An Integrated Platform for Mining Crowdsourced Data for Smart Traffic Prediction

Daniele Cenni
University of Florence
Florence, Italy
daniele.cenni@unifi.it

Chenyang Wang, Ahmed Ferdous Antor, Qi Han
Department of Computer Science, Colorado School of Mines
Golden, Colorado, USA
{chenyangwang, antor, qhan}@mines.edu

2022 IEEE International Smart Cities Conference (ISC2) | 978-1-6654-8561-6/22/$31.00 ©2022 IEEE | DOI: 10.1109/ISC255366.2022.9922015

***Abstract*—Traffic prediction can help people make better travel plans by avoiding traffic jams, and also help the city to more proactively deploy emergency response vehicles. The continuous growth of social networks made possible the use of large amounts of data for traffic prediction. One of the biggest challenges in this regard is to acquire and process crowdsourced data to build effective models for traffic prediction. In this paper we propose a novel framework for processing crowdsourced data, with the goal of building effective traffic prediction models. We apply our solution to predict traffic related events in the busiest interstate in Colorado (USA), using Waze crowdsourced data. The events considered in the dataset are moderate jam, heavy jam, and stand still jam. In addition to traffic alerts crowdsourced data via Waze, we also use the traffic speed and weather data. The proposed solution proves to be effective and highly scalable, and the model's best accuracy on the test set is $\sim$76%. This approach can be easily generalized in order to develop models that are able to provide effective traffic related predictions.**

***Index Terms*—crowdsourcing data, traffic prediction, data processing**

I. Introduction

The development of intelligent systems for traffic management has many benefits. In addition to promoting sustainable mobility, it contributes to a significant reduction in vehicle emissions (and therefore air pollution) and traffic accidents. The prediction of road traffic conditions is of particular interest because it has deep implications in different areas (e.g., energy consumption, pollution, management of public events, design of new roads, hourly planning of public transport).

The popularity of mobile devices, the existence of high-speed mobile networks, and the exploitation of real-time data from social networks, have inspired the development of new low-cost techniques for the prediction of traffic flows and conditions. Further, increasingly powerful data mining techniques make it possible to analyze large amounts of data in a reasonably short time, and build effective models that provide accurate predictions.

However, it must be taken into account that the construction of a platform for traffic data analysis, and the consequent training of predictive models, requires different geographically dispersed sources, for the retrieval of real-time events that are sufficiently detailed, with a high sampling rate. There is no doubt that the availability of large amounts of data poses a number of problems that are not easy to solve, as it is necessary to look for data retrieval solutions that are easily scalable, efficient and cost-effective. One of the most suitable approaches in this regard is to use crowdsourced data that may be available across large geographical regions, sufficiently detailed in terms of traffic metrics, and representative of people moving in urban settings. Crowdsourced data are readily available through a number of platforms, for example by exploiting mobile applications, or directly from the Cloud (e.g., Amazon Mechanical Turk).

Crowdsourced data provide a higher number of widespread observations with respect to traditional data collection strategies (e.g. performed with dedicated sensors), but present some challenges as well. Indeed, crowdsourced data collection, being a participatory method for building datasets with the contribution of large groups of people, has some significant disadvantages that need to be resolved in the data preprocessing phases, and which should not be underestimated. First of all, crowdsourced data often introduce relevant sampling issues (e.g., when data come from different and heterogeneous sources). Though crowdsourcing data do not present a rigorous sampling and data structure, it is mandatory to deal with a large network of contributors, in order to have a representative set producing data at sufficiently regular intervals.

Large amount of data from heterogeneous sources seems to be a common problem in many traffic prediction systems, which make use of different data for building smart decision systems. Standard crowdsourced approaches, though, usually make use of single sources, avoiding the reconciliation of different hybrid sources, coming from different platforms. In our context, however, the use of crowdsourced sources does not depend only on considerations related to the information richness of the data produced, but responds to the need to build scalable solutions that can generalize sparse and heterogeneous data, and are applicable to different scenarios with little or no effort. In this regard, the solution described in the present work aims to build an effective traffic analysis and prediction solution, by enriching crowdsourced data with weather data, and at the same time allowing the integration of different kinds of other crowdsourced data.

