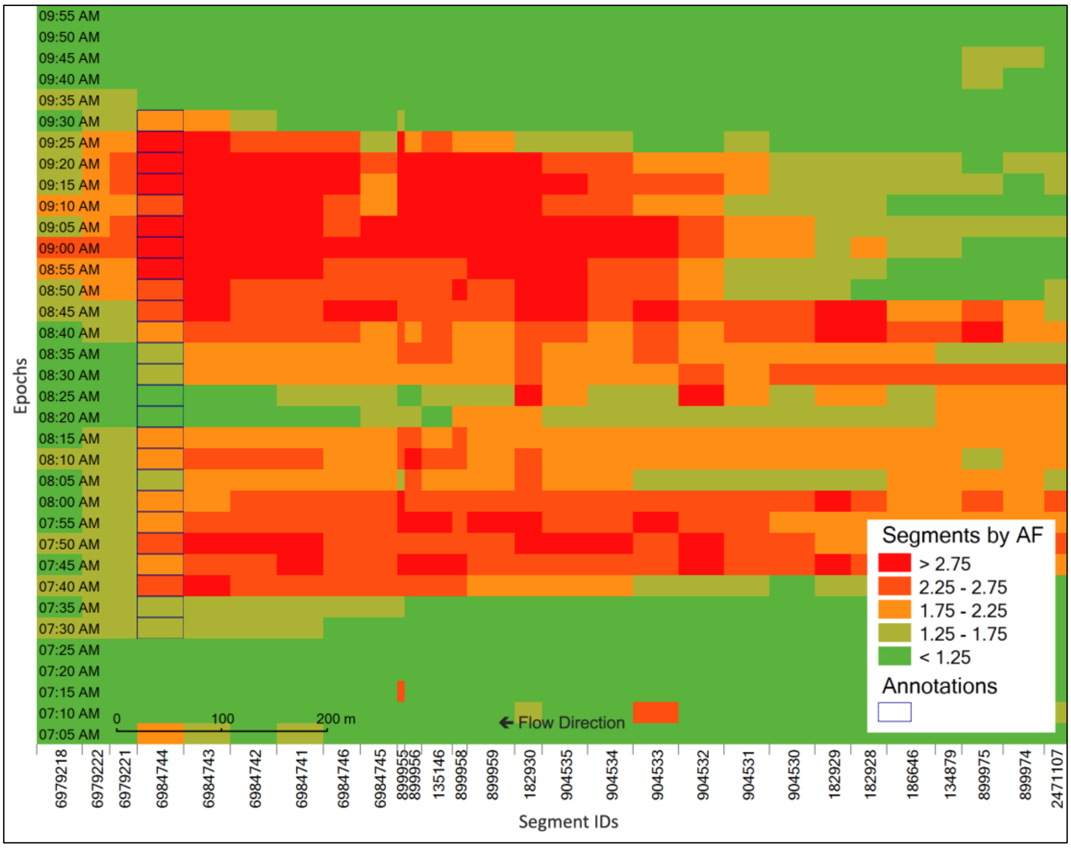


Figure 6.14 Incident related congestion front annotation example for on inbound Eskisehir Road

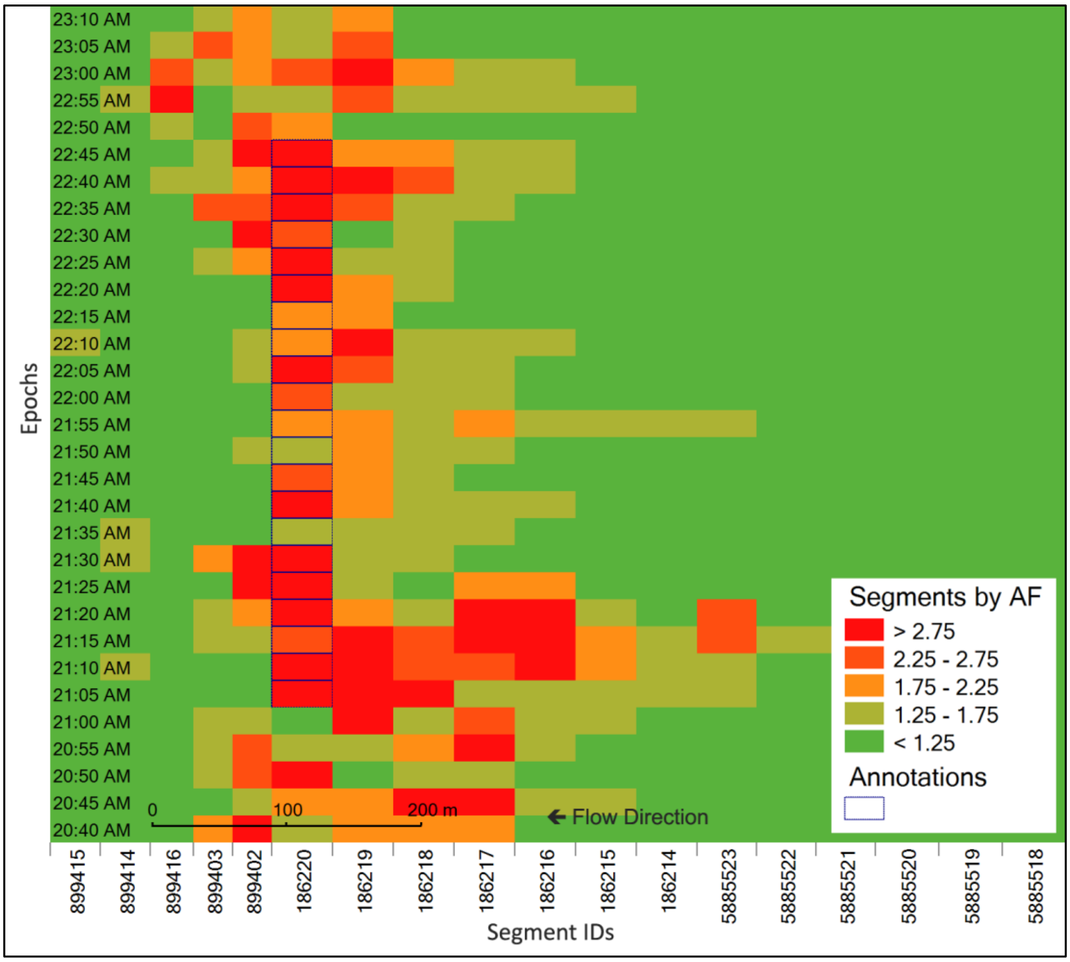


Figure 6.15 A signal induced congestion front annotation example on inbound Eskisehir Road

Classification model is developed using decision tree and support vector machine-based learning models. Input vector for classification model consists of three dimensions (Section 5.2.2)

- Upstream/downstream speed difference: $\Delta V_{s,t}^{u,d}$
- Upstream/downstream speed difference variate: $SND_{s,t}^{u,d,h}$
- Downstream anomaly factor: $AF_{s,t}^{d}$

  Where s denote to segment of interest on epoch t.

  u is subgraph depth consisting of upstream segments of segment s, and d is the number of downstream segments. h is the number of datapoints used to calculate the standard deviate. In the case study last 14 days are used to calculate the SND.

Performance of the models are evaluated using a 5-fold cross validation method, in which data is split using day number of a year. Optimum values for u and d are found for each input vector using a grid-search. Parameters describing number of segments to be used to calculate input values are given in Table 6.31.

A decision tree-based (DT) and a support vector classifier (SVM) is evaluated for congestion front classification. A decision tree classifier based on C4.5 Algorithm is used to carry out classification tests. A grid search seeking for the optimum number of segments to be used to calculate input parameters is performed using values between 0 to 15 (0 to ~525 meters). Each test is performed as a 5-fold cross validation test. DT model using parameters u_{ND} = 10, d_{ND} = 5, u_{SND} = 1, d_{SND} = 10, d_{AF} = 2 performed best with an F1-Score of 81.9%. SVM model achieved highest F1-Score of 74.1% using parameters u_{ND} = 8, d_{ND} = 4, u_{SND} = 8, d_{SND} = 8, d_{AF} = 1.

Table 6.31 Parameters of input values for classification model

Name	Description
u_{ND}	Depth of the upstream subgraph (u), constituting the upstream of a segment, average speeds of which are used to calculate Speed Difference
d_{ND}	Length of downstream of a segment used to calculate Speed Difference (d)
u_{SND}	Depth of upstream subgraph (u), constituting the upstream of a segment, average speeds of which are used to calculate Speed Difference Deviate
d_{SND}	Length of downstream of a segment used to calculate Speed Difference Deviate (d)
d_{AF}	Length of downstream of a segment used to calculate Downstream Anomaly Factor (d)

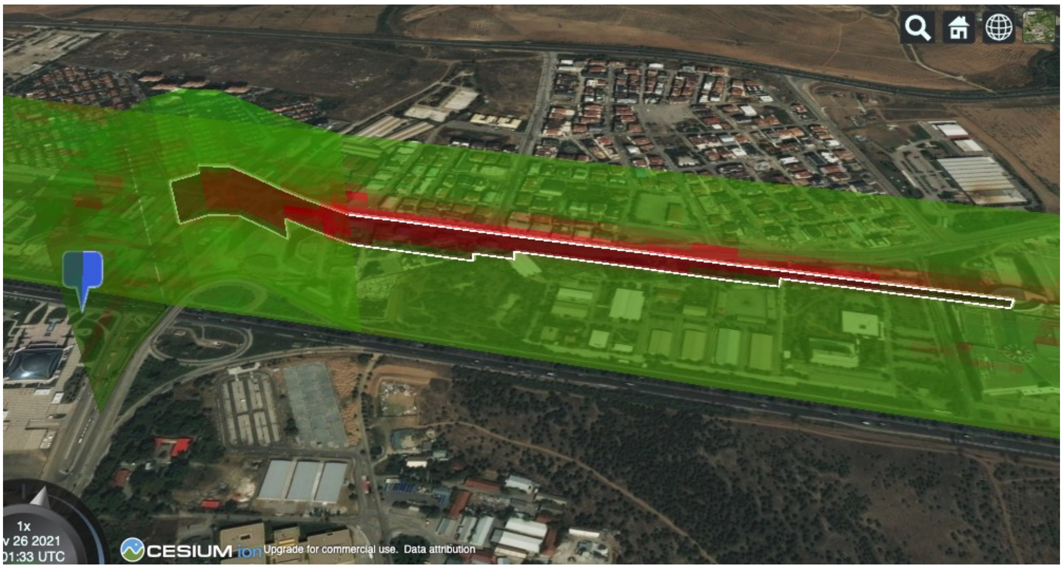


Figure 6.16 Identification Congestion at Bilkent Intersection on Eskisehir Road

Large number of incidents detected with a 5-minute (1 epoch) duration may be an indicator of short-lived fluctuations in speed data caused by non-incident related factors, such as signals or bottleneck locations. When detected NRCs are explored on the network, most of the congestions with a 1-epoch duration are a result of misclassified non-recurrent congestions fronts which took place usually in bottleneck locations, mostly during rush hours (Figure 6.17 and Figure 6.18). However, duration of misdetected NRCs in signalized intersections might extend to multiple epochs (Figure 6.19). In such cases anomaly detection classified travel speed in such segments as anomalous and signal location is also classified as a congestion front, but spatial extent of such congestions is limited. Thus, a timespan threshold along with a spatial extent threshold could be used together to minimize the number of misdetections.

Table 6.32 Number of detected NRCs by their spatial and temporal impact extent

		Timespan							
		≥ 5 min.	≥ 10 min.	≥ 15 min.	≥ 20 min.	≥ 25 min.	≥ 30 min.	≥ 45 min.	≥ 60 min.
Spatial extent	≥ 0 m.	130592	6052	2392	1560	1131	874	418	250
	≥ 100 m.	37791	5864	2376	1557	1129	873	418	250
	≥ 200 m.	14364	4285	2181	1515	1119	867	417	250
	≥ 300 m.	8757	3365	1969	1433	1088	850	416	250
	≥ 400 m.	6555	2819	1784	1345	1043	825	411	247
	≥ 500 m.	5264	2460	1646	1274	994	792	399	241
	≥ 750 m.	3262	1759	1273	1026	819	668	351	220
	≥ 1000 m.	2084	1264	983	819	671	554	304	193
	≥ 1500 m.	936	642	542	462	404	344	198	124
	≥ 2000 m.	453	334	294	257	232	205	130	82
	≥ 2500 m.	272	206	184	168	152	132	84	59
	≥ 5000 m.	21	16	14	13	12	12	9	9

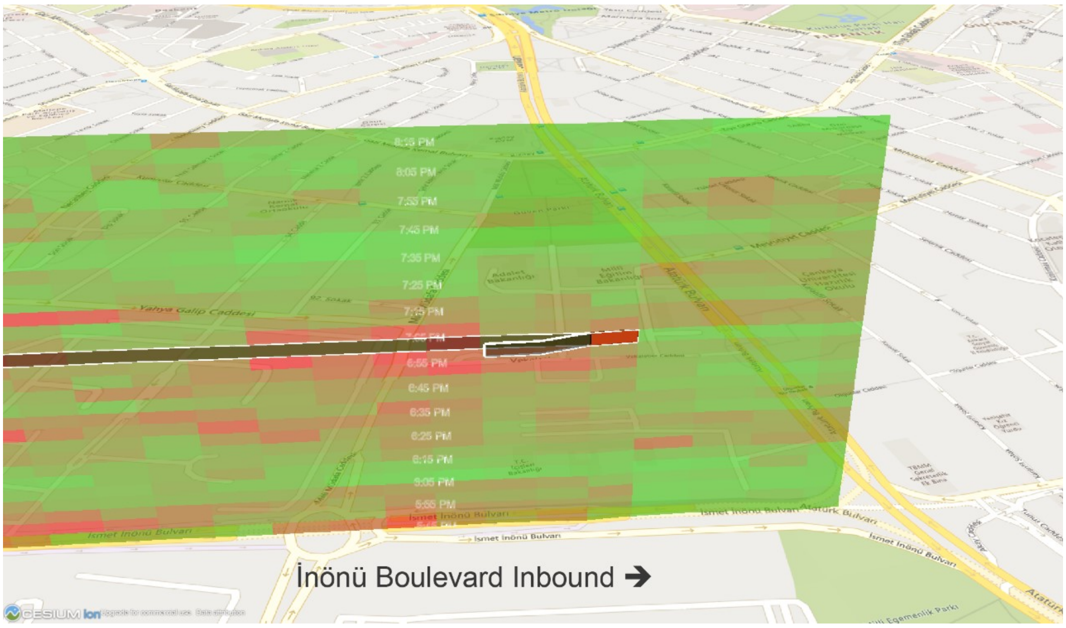


Figure 6.17 A misdetected NRC on a bottleneck location, at an underpass entrance in İnönü Boulevard