Recent advances in the field of machine learning and the extensive use of GPU-based computing systems (also instantiated in the Cloud), have made it possible to develop increasingly accurate models that can be effectively applied in different contexts. However, the huge amount of data coming from

heterogeneous sources requires the development of intelligent platforms, which allow the processing, normalization and cleaning of data (e.g., elimination of duplicates, spurious or null data, data balancing, outliers detection), and the construction of optimized datasets for the incremental training of learning models. Therefore, it is essential to build models and tools capable of efficiently processing large amounts of data, in order to provide accurate predictions on traffic trends with different levels of detail (e.g., hourly, daily).

Our proposed solution, constitutes an efficient platform for the analysis of crowdsourced traffic data, which allows to train models for traffic prediction with a high degree of accuracy, and at the same time has a high scalability, thus allowing us to realize real-time predictions. By completely eliminating the problem of geographic data management and storage, our solution is fast and capable of processing large amounts of traffic data in real time. Moreover, our solution is generalizable and extensible to different traffic sources, hence it allows the management of heterogeneous traffic sources and a more effective data reconciliation. For testing the effectiveness of our proposed platform, we applied our solution to predict the traffic related events in the busiest interstate in the state of Colorado (USA).

The rest of the paper is organized as follows. Section II describes the related work in traffic prediction and traffic data analysis. Section III introduces the problem of data retrieving and processing, and describes how traffic related data were collected, processed, cleaned and compressed, for efficient model training. This section also describe how the data fetching and processing techniques were implemented in order to clean and prepare an optimized dataset, consisting of crowdsourced data. Section IV introduces the general architecture of the implemented model, and describes how different learning schemes can be applied for building effective prediction models. Section V discusses the experiments and presents the results with relevant metrics, including the performances of the model (e.g., accuracy) with different inputs. Section VI summarizes the paper.

II. Related Work

This section briefly discusses work in two areas most related to our work: one is using crowdsourced data, and the other is traffic prediction.

Crowdsourcing data from Google Maps that record the activity of the regions of interest has been used to study the correlations between vehicles' emissions, travel times and popular times in [1]; A framework including data collection, data integration and data mining for crowdsourcing based traffic state estimation is developed in [2]. In addition, in [3], historical traffic speed data is used to construct an offline graph model based on Gaussian Markov Random Field to encode the structure of the traffic network and online crowdsourced road selection is used for speed estimation. However, all these works use not only crowdsourced data but also direct measurements from sensors. In contrast, we only consider the crowdsourced data and no sensor reading is involved. In addition, we include richer features from the crowdsourced data compared to the aforementioned works.

And with the increasing power of machine learning, different approaches and techniques have been adopted for traffic prediction problem. For instance, [4] uses Convolutional Neural Networks and [5] uses Recurrent Neural Networks for traffic prediction. Also, traditional machine learning techniques have also been used for traffic prediction, for instance in [6], the authors propose a Hierarchical Fuzzy Rule-Based System to provide traffic jam predictions. Other approaches include Hidden Markov Models [7], Gaussian Distribution [8], Random Forest [9], Support Vector Machines [10] as well.

Regarding to the data analysis part, existing work on traffic prediction make use of historical traffic flow, weather data, and planned public events data. For example, [11] predicts traffic jam in five different levels using estimated time arrival from Google Maps API, holidays, special events, and weather data, while [12] uses rainfall data for traffic flow prediction where the data is collected from National Meteorological Center of the China Meteorological Administration. In [13], two types of weather data in addition to traffic and map data are used for traffic flow prediction: observation data and forecast data (the weather data types are the highest, lowest and average temperature, average humidity and so on).

Weather data provided by National Oceanographic and Atmospheric Administration is used in [14], as an aid in traffic flow prediction in a certain time window. It only considers traffic speed in its calculation because of the difficulty of considering traffic volume. The relevant weather features are selected using Pearson Correlation Coefficient and Principal Component Analysis (PCA). For traffic jam prediction, [15] uses 3-hour, 12-hour and 24-hour weather data provided by the Korean Meteorological Administration. The collected weather data include temperature, humidity, cloud, rainfall, and wind velocity. Traffic jam is determined from the traffic speeds in the road links.