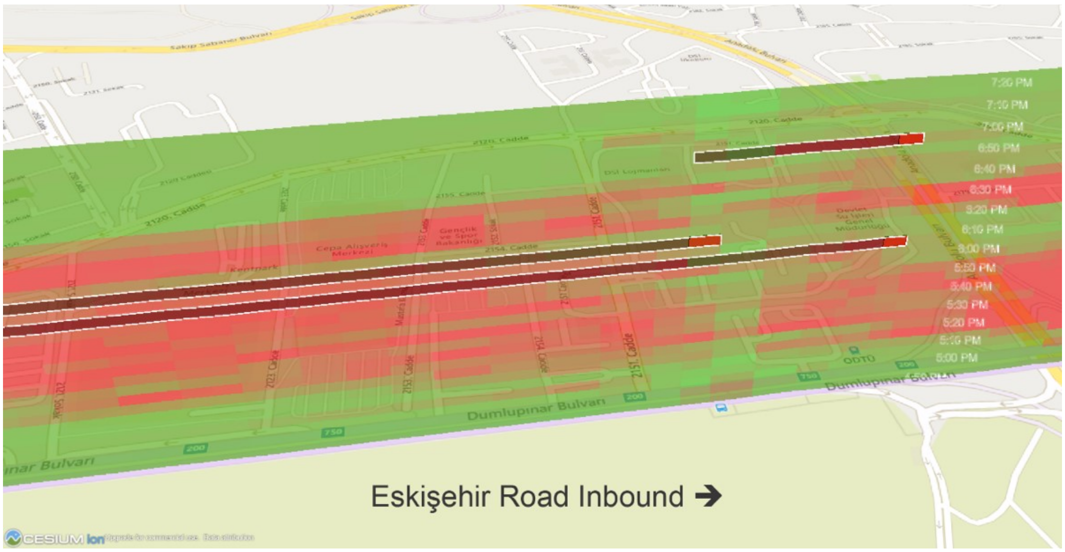


Figure 6.18 Misdetected NRCs at the intersection of Eskisehir Road and Anadolu Boulevard.

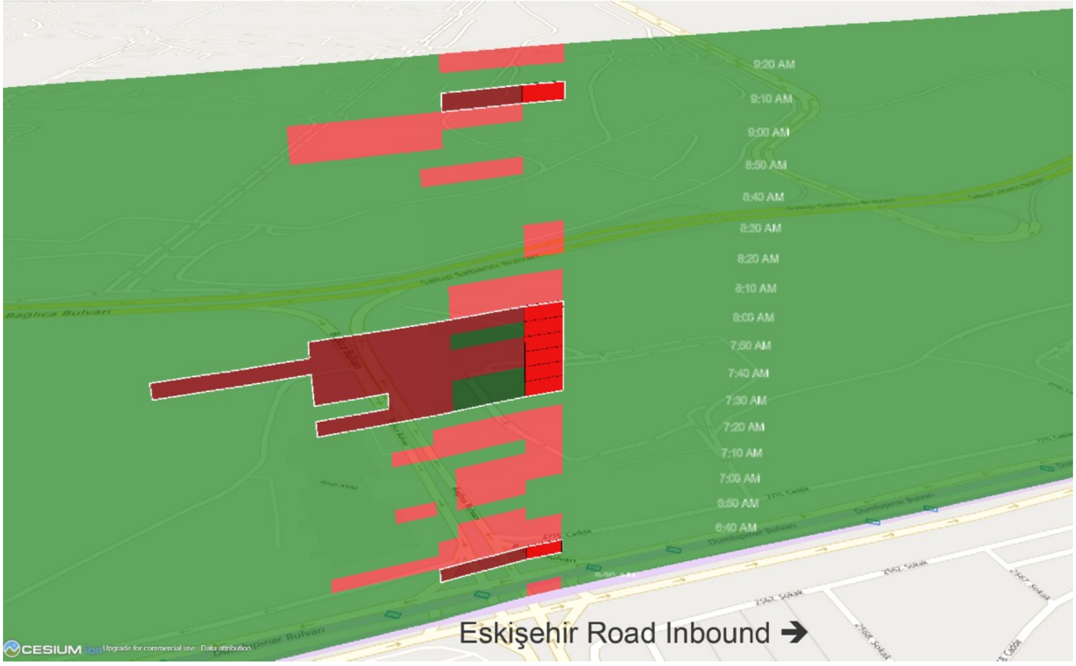


Figure 6.19 Segments with anomalous travel speed in a signalized intersection misdetected as NRC

6.4 Spatiotemporal Information Matching

In this section proposed spatiotemporal information matching method (Section 5.3) is evaluated using congestions detected in Section 6.3.3 and incidents detected in social media. Traffic event tweets classified by SVM classifier are geocoded using Traffic Event Geocoder. While spatial extend of traffic event detection carried out on SMD is Ankara proper area, FCD covers only Eskişehir Road proximity. A geographical extend including Eskişehir Road and its intersections with connecting roads is determined for matching traffic events with NRC. Events impact area of which overflow extends of FCD are eliminated due to missing coverage for congestion analysis. Events which are mislocated by word similarity to the study area extend are removed manually. A total of 161 traffic incident related posts fell within the extend of the FCD. Some traffic incident related posts examples are given by type in Table 6.33 and number of traffic related posts by type is given in Table 6.34.

Traffic incident related posts are matched with detected NRCs using spatiotemporal information matching method using score parameters given in Table 6.35. In order to calculate match confidence estimates, p-values of delta-scores among match scores are calculated. A distribution of delta-score is calculated using simulation. To this aim, 1000 incidents are randomly generated on Eskisehir Road, for time windows starting from 07:00 AM until 21:00 PM all year around in 2017. A distribution of simulated delta-scores is created as a reference to calculate p-values (Figure 6.20). Delta scores are calculated for each available match.

Due to geocoding errors and report time uncertainty, a buffer should be defined in time and distance to match incidents. Match confidence estimate (C) is calculated based on the p-value of corresponding delta score (Section 5.3.1). A match is considered a confident match if C is below 0.05. Various spatial buffer sizes (from 0 to 1,000 meters) and time buffer sizes (from 15 to 60 minutes) are tested in order to explore percentage of matches and confident matches in the corresponding buffer sizes.

Table 6.33 Example traffic related posts by type

Type	Tweet Content
Accident	*Eskişehir yolu konutkent kavşağı... çok ciddi kaza can kaybı olabilir 4 araba birbirine girmiş durumda* Konutkent intersection on Eskişehir Road, a serious accident with possible causalities, 4-vehicle pile-up crash
Accident	*Eskişehir yolu Ortadoğu Üniversitesi kavşağı kazası @radyotrafik06* Accident at Middle East (Technical) University Intersection on Eskişehir Road @radyotrafik06
Breakdown	*@radyotrafik06 Eskişehir yolundan Anadolu Bulvarı'na girişte sağda Tır arızalanmış! Biraz etkileyecektir muhakkak ki..* A truck broke down on the right at the entrance to Anadolu Boulevard from the Eskişehir road! It will probably affect a little.
Breakdown	*@radyotrafik06 eskişehir yolu merkez yönde gordion köprüsünü geçince sağdan ikinci şeritte kamyon arızası duruyor.* Truck malfunction in the second lane from the right, where inbound Eskişehir Road crosses the Gordion Bridge
Traffic State	*Eskişehir yolu Eskişehir istikametine doğru. ATO dan Bilkent kavşağına 10 dakika da geldik. Trafik dur kalk ile ile…* Eskişehir Road towards Eskişehir direction. We came to Bilkent junction from ATO in 10 minutes. With a stop-and-go traffic…
Traffic State	*Eskişehir yolu > kizilay istikameti medicana önünde 10 dakikadır duruyor, ilerlemiyor @radyotrafik06* Eskişehir road > kizilay direction, stalled in front of the medicana for 10 minutes, not moving @radyotrafik06
Other Information	*Eskişehir Yolu Bilkent Şehir Hastaneleri Atık Su Yağmursuyu hatları yapım çalışmaları nedeni ile Eskişehir yönünden gelip Bilkent…* Due to construction work on Bilkent Şehir Hospital wastewater pipelines, coming from Eskişehir direction to Bilkent…
Unknown	*Eskişehir yolu biliyorum Ümitköy köprüsünü geçtikten sonra Kızılay yönünde* I know it's Eskişehir Road, on Kızılar direction after passing Ümitköy overpass

Table 6.34 Traffic related posts in study area by type

Type	Number of posts
Accident	47
Breakdown	6
Traffic State	93
Other Information	12
Unknown	1
Total	161

Table 6.35 Scores used in traffic-event-NRC matching

Score Name	Values	Score threshold	Weight
Spatial Distance Score	1-0	2 km	1
Time Distance Score	1-0	60 minutes	1
Impact Size Score	1-0	-	1
Street Match Score	1 or 0	-	0.5

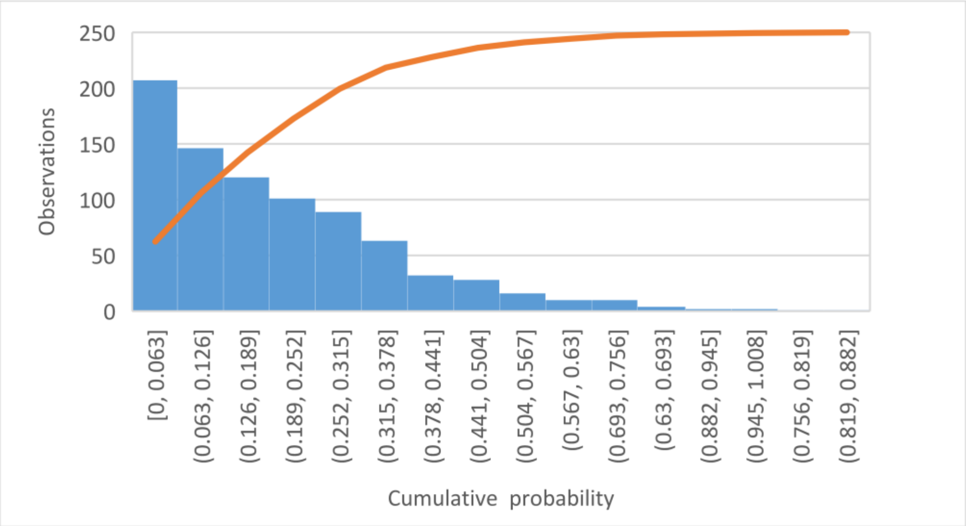


Figure 6.20 Distribution of simulated delta-scores

Matching is performed for NRCs with various timespan thresholds. Percentage of posts matched with a NRC depended highly on the size of buffer, reaching 81.1% percent with a 750-meter by 30-minute buffer, and 83% with a 1000-meter by 45-minute buffer. Whereas percentage of confident matches reached highest value of 47.2% using a 750-meter by 15-minute buffer (Figure 6.21). Low percentage of confident matches could be attributed to misdetected NRCs, which are the small, congested regions caused by operational features of the urban network. In order to remove noise than can mimic incidents, two thresholds are applied to the detected set of NRCs: Time span and spatial extend thresholds. Time span is the total impact duration of an incident in minutes, while spatial extend is the total length of segments impacted from the congestion throughout the time span.

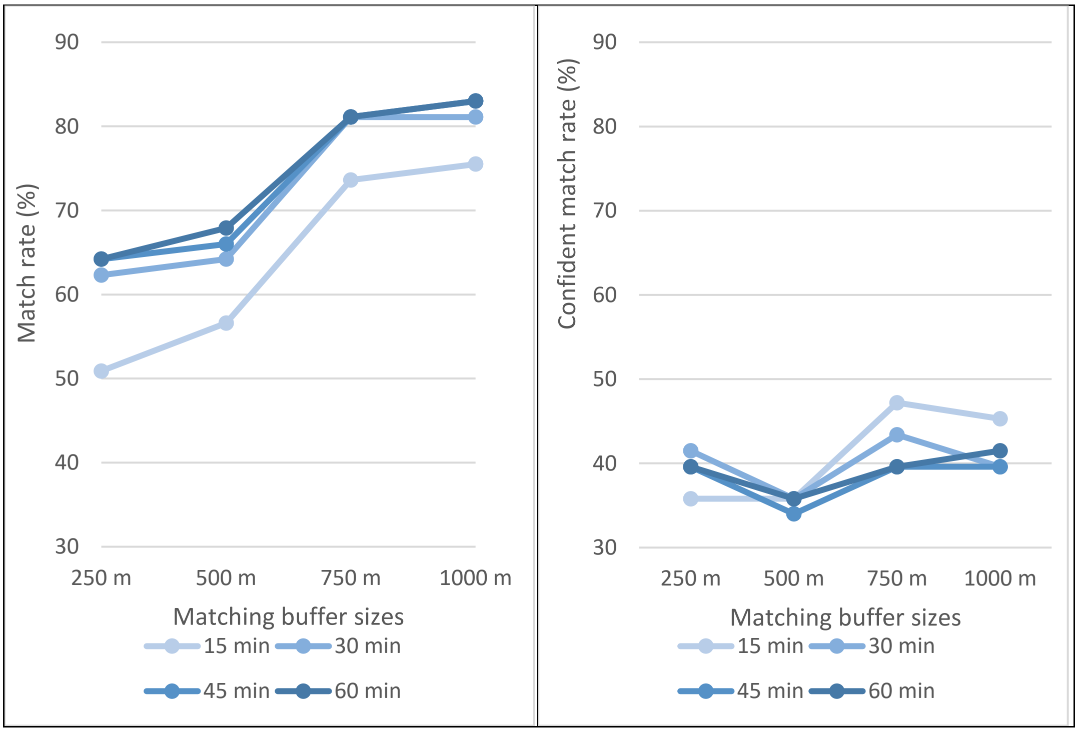


Figure 6.21 Percentage of accident and breakdown posts matched by an NRC by various temporal and spatial buffer sizes

To determine the thresholds to remove noise, matching rates are explored for a subset of accident and breakdown posts, which are expected to have an impact on traffic flow. Subset of accidents and breakdowns are consisted of 53 events. Events are matched with the NRCs using various time spans (5, 10, 15 and 20 minutes) and spatial extends (from 0 to 500 meters), using a 750-meter spatial by 30-minute and 100-meter by 45-minute match buffer. Percentage of posts matching with an NRC decreased with the increasing timespan and spatial extent thresholds applied to NRCs (Figure 6.22), though, when only confident matches are considered, a 10-minute timespan threshold along with an up to 300-meter spatial extent yielded the highest match rate of 54.7% (Figure 6.22a) with a 750-meter by 30-minute buffer and 58.5% with a 1000-meter by 45-minute buffer (Figure 6.22b). Number of detected NRCs dropped from 130,592 to 3,365 for year 2017 with a 10-minute timespan and 300-meter spatial extent thresholds (Figure 6.23). When NRCs are filtered by a 10-minute timespan and 300-meter spatial extent threshold, percentage of posts matched with an NRC dropped from 81.1% to 64.2% using a 750-meter by 30-minute matching buffer, and from 83% to 69.8% with a 1000-meter by 45-minute matching buffer. Whereas percentage of confident matches increased from 43.4% to 54.7% using a 750-meter by 30-minute buffer, and from 39.6% to 58.5%, using a 1000-meter by 45-minute buffer (Figure 6.21 and Figure 6.24). It should be noted that a timespan threshold would increase mean time to detect incidents, as the proposed method would ignore any NRCs until its duration reaches to the determined minimum timespan threshold.