The problem with most of the mentioned approaches is that spatial data analysis is generally a long and computationally complex process, which significantly affects the ability to scale the various solutions under consideration. In particular, geographic data conversion, trajectory clustering, and the development of complex neural networks for geographic information extraction make it necessary to design complex multi-layered architectures. The use of deep neural networks, for example, significantly affects the training time of the models, and makes it difficult to update them in real time to better reflect various changes such as changes in the topology of road networks, the scheduling of public transport services, and big public events (e.g., sport related, concerts).

III. Traffic and Weather Data

For the present study, the crowdsourced traffic data fetched from Waze were provided by the city of Centennial (Colorado, USA). This traffic data include two datasets, alerts and jams, respectively reported in Table I and Table II. We performed experiments with a raw dataset with a size of 74,119 of

rows, spanning from 2018 to 2022 for a total of 26 features. We considered some of the available features as the most representative of the dataset (i.e., year, day, hour, street, reliability, and incident type), and discarded the others, since some other features were redundant or not correlated with traffic incidents (e.g., labels, file ids).

TABLE I
WAZE CROWDSOURCED DATA (ALERTS)

Field	Description	Type
id	Alert id	string
uuid	Alert UUID	string
pub_millis	Timestamp (ms)	integer
pub_utc_date	UTC Date	dateTime
road_type	Road Type	integer
location	Location	string (JSON)
street	Street	string
city	City	string
country	Country ISO 3166-1 Alpha-2 code	string
magvar	Alert direction (0 degrees at North)	integer [0-359]
reliability	Confidence in the alert based on user input	interger
type	Report type (e.g., 'JAM')	string
subtype	Report subtype (e.g., 'JAM MODERATE TRAFFIC')	string
report_by _municipality_user	Whether it is reported by a municipality user	integer
thumbs_up	Number of thumbs up by users	integer
jam_uuid	Jam UUID	string

TABLE II
WAZE CROWDSOURCED DATA (JAMS)

Field	Description	Type
id	Jam id	string
uuid	Jam UUID	string
pub_millis	Timestamp (ms)	integer
pub_utc_date	UTC Date	dateTime
start_node	Nearest Junction, street, city to jam start	string
end_node	Nearest Junction, street, city to jam start	string
road_type	Road Type	integer
street	Street name, no canonical form	string
city	City and state name	string
country	Country ISO 3166-1 Alpha-2 code	string
delay	Jam's delay (s) compared to free flow speed (in case of block, -1)	integer
speed	Current average speed on jammed segments (m/s)	float
speed_kmh	Current average speed on jammed segments in km/h	float
length	Jam length (m)	integer
turn_type	type of turn (i.e., left, right, exit R or L, continue straight or NONE)	string
level	Traffic jam level (0 = free flow 5 = blocked)	integer
line	Alert coordinates	string (JSON)

TABLE III
WAZE EVENTS TYPES

Jam Moderate Traffic	Jam Heavy Traffic
Hazard on Road Ice	Hazard on Shoulder Car Stopped
Hazard on Road Car Stopped	Accident Major
Accident Minor	Hazard on Road Object
Hazard on Road Pot Hole	Hazard on Shoulder Missing Sign
Hazard on Road Traffic Light Fault	Hazard on Road
Hazard Weather Fog	Hazard Weather Flood
Hazard Weather	Hazard Weather Heavy Snow
Jam Stand Still Traffic	Hazard on Shoulder Animals
Hazard on Road Road Kill	Hazard on Road Construction
Hazard on Shoulder	Hazard Weather Hail

The streets that can be found in the datasets are the only ones located in the city of Centennial. However, there are some streets, especially the highways and interstates, that span across cities or even different states. For example, the busiest interstate I-25 goes north and south in both directions across the entire state. In addition, in order to take numerical geological locations presented in the dataset into account as a feature, we chose to use the Geohash system [16] to convert the location information (i.e., the longitude and latitude) into small grids on the map. With a precision of 6, we got 331 grids covering the locations presented in the dataset.

The "incident type" contains string values describing the traffic event. The possible values for event types are reported in Table III. Since we are only concerned about jams prediction, we only consider the jam events in the dataset. Thus, we only use three types of traffic events for our prediction model: moderate jam, heavy jam, and stand still jam.

To consider the truthfulness of the events that have been reported, we use the value of the "reliability" of these multiple instances in the same time period. Since there are three types of events, we have three values for each of them. We then filter out the values that are less then 0.5 to make the prediction label for each event. Thus, we have three new columns in our Waze dataset: moderate jam prediction, heavy jam prediction, and stand still jam prediction, serving as the prediction label for machine learning models training and testing. We then get rid of "reliability" column from the dataset as it has already been taken into account.

To enrich the number of features in our master dataset, we also include weather data. We got these data from *DarkSky.net* by fetching the weather data for a specific location in the dataset, for each instance of the Waze data. The weather data include features such as temperature, pressure, humidity, precipitation, dew point, wind speed, visibility, uvIndex, etc., and they are all numerical values. If some weather feature is not available for some hours, those instances of data are removed

from the master dataset in order to be consistent throughout each data instance. In addition, we removed some weather features that are either not correlated with traffic incidents or having numerical values of 0. The use of meteorological data is particularly important, since, as it is well known, climatic events deeply affect traffic trends, and must be taken into due consideration when building an effective model, for predicting different types of traffic events in real time, based on the analysis of large amounts of crowdsourced data from different sources.

After these intermediate processes are completed, the resulting master dataset has a size of 12,123 and a total of 339 features (i.e., 8 numerical features from the traffic jam and weather plus 331 categorical features encoded from the Geohash system). In addition, all the numerical features have been normalized so that large values of one feature does not overshadow the other ones. The list of the dataset's features is reported in Table IV. These features are the input to the prediction model, which provides a binary classification. The outputs are binary values for each of the three traffic events in the considered area. For example, a value of 1 for heavy jam prediction would mean that the model predicted that there would be a jam occurrence in the next period (couple of hours ahead). On the other hand, a prediction of 0 for moderate traffic would mean there would not be moderate traffic in the next time period. The data is split into 80% training and 20% testing to be used as input into the machine learning model adopted.

TABLE IV
DATASET FEATURES

Name	Type	Unit
delay	float	-
speed	float	-
temperature	float	°C
dew Point	float	°C
wind speed	float	mph
wind gust	float	mph
wind bearing	float	degree
visibility	float	mi
Geohash encodings	string	-

IV. TRAFFIC PREDICTION ARCHITECTURE

As reported in [17], the problem of predicting traffic events include several phases, from data fetching to application of traffic forecasts to different domains.

Data can be spatial-only (e.g., POI information or the road network), temporal-only (e.g., date, timestamp and holiday), spatio-temporal static (e.g., public events), spatial static temporal dynamic (e.g., traffic flows, travel demands, travel time, traffic speed, weather), or spatial dynamic temporal static (e.g., trajectories). Data retrieval is a very complex phase, requiring several steps. For example, it is necessary to take into account aspects related to the confidentiality of the data to be processed, the synchronization of data from different sources, and the imputation of data in order to obtain evenly spaced temporal sequences that can be effectively analyzed.

In the preprocessing phase, different schemes must be applied to data, in order perform the so called map-matching (i.e., match recorded geographic coordinates to a logical model of the real world, using some Geographic Information System). This process also include data cleaning, storage and compression, with the aim of speeding up data utilization (e.g., during the model's training). Next, for the estimation of forecasting, it is mandatory to use the most suitable data mining approach.

The main objective of this work is to build a platform for the generation of datasets containing real-time traffic data from different sources, in order to create effective models for the prediction of traffic events, with a time horizon of both short and medium term. In this regard, the data are processed by a modular platform in order to create a consistent dataset, which allows the construction of a classification model that is able to discriminate between different types of traffic events (e.g., traffic jams).

More specifically, we propose a novel framework that makes use of traffic data provided by Waze, along with publicly available weather data. The developed solution provides traffic related event predictions given the required inputs. The traffic events are categorized into 3 categories: moderate jam, heavy jam, and stand still jam. An effective approach consists in treating geographic data as a categorical feature, in order to eliminate the bottlenecks that inevitably arise during the processing and analysis of the geometries that describe the various geographic elements.

At the same time, the aim is to streamline the process of extracting knowledge from the data, and storing the data, minimizing the impact in terms of complexity of the database, and of the entire platform dedicated to data retrieval and processing. For all intents and purposes, the proposed approach is capable of effectively managing large amounts of data, and is a valid tool for preparing datasets, for building predictive models, that deal with real-time heterogeneous and crowdsourced traffic related events.