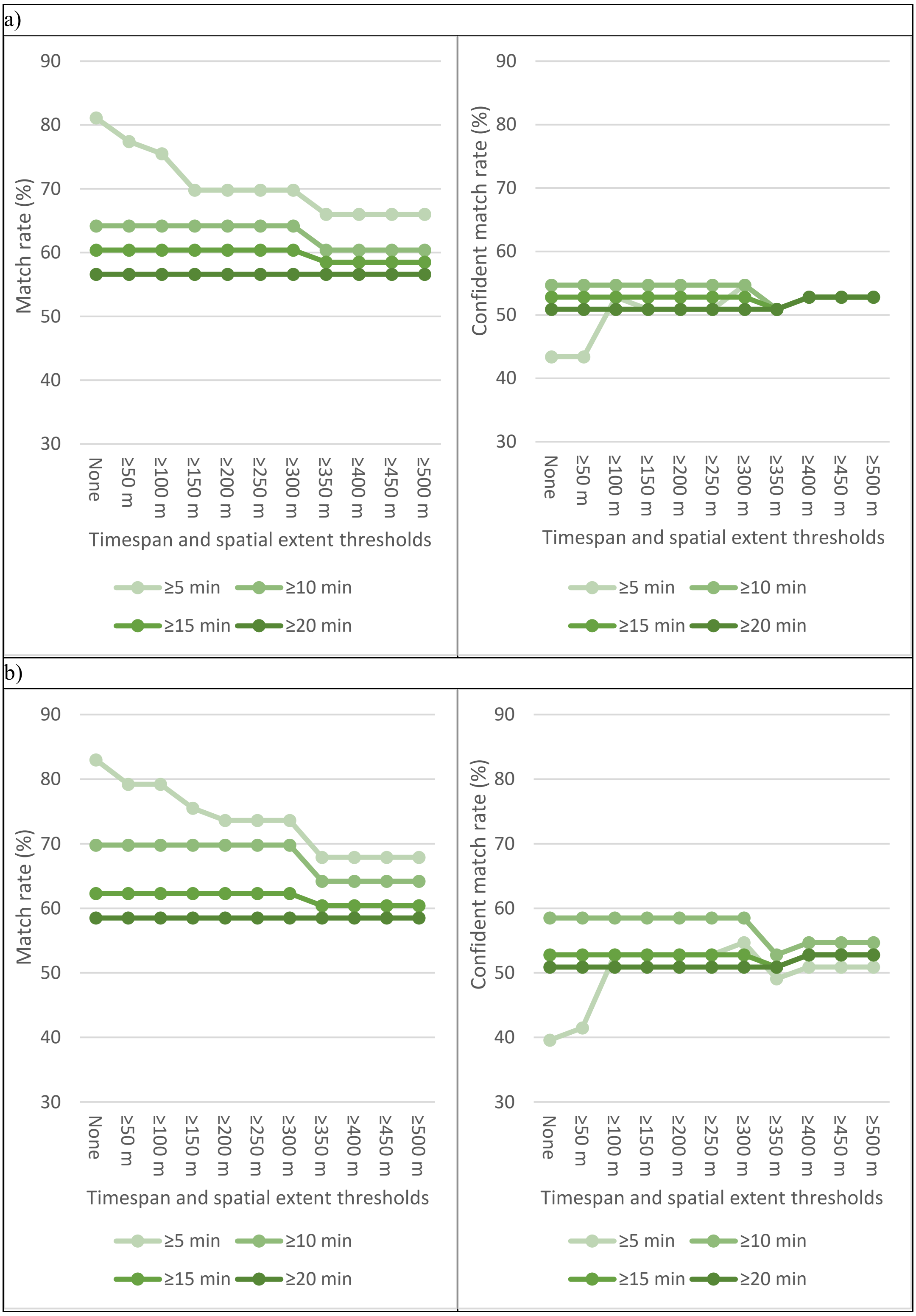


Figure 6.22 Percentage of accident and breakdown posts matched by an NRC using various spatial extends and timespan thresholds using a) 750-meter by 30-minute buffer b) 1000-meter by 45-minute buffer

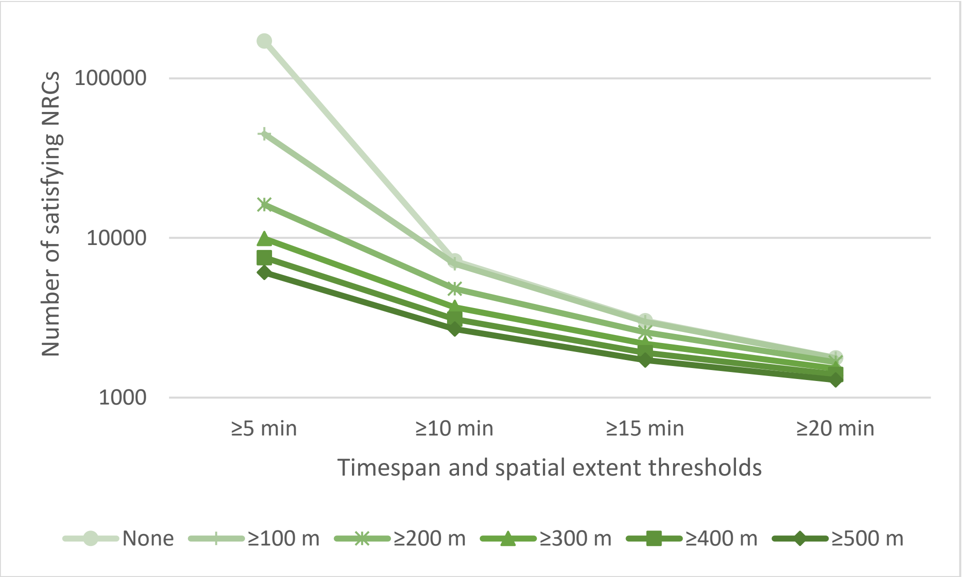


Figure 6.23 Number of NRCs by various spatial extent and timespan thresholds.

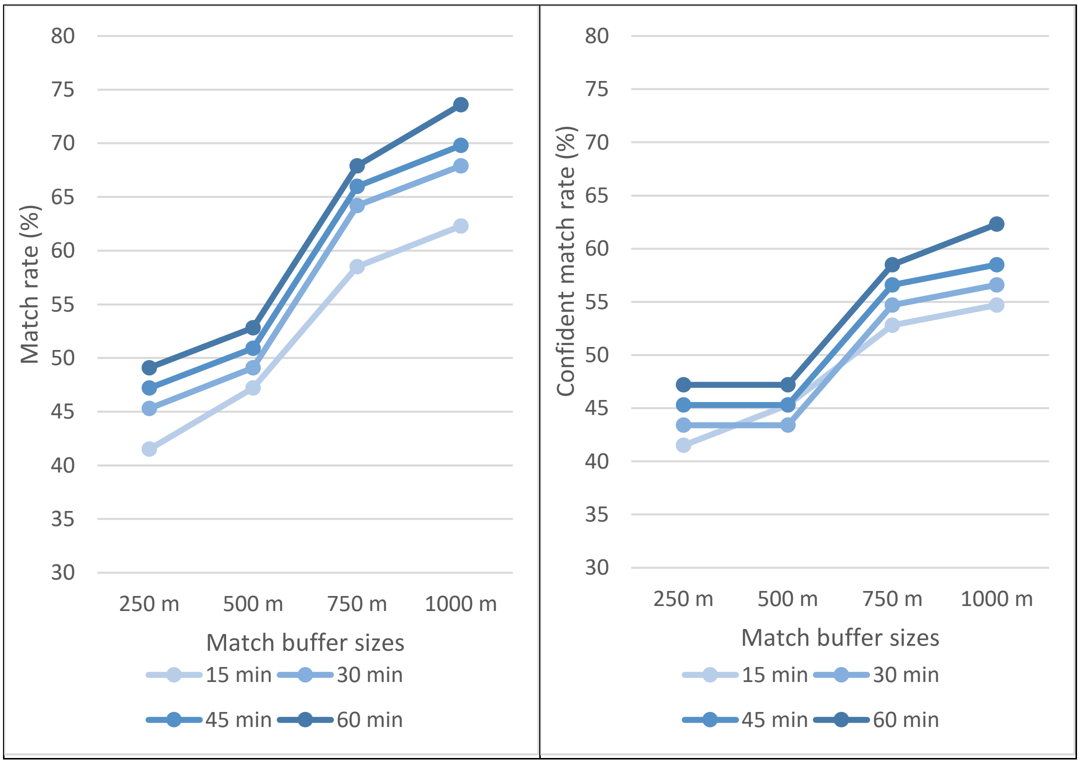


Figure 6.24 Percentage of accident and breakdown posts matched by an NRC by various temporal and spatial buffer sizes, using NRCs satisfying 10-minute timespan and 300-meter spatial extent thresholds

Spatiotemporal information matching method is applied to all types of traffic-event related posts, using a matching buffer of 750-meter by 30-minutes. A 10-minute timespan and 300-meter spatial extent thresholds are used to eliminate noise from detected NRCs to minimize random-matches. Posts reporting accidents and breakdowns has a higher matching rate with NRCs than that of the posts reporting traffic state or other information (63.8%, 66.7% vs 37.6%, 33.3% respectively).

Table 6.36 Match statistics by traffic related post type

Post Type	Number of posts	Matches	Match Rate (%)	HCM
Accident	47	30	63.8	25
Breakdown	6	4	66,7	4
State	93	35	37.6	23
Information	12	4	33.3	4
Unclassified	1	0	0	0

Matching rate of detected NRCs with the posts varied with the size of NRCs. When no timespan or spatial extent thresholds are applied, only 121 of 130,592 NRCs (0.09%) are matched with a post. Verification of NRCs with posts increased with the increase of their impact area. 2.2% of the NRCs are matched with at least one post when a 10-minute timespan and 300-meter spatial extent thresholds are used to filter NRCs. Percentage of matching with at least one post exceeded 10% for NRCs with an impact area above 45-minute timespan and 1000-meter spatial extent. 33% of NRCs with an above 45-minute timespan and an above 5000-meter spatial extent, are verified by at least one tweet post (Figure 6.25).

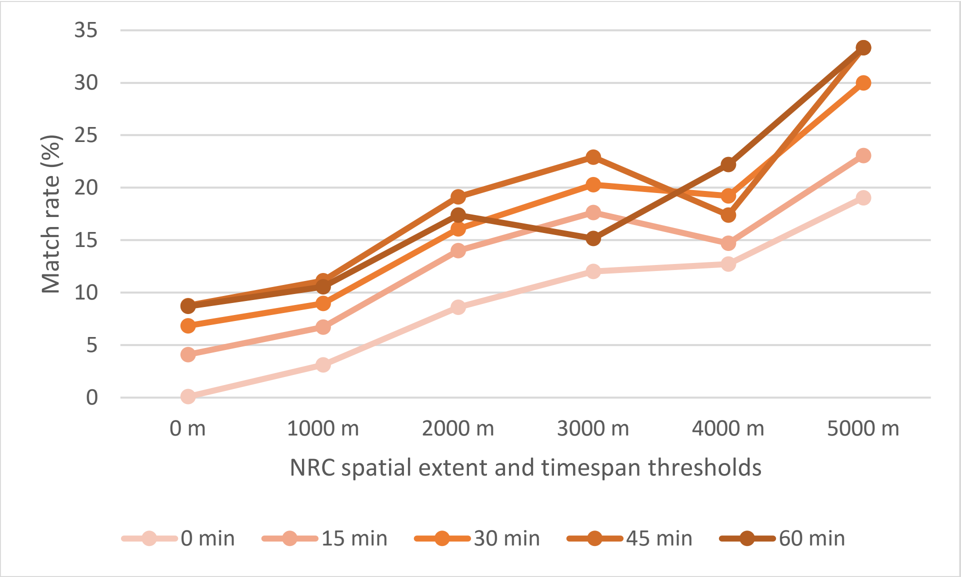


Figure 6.25 Percentage of NRCs matched with a tweet post, by their impact area

6.4.1 Verification using Accident Log

Accident log data released by General Directorate of Security of Ankara consist of events which are registered by police officers. Accidents which are self-reported are not included in the data set. Accident log data has 11,382 records for the year 2017, 197 of which took place on Eskişehir Road. Data includes fields for address definition, geographic coordinates, road features details, injury, and casualty information. In manual inspection of the data, it was observed that there are inconsistencies between the address definition and geographic coordinates of the accidents. Similarly registered times presented an uncertain offset for some accidents when compared with FCD. An accident records set of 59 events with confirmed locations has been created by comparing geographic coordinates with accidents that has a clear definition of address. Various spatial buffer sizes (from 250 to 1,000 meters) and temporal buffer sizes (from 15 to 60 minutes) are evaluated to match accident records with detected NRCs. Percentage of accident records match with an NRC increased with the buffer sizes, reaching 100% with a 1000-meter spatial and 45-minute temporal buffer, whereas the percentage of confident matches achieved highest scores with a 30-minute temporal buffer, reaching 57.3% using a 30-minute by 500-meter and 1000-meter matching buffers (Figure 6.26). Low percentage of confident matches can be an indicator of random matches. Effect of an NRC timespan threshold in decreasing random matches is observed in the matching evaluations performed with various buffer sizes. Along with the timespan thresholds, effect of NRC spatial extent thresholds on the match ratios are explored, using 750-meter by 30-minute and 1000-meter 45-minute matching buffers.