A possible approach is to avoid dealing with raw geographical information, and instead focusing on using these data with a different methodology. Specifically, the geometric information of spatial data (e.g., the Euclidean distance), or the topological structure constraints of road networks can be discarded, in order to accelerate the data processing and training phases, and build a more effective prediction model, with respect to the overall accuracy. For this reason, the application of map-matching methods (e.g., point-distance, path-distance, probability-distance, model-based, learning-based) are not necessary, resulting in a benefit in terms of computational time and costs, when analysing and processing traffic related data. The proposed platform, whose general architecture is shown in Fig. 1, consists of the following functional components:

- *Traffic Data fetching*, in this phase the data coming from Waze are fetched from the database (i.e., an AWS cloud endpoint), with a batch loading procedure.

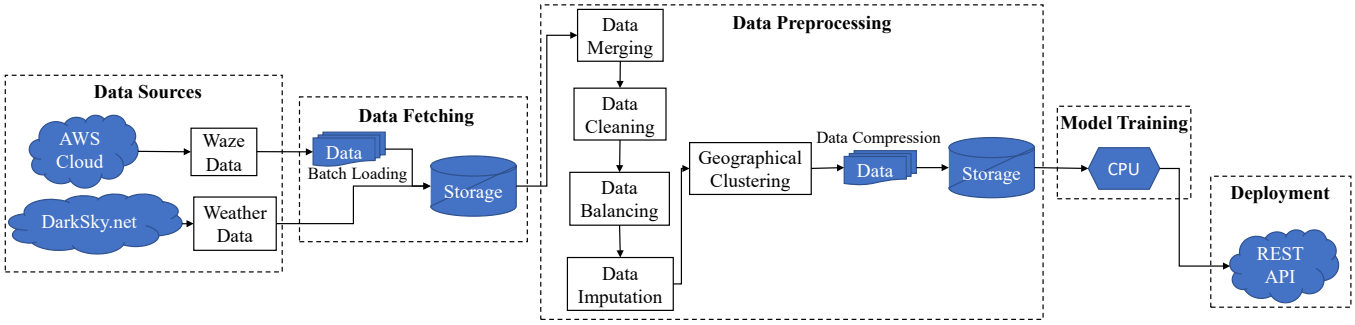

Fig. 1. Traffic Analysis and Prediction Architecture

- *Weather Data fetching*, in this phase the weather data are crawled from *DarkSky.net*.
- *Data Merging*, in this phase Waze data are synchronized with those obtained from Waze. The resulting dataset consists of raw traffic events with relevant weather information.
- *Data Cleaning*, in this phase data are cleaned of any duplicates and null or inconsistent elements. This phase includes removal of data outliers that create noise and could lower the overall performance of the model to be trained. An example of outlier spurious data is a weather metric outside the allowed range, in relation to spatial or temporal neighborhoods. As an example, algorithms such as DBSCAN, could be used, to cluster normal data and then identify potential outliers. Unnecessary columns are removed at this stage of data reconciliation.
- *Data Balancing*, during this phase the data are analyzed to assess whether a balancing scheme is needed. In case of moderate or severe data imbalance, data are balanced using Synthetic Minority Over-Sampling Techniques (SMOTE) [18].
- *Data Imputation*, data are grouped by a given time horizon (e.g., 30 minutes, 1 hour) and the resulting traffic related feature are calculated as the average of all occurrences.
- *Geographical Clustering*, in this phase the geographical information (i.e., latitude and longitude available with WGS84 map projection) are converted into a geographical cluster using various geocoding techniques e.g. the Geohash system aforementioned or the Military Grid Reference System (MGRS). The MGRS is derived from the Universal Transverse Mercator (UTM) grid system and the Universal Polar Stereographic (UPS) grid system, and is used as a geocode system. The local coordinates of a traffic event are transformed into a grid reference (e.g., 4QFJ12345678) consisting of three parts: Grid Zone Designator (GZD), square identifier, and numerical location. Similar to the MGRS, Geohash also converts the coordinates to a grid reference, however the benefit of using Geohash instead of MGRS is that Geohash in general operates faster than the MGRS.
- *Data Compression*, in this phase data are optimized for disk storage, and the dataset is created using the HDF5 format. Data are processed and written in chunks, to effectively deal with huge datasets and memory constraints.
- *Model Training*, in this phase the model is trained on CPU or GPU, using some machine learning schemes.
- *Model Deployment*, the final model can be deployed as a SaaS, and exposed as a REST API service, freely accessible on the Web.