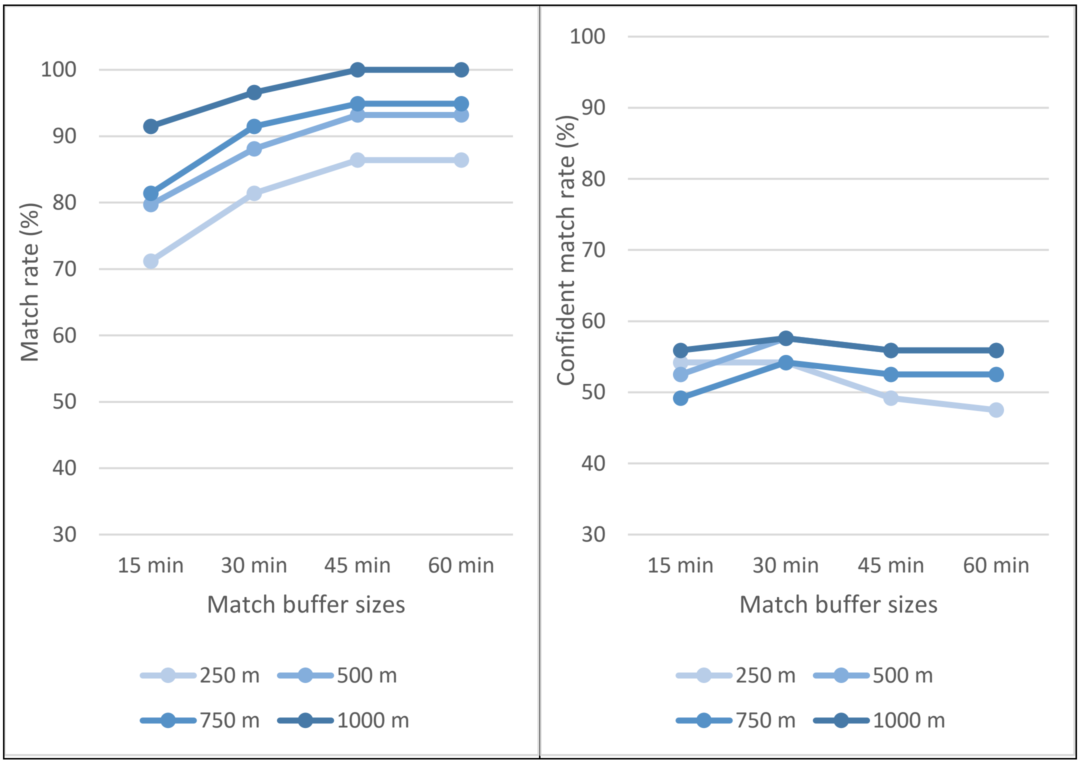


Figure 6.26 Percentage of accident records matched by an NRC by various temporal and spatial buffer sizes

Percentage of matched accident records decreased when any level of timespan or spatial extent are applied to NRCs, match rate was indirectly proportional with the threshold levels. Whereas percentage of accident records matching an NRC with high confidence reached peak level of 66.1% when matched with NRCs with spatial-extent thresholds 0, 50, 100, 150 and 200 meters with a 15-minute timespan threshold

(Figure 6.27). Accident record matching experiments indicate that using a 200-meter spatial-extent along with a 15-minute timespan threshold, would minimize the number of misdetected NRC cases originating from urban operational or other constrains. When NRCs are filtered by a 15-minute timespan and 200-meter spatial extent threshold, percentage of posts matched with an NRC dropped from 91.5% to 72.9% using a 750-meter by 30-minute matching buffer, and from 100% to 79.7% with a 1000-meter by 45-minute matching buffer. Whereas percentage of confident matches increased from 54.2% to 66.1% using a 750-meter by 30-minute buffer, and from 55.9% to 67.8%, using a 1000-meter by 45-minute buffer (Figure 6.27a and Figure 6.27b). A 15-minute interval was also used in Luan et al. (2021) to eliminate data noise in urban road networks.

Proposed incident matching method detected 72.9% of the events in accident log, using a buffer of 750-meter by 30 minutes, using a spatial-extend threshold of 200 meters and timespan threshold of 15 minutes for detected congestions, which are applied to minimize noise and random matches. With the thresholds, number of incidents detected on Eskisehir Road around the year 2017 was 2181. Accident log does not include the self-reported events; therefore, a false-alarm rate could not be calculated.

6.5 Discussion of Results

In this chapter methods proposed to detect and match traffic events in floating car data (FCD) and social media data (SMD) are evaluated on a case study in Ankara. The traffic related event detection method in Twitter stream given in Chapter 3 is evaluated on a collection of tweets collected using a keyword-based search in Ankara. Detection is performed in a morphologically complex language. Hence, proposed method employing morphological analysis improved the performance of the information retrieval tasks in Turkish language.

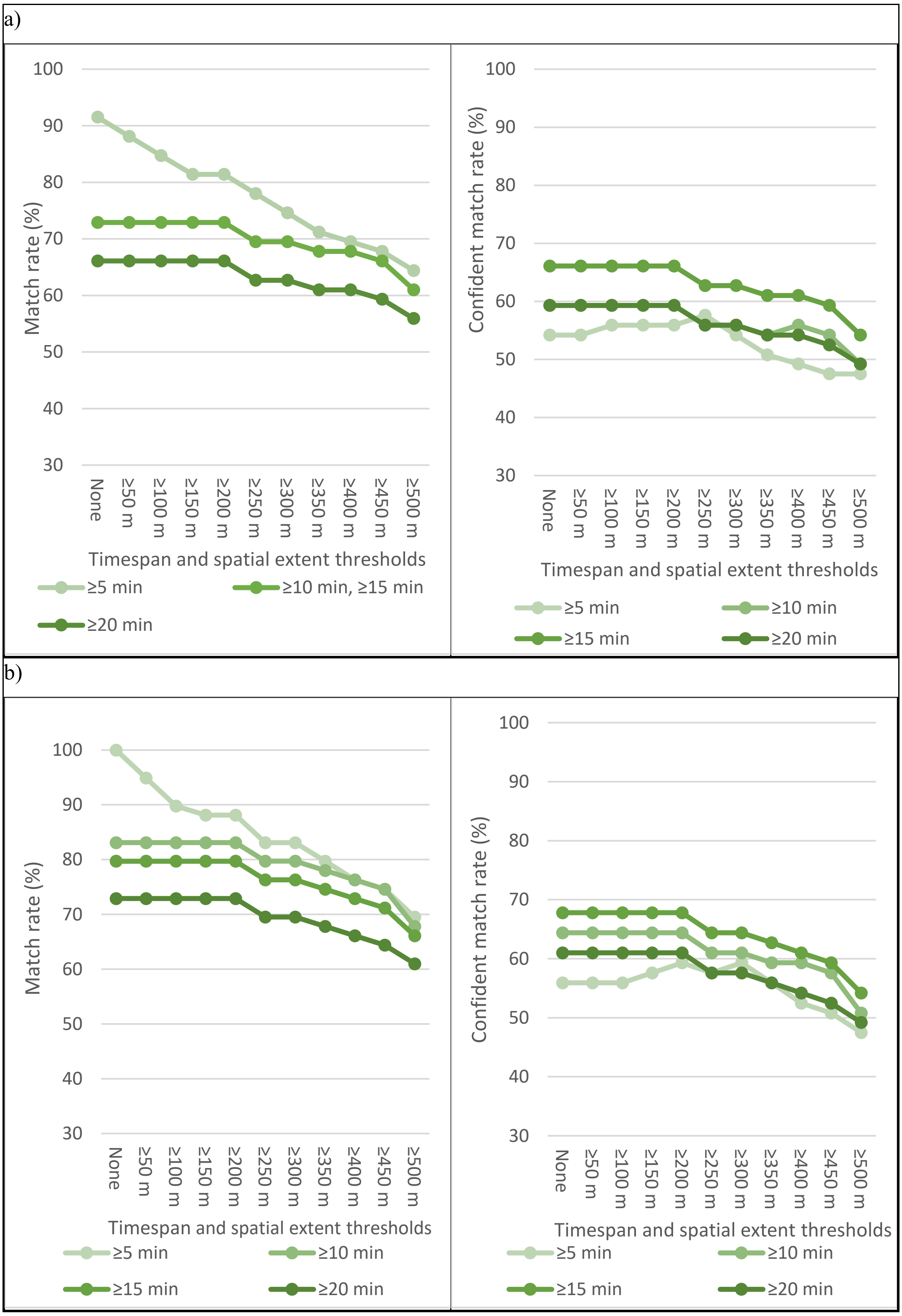


Figure 6.27 Percentage of accident records matched by an NRC using various spatial extends and timespan thresholds using a) 750-meter by 30-minute buffer b) 1000-meter by 45-minute buffer

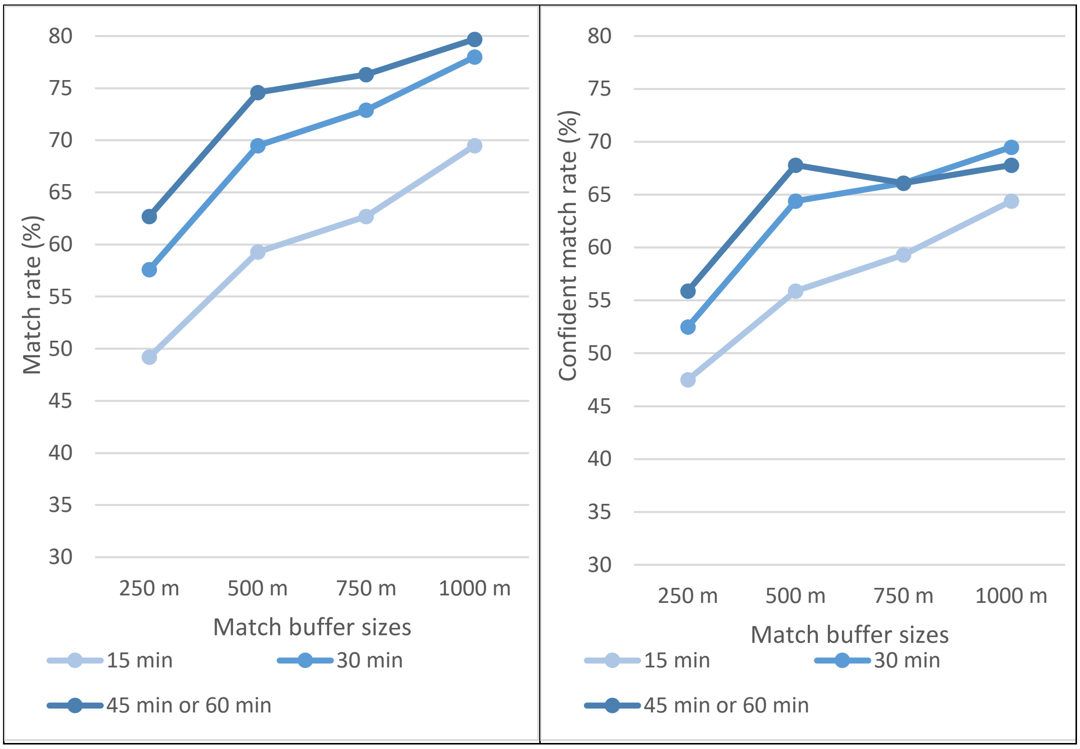


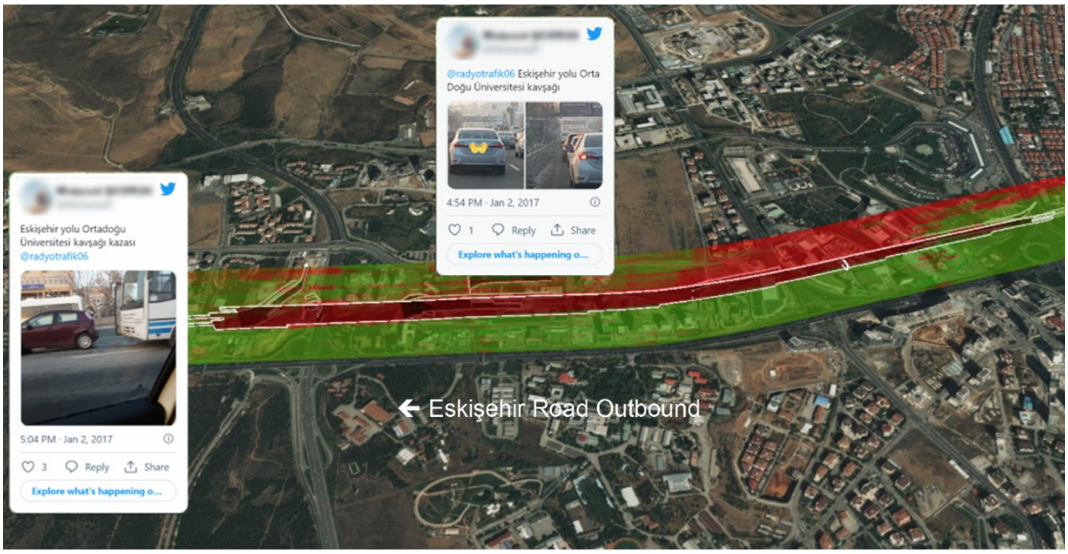
Figure 6.28 Percentage of accident records matched by an NRC by various temporal and spatial buffer sizes, using NRCs satisfying 15-minute timespan and 200-meter spatial extent thresholds

The method has been tested on a set of tweets that is collected within this study. Only the tweets that are direct and immediate reports of incidents are considered as traffic event related within the ground truth. Even though the strict inclusion criteria for labeling exposed classification challenges, the performance of overall system is promising as a cost-effective solution to retrieve traffic-related incidents. Experiments reveal that extracting traffic related entities contributed to the classification performance resulting with higher F1-Scores.

For localization of detected events, Traffic Event Geocoder (TEG), is evaluated on a manually located traffic event data set along with the commonly used off-the-shelf geocoders. TEG employs a named entity recognition model (TEER) customized for detecting terms related to traffic incidents and a rule-based geocoder that interprets the output of TEER, by mapping location entities along with their detected functions using topological relations on the road network. To the best of our knowledge, this is the first work presenting a road-segment level event localization method using a named entity recognition model and a geocoder integrated in a way to complement one another's capabilities. Results show that TEG consisting of a simple set of rules running on a GIS using OpenStreetMap (OSM) data achieves better results localizing unstructured location definitions than the reference geocoding services, geocoding 65.9% of the events under 750-meter positional error.

For detection of non-recurrent congestions (NRCs) in FCD, a two-phase approach is proposed. In, anomaly detection phase, an anomaly factor comparing the estimated and observed travel speeds on FCD segments is calculated. The proposed anomaly detection metric, Road Network-Based Estimator, which is only valid in the existence of a city-wide travel speed data, such as FCD, performed best among other commonly used methods in the anomaly detection studies. In spatiotemporal congestion identification phase, congestion fronts and spillbacks are detected using travel speeds and anomaly factors calculated in the first phase.

Detected NRCs are confirmed with social media posts using the proposed spatiotemporal information matching method. It is observed that percentage of detected NRC could reach up to 33% percent for big-scale events. Figure 6.29 presents a congestion impacting a 2625-meter spatial extent and a 50-minute timespan, which is reported by two tweets, one showing photos about the extent of the congestion, while another includes a photo of the crash causing the congestion. Similarly, in Figure 6.30, an accident causing a congestion is described in detail with the help of a photo.



Eskişehir yolu Ortadoğu Üniversitesi kavşağı kazası @radyotrafik06

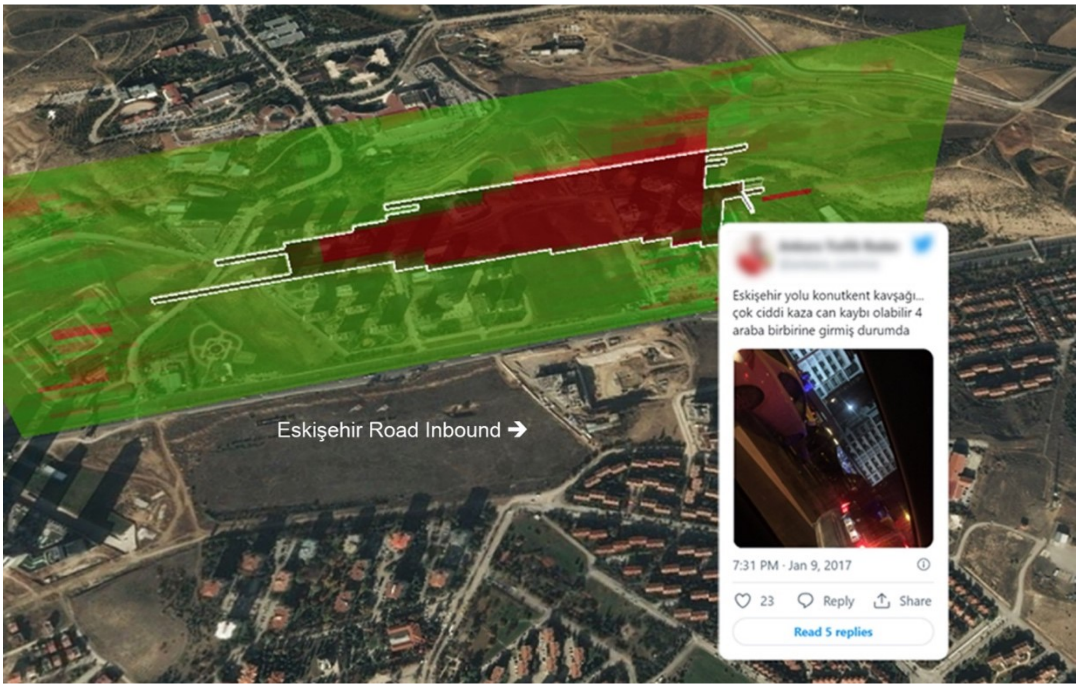
5:04 PM · Jan 2, 2017

@radyotrafik06 Eskişehir yolu Orta Doğu Üniversitesi kavşağı

4:54 PM · Jan 2, 2017

Figure 6.29 A congestion at METU Intersection on Eskisehir Road, with matched traffic-event tweets

In this study 1-minute resolution FCD was aggregated to 5-minute time-windows for computational simplicity. It was possible to detect minor events taking place during off-peak hours using an FCD resampled to 5-minute time resolution. For instance, a minor incident taking place in a 4-lane section of Eskisehir Road during off-peak hours has been detected using 5-minute aggregated FCD (Figure 6.31).

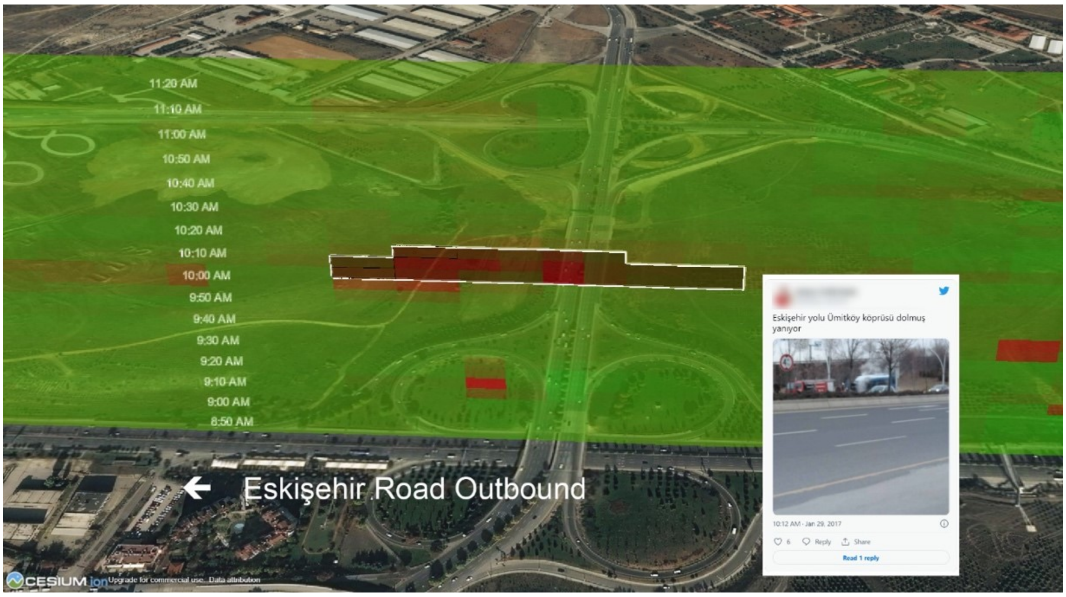


[redacted]

Eskişehir yolu konutkent kavşağı... çok ciddi kaza can kaybı olabilir 4 araba birbirine girmiş durumda

7:31 PM · Jan 9, 2017

Figure 6.30 A congestion at Konutkent Intersection on Eskisehir Road, with matched traffic-event tweet



[redacted]

Eskişehir yolu Ümitköy köprüsü dolmuş yanıyor

10:12 AM · Jan 29, 2017

Figure 6.31 A congestion due to a burning bus on Eskisehir Road outbound, with matched traffic-event tweet

CHAPTER 7

CONCLUSION AND FURTHER RECOMENDATIONS

Traffic management in big cities is a major challenge due to complexity of the road network as well as congestion during peak hours. Traffic incidents further exacerbate the situation as they create congestion and delays in the system, too. Management traffic incidents require real time-detection of them; in intercity road network with limited entry and exit points, this can be done by traditional road sensors which are physically installed at certain locations to detect flow and/or speed data. However, for urban networks, such methods require enormous number of sensors that is not economically not feasible. As an alternative, there is a new source of speed data, called floating car data (FCD), obtained from GPS-equipped vehicles traveling in the network. Especially, the commercial FCD provides a continuous spatiotemporal data which can be further investigated for anomalies (such as queues, congested regimes, etc.) as indicators of traffic incidents. On the other hand, it requires confirmation from other sources as it traces incidents indirectly.

In this study, a framework is proposed to detect and describe non-recurrent congestions using two independent crowdsourced data streams, social media data (SMD) and FCD. Methods are proposed to detect events for each data stream independently and match the results in order to present potential of each data stream to describe and verify non-recurrent congestions.

While SMD is a rich data source including verbal and visual descriptions, it is a challenge to filter out necessary details and detect an "event" of interest, as it requires analysis of the messages using information retrieval techniques. To the best of our knowledge, this is the first study in Turkish language on traffic event detection in social media. Taking agglutinative nature of Turkish language into account, a language model is proposed based on morphological segments of the words. The proposed method employing morphological analysis improved the performance of the information retrieval tasks in Turkish language. A customized named-entity recognition model, Traffic Event Entity Recognition model (TEER), is developed to identify traffic event related terms in Twitter posts. The proposed language model achieved higher accuracy rates in event detection, using commonly used classification methods in event detection studies. Though presented methods are not domain specific and can be used to detect relevant information posted in social media concerning other topics, such as disaster management, health monitoring and various public administration purposes. Proposed language model for Turkish language can also be employed in information retrieval tasks in other domains to achieve higher accuracy.

A knowledge-based approach, Traffic Event Geocoder (TEG), is presented to locate detected traffic event related tweets with road segment level granularity. The method employed the customized named-entity-model, TEER and a rule-based geocoder that is designed to localize traffic event location entities using OpenStreetMap (OSM) data as the knowledgebase. Combining strengths of named entity recognition and GIS, the integrated method provides a better geocoding solution for localizing informal location terms and fills the gap of an end-to-end solution. OSM data is presented as a useful data-source for transportation applications. TEG outperforms reference commercial geocoding services with the existing completeness level of OSM in the study area, and the results are improved further by simply enriching OSM data with a small set of commonly used landmarks.

TEG presented promising results as a specialized geocoder to fulfill the requirements of intelligent transportation applications. The overall geocoding method achieved a median positional error of 379.2 meters in the experiments. Experiments show that geocoding can further be improved by enriching map data. The proposed method geocodes 80.5% of the events under 750-meter positional error with an enriched version of OSM data.

TEG is evaluated on a manually annotated dataset with a limited size. The performance of the proposed geocoding method should further be tested on larger datasets and different regions. In TEER, a generic set of location tags and rules are defined in the method for easy applicability to other urban areas or languages. Relying solely on a freely available dataset with a global coverage, the proposed method can be applied as-is or customized for localizing events in other regions. Retrieving localized information from social content is valuable in domains such as disaster management, local governance, social analytics, and marketing. As a future research direction, the proposed integrated approach can be customized for such applications where localization of unstructured location definitions is needed.

A two-step approach is proposed to detect non-recurrent congestions (NRCs) in floating car data (FCD); anomaly detection and spatiotemporal congestion impact area construction. In anomaly detection step, anomalous travel speeds observed in FCD segments in each epoch are quantified using commonly used statistical metrics, a long short-term memory-based model and a novel metric, Road Network-based Estimator (RNE), which make use of available spatiotemporal travel speed data in FCD. RNE presented better performance than the reference metrics in quantifying anomaly levels on FCD segments. In the case study, time-series data used in statistical methods are not tested for normality, due to limited size of the datasets. Further robust statistical methods can be applied to anomaly detection step to evaluate their performance in anomaly quantification using time-of-day day-of-week based time series data obtained from FCD.

NRCs detected in FCD are matched with the events two event sets, 1) traffic related events detected in social media data and official accident log dataset from General Directorate of Security of Ankara. Matching experiments presented that up to 33% of the NRCs can be confirmed by at least one tweet for large scale traffic events. Congestions detected in FCD provided us with a detailed coverage of impacted street stretches and time intervals, whereas tweets verified and complemented FCD to describe event details causing the congestion.

The matching experiments with the official accident log revealed inconsistencies in the accident location and time fields in the dataset. Incident location inconsistencies are observed between geo-coordinate and address definition fields. Whereas spatiotemporal information matching results indicate possible errors also in registry times of the accidents. Due to quality issues observed in the official accident log, the methods presented lack a validation with a ground-truth data. Validation of the proposed methods with a consistent and complete data set is recommended as a future work.

The methods presented in the study rely on public data created by individuals. The proposed event detection method using social media data is based on the observations of individuals, or *human sensors*. In that sense, *human sensors* contribute to event detection studies as *citizen scientists*. OpenStreetMap (OSM), which is used as the knowledge base of event geolocation, can also be considered as a citizen science project.

The presented method offers a cost-effective solution to traffic event detection, which can be used as a component of an incident management system or other decision support systems concerning traffic events. Relying solely on the data collected from vehicles and social media users, proposed method does not require any instruments to be installed on the road network. Hence, it could be used in any region with any scale for traffic management applications.

REFERENCES

Abdelhaq, H., Sengstock, C., & Gertz, M. (2013). EvenTweet: Online Localized Event Detection from Twitter. *Proc. VLDB Endow.*, *6*(12), 1326–1329. https://doi.org/10.14778/2536274.2536307

Abel, F., Hauff, C., Houben, G.-J., Stronkman, R., & Tao, K. (2012). Twitcident. *Proceedings of the 21st International Conference Companion on World Wide Web - WWW '12 Companion*, 305. https://doi.org/10.1145/2187980.2188035

Agarwal, P., Vaithiyanathan, R., Sharma, S., & Shroff, G. (2012). Catching the long-tail: Extracting local news events from Twitter. *ICWSM 2012 - Proceedings of the 6th International AAAI Conference on Weblogs and Social Media*, 379–382.

Aggarwal, C. C., & Zhai, C. X. (2012). Mining text data. In *Mining Text Data*. Springer Science & Business Media. https://doi.org/10.1007/978-1-4614-3223-4

Altintasi, O., Tuydes-Yaman, H., & Tuncay, K. (2017). Detection of urban traffic patterns from Floating Car Data (FCD). *Transportation Research Procedia*, *22*, 382–391. https://doi.org/10.1016/j.trpro.2017.03.057

Altintasi, O., Tuydes-Yaman, H., & Tuncay, K. (2019). Monitoring urban traffic from floating car data (FCD): Using speed or a los-based state measure. In *Lecture Notes in Networks and Systems* (Vol. 51). Springer International Publishing. https://doi.org/10.1007/978-3-319-98615-9_15

Anbaroglu, B., Heydecker, B., & Cheng, T. (2014). Spatio-temporal clustering for non-recurrent traffic congestion detection on urban road networks. *Transportation Research Part C: Emerging Technologies*, *48*(November), 47–65. https://doi.org/10.1016/j.trc.2014.08.002

Asakura, Y., Kusakabe, T., Long, N. X., & Ushiki, T. (2015). Incident Detection Methods Using Probe Vehicles with On-board GPS Equipment. *Transportation*

Research Procedia, *6*(June 2014), 17–27. https://doi.org/10.1016/j.trpro.2015.03.003

Bakerman, J., Pazdernik, K., Wilson, A., Fairchild, G., & Bahran, R. (2019). *Twitter Geolocation: A Hybrid Approach*. *24700*.

Balke, K., Dudek, C. L., & Mountain, C. E. (1996). Using Probe-Measured Travel Times to Detect Major Freeway Incidents in Houston, Texas. *Transportation Research Record*, *1554*, 213–220. https://doi.org/10.3141/1554-25

Bayraktar, Ö., & Temizel, T. T. (2008). Person name extraction from Turkish financial news text using local grammar-based approach. *2008 23rd International Symposium on Computer and Information Sciences, ISCIS 2008*, 1–4. https://doi.org/10.1109/ISCIS.2008.4717897

Becker, H., & Gravano, L. (2011). Beyond Trending Topics: Real-World Event Identification on Twitter (Tech Report).pdf. *Proceedings of the International AAAI Conference on Web and Social Media*, *5*(1), 438–441. https://pdfs.semanticscholar.org/2573/060fb7b47e1a69933a28118fc9fd60c393ff.pdf

Bikel, D. M., Miller, S., Schwartz, R., & Weischedel, R. (1997). Nymble: A High-performance Learning Name-finder. *Proceedings of the Fifth Conference on Applied Natural Language Processing*, 194–201. https://doi.org/10.3115/974557.974586

Borthwick, A. (1999). *A Maximum Entropy Approach to Named Entity Recognition*.

Boser, B. E., Guyon, I. M., & Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. *Proceedings of the Fifth Annual Workshop on Computational Learning Theory - COLT '92*, 144–152. https://doi.org/10.1145/130385.130401

Boyd, D., & Crawford, K. (2012). Critical Questions for Big Data. *Information, Communication & Society*, *15*(5), 1–5. https://doi.org/10.1126/science.1243089

Can, F., Kocberber, S., Baglioglu, O., Kardas, S., Ocalan, H. C., & Uyar, E. (2010). New Event Detection andTopicTracking inTurkish. *Journal of the American Society for Information Science and Technology*, *61*(4), 802–819.

Chakraborty, P., Hegde, C., & Sharma, A. (2019). Data-driven parallelizable traffic incident detection using spatio-temporally denoised robust thresholds. *Transportation Research Part C: Emerging Technologies*, *105*, 81–99. https://doi.org/10.1016/j.trc.2019.05.034

Chen, P.-T., Feng, C., & Zhen, Q. (2014). Road Traffic Congestion Monitoring in Social Media with Hinge-Loss Markov Random Fields. *2014 IEEE International Conference on Data Mining*, 80–89. https://doi.org/10.1109/ICDM.2014.139

Chen, Y., Lv, Y., Wang, X., Li, L., & Wang, F. Y. (2019). Detecting traffic information from social media texts with deep learning approaches. *IEEE Transactions on Intelligent Transportation Systems*, *20*(8). https://doi.org/10.1109/TITS.2018.2871269

Chen, Z., Liu, X. C., & Zhang, G. (2016). Non-recurrent congestion analysis using data-driven spatiotemporal approach for information construction. *Transportation Research Part C: Emerging Technologies*, *71*, 19–31. https://doi.org/10.1016/j.trc.2016.07.002

Cheng, Z., Caverlee, J., & Lee, K. (2010). You are where you tweet. *Proceedings of the 19th ACM International Conference on Information and Knowledge Management - CIKM '10*, 759. https://doi.org/10.1145/1871437.1871535

Cheu, R. L., Qi, H., & Lee, D. H. (2002). Mobile sensor and sample-based algorithm for freeway incident detection. *Transportation Research Record*, *1811*, 12–20. https://doi.org/10.3141/1811-02

Chieu, H. L., & Ng, H. T. (2002). Named entity recognition. In *Proceedings of the 19th international conference on Computational linguistics* - (Vol. 1, pp. 1–7). https://doi.org/10.3115/1072228.1072253

Chin, E., Franzese, O., Greene, D., Hwang, H., & Gibson, R. (2002). Temporary losses of highway capacity and impacts on performance. In *Applications of Advanced Technologies in Transportation* (Issue May).

Chong, W. H., & Lim, E. P. (2017). Exploiting contextual information for fine-grained tweet geolocation. *Proceedings of the 11th International Conference on Web and Social Media, ICWSM 2017*, *Icwsm*, 488–491.

Colas, F., & Brazdil, P. (2006). On the behavior of SVM and some older algorithms in binary text classification tasks. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *4188 LNCS*, 45–52. https://doi.org/10.1007/11846406_6

Coltekin, C. (2010). A Freely Available Morphological Analyzer for Turkish. In N. C. (Conference Chair), K. Choukri, T. Declerck, H. Loftsson, B. Maegaard, J. Mariani, A. Moreno, J. Odijk, & S. Piperidis (Eds.), *Proceedings of the 7th International Conference on Language Resources and Evaluation* (pp. 820–827).

Coltekin, C. (2014). A Set of Open Source Tools for Turkish Natural Language Processing. In N. C. (Conference Chair), K. Choukri, T. Declerck, H. Loftsson, B. Maegaard, J. Mariani, A. Moreno, J. Odijk, & S. Piperidis (Eds.), *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*. European Language Resources Association (ELRA.

Cucerzan, S., & Yarowsky, D. (1997). Language Independent Named Entity Recognition Combining Morphological and Contextual Evidence. *Emnlp*, 90–99.

Dabiri, S., & Heaslip, K. (2019). Developing a Twitter-based traffic event detection model using deep learning architectures. *Expert Systems with Applications*, *118*, 425–439. https://doi.org/10.1016/j.eswa.2018.10.017

D'Andrea, E., Ducange, P., Lazzerini, B., & Marcelloni, F. (2015). Real-Time Detection of Traffic from Twitter Stream Analysis. *IEEE Transactions on Intelligent Transportation Systems*, *16*(4), 2269–2283. https://doi.org/10.1109/TITS.2015.2404431

Davis, C. A., Pappa, G. L., de Oliveira, D. R. R., & de, F. (2011). Inferring the location of twitter messages based on user relationships. *Transactions in GIS*, *15*(6), 735–751. https://doi.org/10.1111/j.1467-9671.2011.01297.x

Dowling, R., Skabardonis, A., Carroll, M., & Wang, Z. (2004). *Methodology for Measuring Recurrent and*. *1867*, 60–68.

Dudek, C. L., Messer, C. J., & Nuckles, N. B. (1974). Incident detection on urban freeways. *Transportation Research Record*, *495*, 12–24.

Earle, P. S., Bowden, D. C., & Guy, M. (2011). Twitter earthquake detection: Earthquake monitoring in a social world. *Annals of Geophysics*, *54*(6), 708–715. https://doi.org/10.4401/ag-5364

Erdogan, A. E., Ylmaz, T., Sert, O. C., Akyüz, M., Ozyer, T., & Alhajj, R. (2017). From social media analysis to ubiquitous event monitoring: The case of Turkish tweets. *Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, ASONAM 2017*, 1088–1095. https://doi.org/10.1145/3110025.3120986

Ertugrul, A. M., Velioglu, B., & Karagoz, P. (2017). Word embedding based event detection on social media. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *10334 LNCS*, 3–14. https://doi.org/10.1007/978-3-319-59650-1_1

Fabritiis, C. De, Ragona, R., & Valenti, G. (2008). Traffic Estimation And Prediction Based On Real Time Floating Car Data. *11th International IEEE Conference on Intelligent Transportation Systems*, 197–203. https://doi.org/10.1109/ITSC.2008.4732534

Flatow, D., Naaman, M., Xie, K. E., Volkovich, Y., & Kanza, Y. (2015). On the accuracy of hyper-local geotagging of social media content. *WSDM 2015 - Proceedings of the 8th ACM International Conference on Web Search and Data Mining*, 127–136. https://doi.org/10.1145/2684822.2685296

Fung, G. P. C., Yu, J. X., Yu, P. S., & Lu, H. (2005). Parameter free bursty events detection in text streams. *VLDB '05 Proceedings of the 31st International Conference on Very Large Data Bases*, *1*, 181–192. https://doi.org/10.1.1.60.2671

Gelernter, J., & Balaji, S. (2013). An algorithm for local geoparsing of microtext. *GeoInformatica*, *17*(4), 635–667. https://doi.org/10.1007/s10707-012-0173-8

Gelernter, J., & Mushegian, N. (2011). Geo-parsing messages from microtext. *Transactions in GIS*, *15*(6), 753–773. https://doi.org/10.1111/j.1467-9671.2011.01294.x

Genc, H., & Yilmaz, B. (2019). Text-Based Event Detection: Deciphering Date Information Using Graph Embeddings. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *11708 LNCS*, 266–278. https://doi.org/10.1007/978-3-030-27520-4_19

Gu, Y., Qian, Z., & Chen, F. (2016). From Twitter to detector: Real-time traffic incident detection using social media data. *Transportation Research Part C: Emerging Technologies*, *67*, 321–342. https://doi.org/10.1016/j.trc.2016.02.011

Gutierrez, C., Figuerias, P., Oliveira, P., Costa, R., & Jardim-Goncalves, R. (2015). Twitter mining for traffic events detection. *Proceedings of the 2015 Science* *and Information Conference, SAI 2015*, 371–378. https://doi.org/10.1109/SAI.2015.7237170

Hall, R. W. (1993). Non-recurrent congestion: How big is the problem? Are traveler information systems the solution? *Transportation Research Part C*, *1*(1), 89–103. https://doi.org/10.1016/0968-090X(93)90022-8

Han, B., Cook, P., & Baldwin, T. (2014). Text-based twitter user geolocation prediction. *Journal of Artificial Intelligence Research*, *49*, 451–500. https://doi.org/10.1613/jair.4200

Hochenbaum, J., Vallis, O. S., & Kejariwal, A. (2017). *Automatic Anomaly Detection in the Cloud Via Statistical Learning*. http://arxiv.org/abs/1704.07706

Hoose, N., Vicencio, M.A., Zhang, X. (1992). Incident detection in urban roads using computer image processing. *Traffic Engineering and Control*, *33*, 236–244.

Ishak, S., & Al-Deek, H. (1999). Performance of automatic ANN-based incident detection on freeways. *Journal of Transportation Engineering*, *125*(4), 281–290. https://doi.org/10.1061/(ASCE)0733-947X(1999)125:4(281)

Isozaki, H., & Kazawa, H. (2002). Efficient Support Vector Classifiers for Named Entity Recognition. *Coling '02*, *1*, 1–7. https://doi.org/10.3115/1072228.1072282

Ivan, J. N. (1997). Neural network representations for arterial street incident detection data fusion. *Transportation Research Part C: Emerging Technologies*, *5*(3–4), 245–254. https://doi.org/10.1016/S0968-090X(97)00018-1

Ivan, J. N., Schofer, J. L., Koppelman, F. S. and Massone, L. L. E. (1995). Real-time data fusion for arterial street incident detection using neural networks. *Transportation Research Record*, *1497*, 27–35.

Jalaparthi, A., & Kumar, A. S. (2016). Monitoring and analysis of real time detection of traffic from Twitter stream analysis. *International Journal of Scientific Research in Computer Science and Engineering*, *4*(3), 33–36.

Jiang, Z., Zhang, S., & Zeng, J. (2013). A hybrid generative/discriminative method for semi-supervised classification. *Knowledge-Based Systems*, *37*, 137–145. https://doi.org/10.1016/j.knosys.2012.07.020

Karagoz, P., Oguztuzun, H., Cakici, R., Ozdikis, O., Onal, K. D., & Sagcan, M. (2016). Extracting Location Information from Crowd-sourced Social Network Data. *European Handbook of Crowdsourced Geographic Information*, 195–204.

Karim, A., & Adeli, H. (2002). Incident detection algorithm using wavelet energy representation of traffic patterns. *Journal of Transportation Engineering*, *128*(3), 232–242. https://doi.org/10.1061/(ASCE)0733-947X(2002)128:3(232)

Khan, S. M., Chowdhury, M., Ngo, L. B., & Apon, A. (2020). Multi-class twitter data categorization and geocoding with a novel computing framework. *Cities*, *96*. https://doi.org/10.1016/j.cities.2019.102410

Kinsella, S., Murdock, V., & O'Hare, N. (2011). "I'm Eating a Sandwich in Glasgow": Modeling Locations with Tweets. *Proceedings of the 3rd International Workshop on Search and Mining User-Generated Contents*, 61–68. https://doi.org/10.1145/2065023.2065039

Kleinberg, J. (2003). Bursty and Hierarchichal structure in streams. *Data Mining and Knowledge Discovery*, *7(4)*, 373–397. https://doi.org/10.1023/A:1024940629314

Krikorian, R. (2013). *New Tweets per second record, and how!* Twitter Blog. https://blog.twitter.com/2013/new-tweets-per-second-record-and-how

Küçük, D., Jacquet, G., & Steinberger, R. (2014). Named Entity Recognition on Turkish Tweets. *Language Resources and Evaluation (LREC)*, 450–454.

Küçük, D., & Yazıcı, A. (2009). *Named Entity Recognition Experiments on*. 524–535.

Kurkcu, A., Morgul, E. F., & Ozbay, K. (2015). Extended Implementation Methodology for Virtual Sensors: Web-based Real Time Transportation Data Collection and Analysis for Incident Management. *Transportation Research Record: Journal of the Transportation Research Board 2528*, *2528*, 27–37.

Lafferty, J., McCallum, A., & Pereira, F. C. N. (2001). *Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data*. *2001*. https://doi.org/10.1038/nprot.2006.61

Laylavi, F., Rajabifard, A., & Kalantari, M. (2016). A multi-element approach to location inference of Twitter: A case for emergency response. *ISPRS International Journal of Geo-Information*, *5*(5), 1–16. https://doi.org/10.3390/ijgi5050056

Lewis, D. D. (1998). Naive (Bayes) at Forty : The Independence Assumption in Information Retrieval. *European Conference on Machine Learning*, 4–15.

Li, C., & Sun, A. (2017). Extracting fine-grained location with temporal awareness in tweets: A two-stage approach. *Journal of the Association for Information Science and Technology*, *68*(7), 1652–1670. https://doi.org/10.1002/asi.23816

Li, C., Sun, A., & Datta, A. (2012a). Twevent: Segment-based Event Detection from Tweets. *Cikm*, 155–164. https://doi.org/10.1145/2396761.2396785

Li, R., Lei, K. H., Khadiwala, R., & Chang, K. C. C. (2012b). TEDAS: A twitter-based event detection and analysis system. *Proceedings - International Conference on Data Engineering*, 1273–1276. https://doi.org/10.1109/ICDE.2012.125

Li, R., Wang, S., & Chang, K. C. C. (2012c). Multiple location profiling for users and relationships from social network and content. *Proceedings of the VLDB Endowment*, *5*(11), 1603–1614. https://doi.org/10.14778/2350229.2350273

Li, C. sen, & Chen, M. C. (2013). Identifying important variables for predicting travel time of freeway with non-recurrent congestion with neural networks. *Neural Computing and Applications*, *23*(6), 1611–1629. https://doi.org/10.1007/s00521-012-1114-z

Li, X., Lam, W. H. K., & Tam, M. L. (2013). New automatic incident detection algorithm based on traffic data collected for journey time estimation. *Journal of Transportation Engineering*, *139*(8), 840–847. https://doi.org/10.1061/(ASCE)TE.1943-5436.0000566

Li, Y., & Jain, A. K. (1998). Classification of Text Documents c j c j. *Machine Learning*, *41*(8), 1–3.

Li, Y., & McDonald, M. (2005). Motorway incident detection using probe vehicles. *Proceedings of the Institution of Civil Engineers: Transport*, *158*(1), 11–15. https://doi.org/10.1680/tran.2005.158.1.11

Liu, Z., Lv, X., Liu, K., & Shi, S. (2010). Study on SVM compared with the other text classification methods. *2nd International Workshop on Education Technology and Computer Science, ETCS 2010*, *1*, 219–222. https://doi.org/10.1109/ETCS.2010.248

Long, J., Gao, Z., Ren, H., & Lian, A. (2008). Urban traffic congestion propagation and bottleneck identification. *Science in China, Series F: Information Sciences*, *51*(7), 948–964. https://doi.org/10.1007/s11432-008-0038-9

Longueville, B. de, Smith, R. S., & Luraschi, G. (2009). " OMG , from here , I can see the flames !": a use case of mining Location Based Social Networks to acquire spatio- temporal data on forest fires. *Proceedings of the 2009 International Workshop on Location Based Social Networks.*, *c*, 73–80. https://doi.org/10.1145/1629890.1629907

Luan, S., Ma, X., Li, M., Su, Y., & Dong, Z. (2021). Detecting and interpreting non-recurrent congestion from traffic and social media data. *IET Intelligent Transport Systems*. https://doi.org/10.1049/itr2.12104

Mahmud, J., Nichols, J., & Drews, C. (2014). Home location identification of twitter users. *ACM Transactions on Intelligent Systems and Technology*, *5*(3). https://doi.org/10.1145/2528548

Manning, C. D. (2008). *Introduction to information retrieval*. Syngress Publishing,.

Masters, P. H., Lam, J. K., & Wong, K. (1991). Incident detection algorithms for COMPASS. An advanced Traffic Management System. *Proceedings - Society of Automotive Engineers*, *P-253 pt 1*, 295–310. https://doi.org/10.1109/vnis.1991.205776

McCallum, A., & Li, W. (2003). Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003* -, *4*, 188–191. https://doi.org/10.3115/1119176.1119206

McCallum, A., & Nigam, K. (1998). A Comparison of Event Models for Naive Bayes Text Classification. *AAAI-98 Workshop on Learning for Text Categorization*, *752*(1), 41–48. https://doi.org/10.1002/em.22125

Middleton, S. E., Middleton, L., & Modafferi, S. (2014). Real-time crisis mapping of natural disasters using social media. *IEEE Intelligent Systems*, *29*(2), 9–17. https://doi.org/10.1109/MIS.2013.126

Mitchell, T. M. (1997). *Machine Learning*. McGraw-hill. https://doi.org/10.1016/B978-0-32-385787-1.00011-7

Nadeau, D. (2007). A survey of named entity recognition and classification. *Linguisticae Investigationes*, *30*, 3–26. https://doi.org/10.1075/li.30.1.03nad

Neudorff, L. G., Randall, J., Reiss, R. A., & Gordon, R. L. (2003). *Freeway management and operations handbook* (Issue September).

Nguyen, H., Liu, W., Rivera, P., & Chen, F. (2016). TrafficWatch: Real-time traffic incident detection and monitoring using social media. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence* *and Lecture Notes in Bioinformatics)*, *9651*(Ml), 540–551. https://doi.org/10.1007/978-3-319-31753-3_43

Oflazer, K. (1990). *Two-level Description of Turkish Morphology Kemal Oflazer Two-level description of Turkish morphology 3 Example Output*. 472.

Ozdikis, O., Oguztuzun, H., & Karagoz, P. (2016). Evidential estimation of event locations in microblogs using the Dempster-Shafer theory. *Information Processing and Management*, *52*(6), 1227–1246. https://doi.org/10.1016/j.ipm.2016.06.001

Ozdikis, O., Oğuztüzün, H., & Karagoz, P. (2017). A survey on location estimation techniques for events detected in Twitter. *Knowledge and Information Systems*, *52*(2), 291–339. https://doi.org/10.1007/s10115-016-1007-z

Özkaya, S., & Diri, B. (2011). Named Entity Recognition by Conditional Random Fields from Turkish informal texts. *2011 IEEE 19th Conference on Signal Processing and Communications Applications (SIU)*.

Paraskevopoulos, P., & Palpanas, T. (2015). Fine-grained geolocalisation of non-geotagged tweets. *Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, ASONAM 2015*, *August*, 105–112. https://doi.org/10.1145/2808797.2808869

Paule, J. D. G., Sun, Y., & Moshfeghi, Y. (2019). On fine-grained geolocalisation of tweets and real-time traffic incident detection. *Information Processing and Management*, *56*(3), 1119–1132. https://doi.org/10.1016/j.ipm.2018.03.011

Payne, H. J., & Tignor, S. C. (1978). Freeway incident detection algorithms based on decision tree with states. *Transportation Research Record*, 30–37. http://onlinepubs.trb.org/Onlinepubs/trr/1978/682/682-005.pdf

Petty, K. F., Skabardonis, A., & Varaiya, P. P. (1997). Incident Detection with Probe Vehicles: Performance, Infrastructure Requirements, and Feasibility. *IFAC*

Proceedings Volumes, *30*(8), 125–130. https://doi.org/10.1016/s1474-6670(17)43812-2

Priedhorsky, R., Culotta, A., & del Valle, S. Y. (2014). Inferring the origin locations of tweets with quantitative confidence. *Proceedings of the ACM Conference on Computer Supported Cooperative Work, CSCW*, 1523–1536. https://doi.org/10.1145/2531602.2531607

Quinlan, J. R. (1986). Induction of Decision Trees. *Machine Learning*, *1*(1), 81–106. https://doi.org/10.1023/A:1022643204877

Quinlan, J. R. (2014). *C4.5: Programs for Machine Learning*. Elsevier.

Rau, L. F. (1991). Extracting company names from text. *[1991] Proceedings. The Seventh IEEE Conference on Artificial Intelligence Application*, *i*, 29–32. https://doi.org/10.1109/CAIA.1991.120841

Ribeiro, S. S., Davis, C. A., Oliveira, D. R. R., Meira, W., Gonçalves, T. S., & Pappa, G. L. (2012). Traffic observatory. *Proceedings of the 5th International Workshop on Location-Based Social Networks - LBSN '12*, 5. https://doi.org/10.1145/2442796.2442800

Ritter, A., Clark, S., Mausam, & Etzioni, O. (2011). Named Entity Recognition in Tweets: An Experimental Study. *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, 1524–1534. https://doi.org/10.1075/li.30.1.03nad

Rodrigues, E., Assunção, R., Pappa, G. L., Renno, D., & Meira, W. (2016). Exploring multiple evidence to infer users' location in Twitter. *Neurocomputing*, *171*, 30–38. https://doi.org/10.1016/j.neucom.2015.05.066

Roller, S., Speriosu, M., Rallapalli, S., Wing, B., & Baldridge, J. (2012). Supervised text-based geolocation using language models on an adaptive grid. *EMNLP-CoNLL 2012 - 2012 Joint Conference on Empirical Methods in Natural*

Language Processing and Computational Natural Language Learning, Proceedings of the Conference, *July*, 1500–1510.

Rosner, B. (1983). Percentage points for a generalized ESD many-outlier procedure. *Technometrics*, *25*(2), 165–172.

Ryoo, K. M., & Moon, S. (2014). Inferring twitter user locations with 10km accuracy. *WWW 2014 Companion - Proceedings of the 23rd International Conference on World Wide Web*, 643–648. https://doi.org/10.1145/2567948.2579236

Sakaki, T., Matsuo, Y., Yanagihara, T., Chandrasiri, N. P., & Nawa, K. (2012). Real-time event extraction for driving information from social sensors. *Proceedings - 2012 IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems, CYBER 2012*, 221–226. https://doi.org/10.1109/CYBER.2012.6392557

Sakaki, T., Okazaki, M., & Matsuo, Y. (2010). Earthquake Shakes Twitter Users: Real-time Event Detection by Social Sensors. *Proceedings of the 19th International Conference on World Wide Web*, 851–860. https://doi.org/10.1145/1772690.1772777

Schulz, A., Hadjakos, A., Paulheim, H., Nachtwey, J., & Mühlhäuser, M. (2013a). A multi-indicator approach for geolocalization of tweets. *Proceedings of the 7th International Conference on Weblogs and Social Media, ICWSM 2013*, 573–582.

Schulz, A., Ristoski, P., & Paulheim, H. (2013b). I see a car crash: Real-time detection of small scale incidents in microblogs. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *7955 LNCS*, 22–33. https://doi.org/10.1007/978-3-642-41242-4_3

Schulz, A., Schmidt, B., & Strufe, T. (2015). Small-Scale Incident Detection Based on Microposts. *Proceedings of the 26th ACM Conference on Hypertext &#38; Social Media*, 3–12. https://doi.org/10.1145/2700171.2791038

Şeker, G. A., & Eryiğit, G. (2012). Initial Explorations on using {CRF}s for {T}urkish Named Entity Recognition. *Proceedings of COLING 2012*, 2459–2474.

Sermons, M. W., & Koppelman, F. S. (1996). Use of vehicle positioning data for arterial incident detection. *Transportation Research Part C: Emerging Technologies*, *4*(2), 87–96. https://doi.org/10.1016/0968-090X(96)00003-4

Sethi, V., Bhandari, N., S. Koppelman, F., & L. Schofer, J. (1995). Arterial incident detection using fixed detector and probe vehicle data. *Transportation Research Part C*, *3*(2), 99–112. https://doi.org/10.1016/0968-090X(94)00017-Y

Souza, C., Kirillov, A., Catalano, M. D., & contributors, Accord. N. (2014). *The Accord.NET Framework*. https://doi.org/10.5281/zenodo.1029480

Suat-Rojas, N., Gutierrez-Osorio, C., & Pedraza, C. (2022). Extraction and Analysis of Social Networks Data to Detect Traffic Accidents. *Information (Switzerland)*, *13*(1). https://doi.org/10.3390/info13010026

Sun, H., Wu, J., Ma, D., & Long, J. (2014). Spatial distribution complexities of traffic congestion and bottlenecks in different network topologies. *Applied Mathematical Modelling*, *38*(2), 496–505. https://doi.org/10.1016/j.apm.2013.06.027

Sun, R., Liu, J., Sun, J., & Guan, J. (2010). The Impact Analysis of the Urban Highway Traffic Incidents on FCD. *ICCTP 2010: Integrated Transportation Systems*, 2515–2521.

Takeuchi, K., & Yuji, M. (1995). HMM Parameter Learning for Japanese Morphological Analyzer. In *Proceedings of the 10th Pacific Asia Conference* *on Language, Information and Computation* (pp. 163–172). http://aclweb.org/anthology/Y95-1022

Tatar, S., & Cicekli, I. (2011). Automatic rule learning exploiting morphological features for named entity recognition in Turkish. *Journal of Information Science*, *37*(2), 137–151. https://doi.org/10.1177/0165551511398573

Thomas, N. E. (1998). Multi-state and multi-sensor incident detection systems for arterial streets. *Transportation Research Part C: Emerging Technologies*, *6*(5–6), 337–357. https://doi.org/10.1016/S0968-090X(99)00003-0

Tür, G., Hakkani-Tür, D., & Oflazer, K. (2003). A statistical information extraction system for Turkish. *Natural Language Engineering*, *9*(2), S135132490200284X. https://doi.org/10.1017/S135132490200284X

Twitter. (2016). *About Company*. http://about.twitter.com/company

Twitter - Statistics & Facts. (2022). https://www.statista.com/topics/737/twitter/

Twitter Usage Statistics. (2022). https://www.internetlivestats.com/twitter-statistics

Vallejos, S., Alonso, D. G., Caimmi, B., Berdun, L., Armentano, M. G., & Soria, Á. (2021). Mining Social Networks to Detect Traffic Incidents. *Information Systems Frontiers*, *23*(1), 115–134. https://doi.org/10.1007/s10796-020-09994-3

Vapnik, V. (1999). *The nature of statistical learning theory*. Springer science \& business media.

Wang, S., He, L., Stenneth, L., Yu, P. S., & Li, Z. (2015). Citywide traffic congestion estimation with social media. *GIS: Proceedings of the ACM International Symposium on Advances in Geographic Information Systems*, *03-06-Nove*. https://doi.org/10.1145/2820783.2820829

Wang, S., Zhang, X., Cao, J., He, L., Stenneth, L., Yu, P. S., Li, Z., & Huang, Z. (2017). Computing urban traffic congestions by incorporating sparse GPS probe data and social media data. *ACM Transactions on Information Systems*, *35*(4). https://doi.org/10.1145/3057281

Xing, Y., Ban, X., Liu, X., & Shen, Q. (2019). Large-scale traffic congestion prediction based on the symmetric extreme learning machine cluster fast learning method. *Symmetry*, *11*(6), 1–19. https://doi.org/10.3390/sym11060730

Yang, Y. (1999). An evaluation of statistical approaches to text categorization. *Information Retrieval*, *1*(1–2), 69–90. https://doi.org/10.1023/a:1009982220290

Yeniterzi, R. (2011). Exploiting Morphology in Turkish Named Entity Recognition System. *Proceedings of the ACL 2011 Student Session*, *June*, 105–110.

Yuan, F., & Cheu, R. L. (2003). Incident detection using support vector machines. *Transportation Research Part C: Emerging Technologies*, *11*(3–4), 309–328. https://doi.org/10.1016/S0968-090X(03)00020-2

Zhang, H., & Li, D. (2008). *Naive Bayes Text Classifier*. 708–708. https://doi.org/10.1109/grc.2007.40

Zhang, K., & Taylor, M. A. P. (2006). Effective arterial road incident detection: A Bayesian network based algorithm. *Transportation Research Part C: Emerging Technologies*, *14*(6), 403–417. https://doi.org/10.1016/j.trc.2006.11.001

Zhang, W., & Gelernter, J. (2014). Geocoding location expressions in Twitter messages: A preference learning method. *Journal of Spatial Information Science*, *9*(9), 37–70. https://doi.org/10.5311/josis.v0i9.170

Zhang, Z., & He, Q. (2016). *On-site Traffic Accident Detection with Both Social Media and Traffic Data*. *3*.

Zhang, Z., He, Q., Gao, J., & Ni, M. (2018). A deep learning approach for detecting traffic accidents from social media data. *Transportation Research Part C: Emerging Technologies*, *86*. https://doi.org/10.1016/j.trc.2017.11.027

Zhao, J., Gao, Y., Bai, Z., Wang, H., & Lu, S. (2019). Traffic speed prediction under non-recurrent congestion: based on lstm method and beidou navigation satellite system data. *IEEE Intelligent Transportation Systems Magazine*, *11*(2), 70–81. https://doi.org/10.1109/MITS.2019.2903431

Zhao, X., Weng, J. C., & Rong, J. (2010). Urban expressway incident detection algorithm based on floating car data. *ICCTP 2010: Integrated Transportation Systems: Green, Intelligent, Reliable - Proceedings of the 10th International Conference of Chinese Transportation Professionals*, *382*(December), 2132–2139. https://doi.org/10.1061/41127(382)229

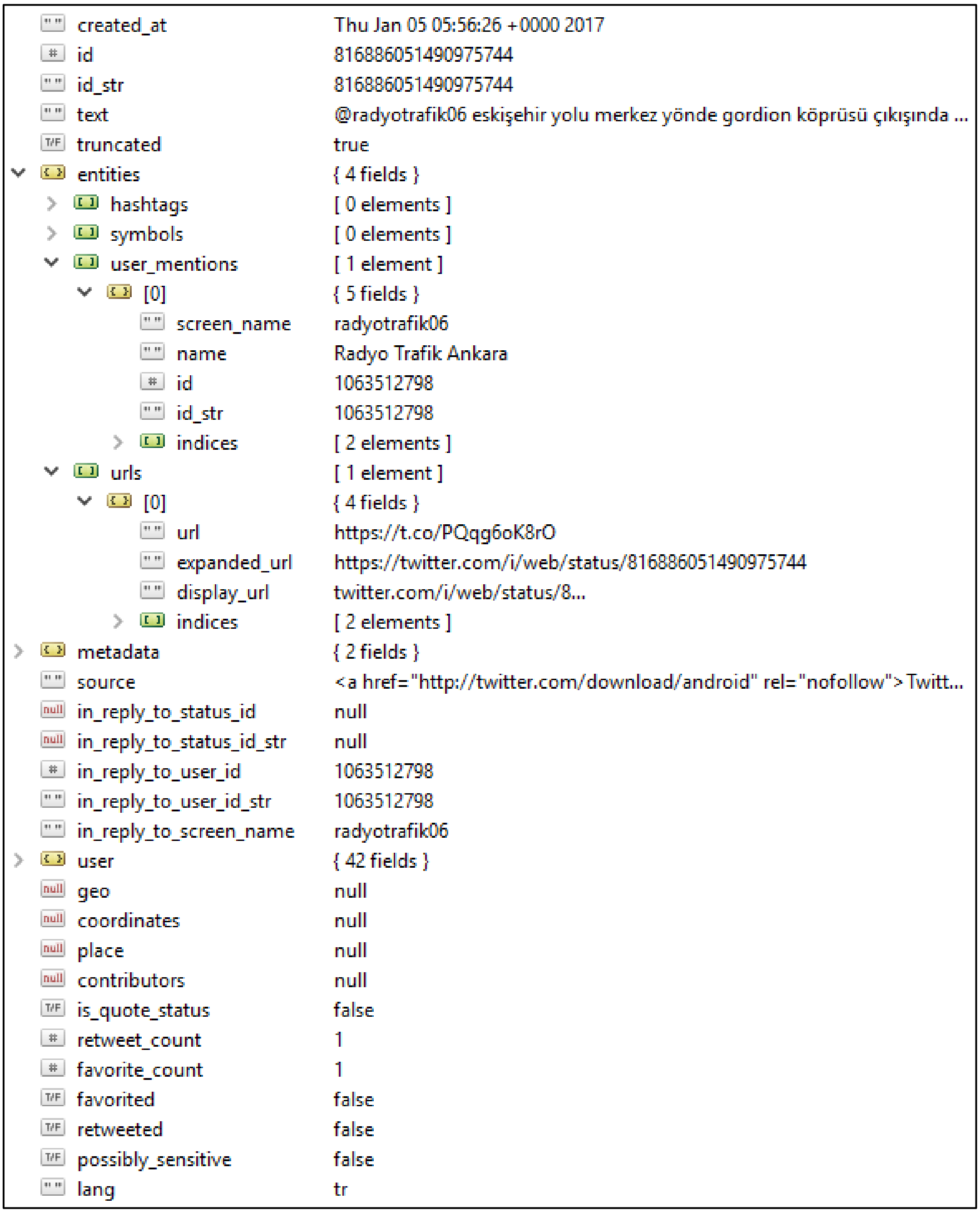
Zheng, X., Han, J., & Sun, A. (2018). A Survey of Location Prediction on Twitter. *IEEE Transactions on Knowledge and Data Engineering*, *30*(9), 1652–1671. https://doi.org/10.1109/TKDE.2018.2807840

Zhou, G., & Su, J. (2002). Named Entity Recognition using an HMM-based Chunk Tagger. *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, *July*, 473–480. https://doi.org/10.3115/1073083.1073163

Zhu, T., Wang, J., & Lv, W. (2009). Outlier mining based automatic incident detection on urban arterial road. *Proceedings of the 6th International Conference on Mobile Technology, Application & Systems (Mobility '09)*, 29:1-29:6. https://doi.org/10.1145/1710035.1710064

APPENDICES

A. Twitter Data Structure



```
created_at                Thu Jan 05 05:56:26 +0000 2017
id                        816886051490975744
id_str                    816886051490975744
text                      @radyotrafik06 eskişehir yolu merkez yönde gordion köprüsü çıkışında ...
truncated                 true
entities                  { 4 fields }
  hashtags                [ 0 elements ]
  symbols                 [ 0 elements ]
  user_mentions           [ 1 element ]
    [0]                   { 5 fields }
      screen_name         radyotrafik06
      name                Radyo Trafik Ankara
      id                  1063512798
      id_str              1063512798
      indices             [ 2 elements ]
  urls                    [ 1 element ]
    [0]                   { 4 fields }
      url                 https://t.co/PQqg6oK8rO
      expanded_url        https://twitter.com/i/web/status/816886051490975744
      display_url         twitter.com/i/web/status/8...
      indices             [ 2 elements ]
metadata                  { 2 fields }
source                    <a href="http://twitter.com/download/android" rel="nofollow">Twitt...
in_reply_to_status_id     null
in_reply_to_status_id_str null
in_reply_to_user_id       1063512798
in_reply_to_user_id_str   1063512798
in_reply_to_screen_name   radyotrafik06
user                      { 42 fields }
geo                       null
coordinates               null
place                     null
contributors              null
is_quote_status           false
retweet_count             1
favorite_count            1
favorited                 false
retweeted                 false
possibly_sensitive        false
lang                      tr
```

Figure A.1 Data structure of a tweet

B. Grid-search Results for Congestion Front Classification Parameters

Table B.1 Top F1-Scores in grid-search for parameters of input values using decision tree-based model

u_{ND}	d_{ND}	u_{SND}	d_{SND}	d_{AF}	TP	FN	TN	FP	Precision	Recall	F1-Score
10	5	1	10	2	135	34	588	26	83.9	79.9	81.9
10	5	1	10	3	134	35	589	25	84.3	79.3	81.7
10	5	1	10	4	135	34	586	28	82.8	79.9	81.3
10	5	1	10	5	135	34	586	28	82.8	79.9	81.3
14	6	1	9	2	126	43	597	17	88.1	74.6	80.8
14	6	1	10	2	126	43	597	17	88.1	74.6	80.8
12	8	1	10	3	130	39	591	23	85	76.9	80.7
10	5	1	14	4	133	36	586	28	82.6	78.7	80.6
10	5	5	4	3	127	42	595	19	87	75.1	80.6
13	4	1	9	3	126	43	596	18	87.5	74.6	80.5
10	5	5	10	3	132	37	587	27	83	78.1	80.5
13	5	1	9	6	127	42	594	20	86.4	75.1	80.4
13	5	1	10	6	127	42	594	20	86.4	75.1	80.4
12	6	1	10	2	127	42	594	20	86.4	75.1	80.4
13	5	1	9	4	126	43	595	19	86.9	74.6	80.3
12	7	1	10	3	131	38	588	26	83.4	77.5	80.3
11	5	1	10	5	134	35	583	31	81.2	79.3	80.2
11	6	1	10	5	134	35	583	31	81.2	79.3	80.2
12	6	1	8	2	136	33	580	34	80	80.5	80.2
12	6	1	10	1	132	37	586	28	82.5	78.1	80.2
12	6	1	11	2	128	41	592	22	85.3	75.7	80.2
11	6	5	10	5	130	39	589	25	83.9	76.9	80.2
12	4	1	8	3	127	42	593	21	85.8	75.1	80.1
13	4	1	9	8	127	42	593	21	85.8	75.1	80.1
10	5	1	13	3	133	36	584	30	81.6	78.7	80.1

Table B.2 Top F1-Scores in grid-search for parameters of input values using SVM model

u_{ND}	d_{ND}	u_{SND}	d_{SND}	d_{AF}	TP	FN	TN	FP	Precision	Recall	F1-Score
9	5	2	9	2	120	49	579	35	77.4	71	74.1
10	5	2	13	4	118	51	579	35	77.1	69.8	73.3
6	3	1	7	3	114	55	582	32	78.1	67.5	72.4
9	5	1	1	1	113	56	583	31	78.5	66.9	72.2
10	6	2	10	3	116	53	578	36	76.3	68.6	72.2
10	5	2	2	2	112	57	584	30	78.9	66.3	72.1
9	5	2	1	3	113	56	582	32	77.9	66.9	72
9	5	2	13	1	113	56	582	32	77.9	66.9	72
12	7	2	6	5	113	56	582	32	77.9	66.9	72
9	5	1	1	10	114	55	580	34	77	67.5	71.9
9	5	1	1	12	114	55	580	34	77	67.5	71.9
10	5	1	1	2	111	58	585	29	79.3	65.7	71.9
7	5	1	3	5	114	55	580	34	77	67.5	71.9
7	5	2	3	2	114	55	580	34	77	67.5	71.9
7	4	5	10	2	115	54	578	36	76.2	68	71.9
12	4	5	13	12	110	59	587	27	80.3	65.1	71.9
8	5	8	14	13	115	54	578	36	76.2	68	71.9
8	4	1	1	1	113	56	581	33	77.4	66.9	71.8
9	4	1	9	1	112	57	583	31	78.3	66.3	71.8
9	5	1	1	2	112	57	583	31	78.3	66.3	71.8
9	5	1	1	4	112	57	583	31	78.3	66.3	71.8
9	5	1	1	6	113	56	581	33	77.4	66.9	71.8
11	5	1	3	1	112	57	583	31	78.3	66.3	71.8
7	6	1	3	1	113	56	581	33	77.4	66.9	71.8
8	6	1	3	1	113	56	581	33	77.4	66.9	71.8

CURRICULUM VITAE

Surname, Name: Ünsal, Ahmet Dündar

EDUCATION

Degree	Institution	Year of Graduation
MS	METU Geodetic & Geographic Information Technologies	2006
BS	METU City and Regional Planning	2000
High School	Antalya Anatolian High School	1995

WORK EXPERIENCE

Year	Place	Enrollment
2019 December	TaleWorlds Entertainment	Senior Software Developer
2018 October	Peak, Istanbul	Backend Engineer
2013 June	TaleWorlds Entertainment	Senior Software Developer
2010 June	Başarsoft	Head of Mobile Applications Department
2008 September	Proje Enerji	GIS Software Developer
2002 August	Inta Spaceturk	GIS Software Developer
2001 September	Başarsoft	GIS Software Developer

FOREIGN LANGUAGES

Advanced English

PUBLICATIONS

1. Unsal, A. D., Usul, N., & Tuydes, H. (2010). Estimation of Time-Dependent Link Costs Using GPS Track Data. Proceedings of IEDC 2010, 1-3 July, 2010, 307-316.

2. Unsal, A. D., Tuydes-Yaman, H., & Karagoz, P. (2019). Traffic Event Related Blog Post Classification by Using Traffic Related Named Entities. 2018 IEEE International Smart Cities Conference, ISC2 2018, 1–8. https://doi.org/10.1109/ISC2.2018.8656940