The platform described has the significant advantage of being robust with respect to data errors or missing values, and is capable of handling extremely unbalanced datasets effectively. The ability to discriminate between different or sporadic events is of paramount importance in this regard, because it allows to provide accurate predictions in the case of events attributable to traffic dynamics of rare occurrence, and which in general is difficult to capture with traditional methods used for time series analysis.

The analysis of large amounts of traffic data opens new scenarios, so that the proposed solution can be easily extended to handle different and heterogeneous sources. The described system can be quickly integrated with the most popular Cloud platforms (e.g., Microsoft Azure, Amazon AWS), and allows significant savings in terms of time and costs, being optimized from the point of view of disk space usage, and being easily scalable and suitable for processing big datasets in real time.

The above architecture, which covers the whole workflow of processes required to fetch, analyze, preprocess, reconcile, and optimize traffic crowdsourced data, is intrinsically modular, and has the advantage of being able to be instantiated as a set of microservices, residing on different and interacting systems. In this way, it is possible to deploy the various components of the platform on-premises, even of non-enterprise grade, or simply in the Cloud, with low maintenance costs and with the obvious advantage of being able to easily scale up each component, due to the additional load that should be necessary to manage.

V. Experiments and Discussion

A. Experiment Setup

The dataset we used consists of data related to the roads from the Centennial area near the city of Denver. Each

event (i.e., moderate jam, heavy jam, and stand still jam) was assigned a unique class, thus making it as a multi-class classification problem. With the goal of building a solution that could scale seamlessly and process huge amounts of real-time data, we chose to adopt Random Forest (RF) as our main method for the aforementioned Model Training module, to help solve or mitigate many of the problems mentioned above, and we also used Support Vector Machine (SVM), and k-Nearest Neighbors (kNN) for the purpose of comparison.

On the other hand we chose not to use other machine learning methods requiring the setup of deep neural networks which cannot deal with real-time streaming data. One of the main benefits of Random Forest is its capacity of handling huge data sets with high dimensionality. It can handle thousands of features and is able to identify the most significant ones (i.e., calculating features' importance), so it is regarded as an effective dimensionality reduction method.

Random Forest provides consistent accuracy and, when large amounts of data are missing, it includes methods for balancing errors in datasets where classes are highly imbalanced. Random Forest is robust to overfitting and does not require to normalize or standardize the dataset, since it is a ensemble learning method based on decision trees.

Data Preprocessing is a critical part of the proposed platform, since the data we are dealing with was crowdsourced. There are many problems with the crowdsourced data, as mentioned in the previous section, for example the platform needs to deal with data outliers and noise that may come from users' conflicting judgements (i.e., users who report stand still jam when the speed is actually high as well as the delay is actually low).

It is extremely important to make an assessment of the traffic reports, which can create noise in the dataset and negatively alter the performance of the final model. In the present study, special care was taken to remove data that were clearly in conflict with metrics measured on the road. In these cases it was decided to remove the data from the dataset, resulting in an increase in the final accuracy of the model.

Thus, we have compared the model performance between the raw dataset obtained directly from the cloud and the preprocessed dataset, after going through the data preprocessing module in the proposed platform. In order to simulate the scenario of this platform running on a cloud computing environment, we specifically chose to run the experiment with an Intel Xeon CPU with an Nvidia Tesla P100 GPU.

B. Results Analysis

This section presents the results analysis of the experiments performed in this study. Specifically, we show the results of the comparison of the Machine Learning models performance between the raw crowdsourced data and the preprocessed data, after going through our proposed pipeline. The raw dataset has a size of 74,119 rows and the preprocessed dataset has a size of 12,123 rows with both of them having the same features. The overall performance difference can be found in Fig. 2. From this bar graph, we can clearly observe that there is a significant performance increase when our preprocessing pipeline is applied to the data. Specifically, there is a 44.37% increase of prediction accuracy for the RF model, a 45.81% increase for the SVM model, and a 59.81% increase for the kNN model, indicating the importance of the need to clean up the crowdsourced data.

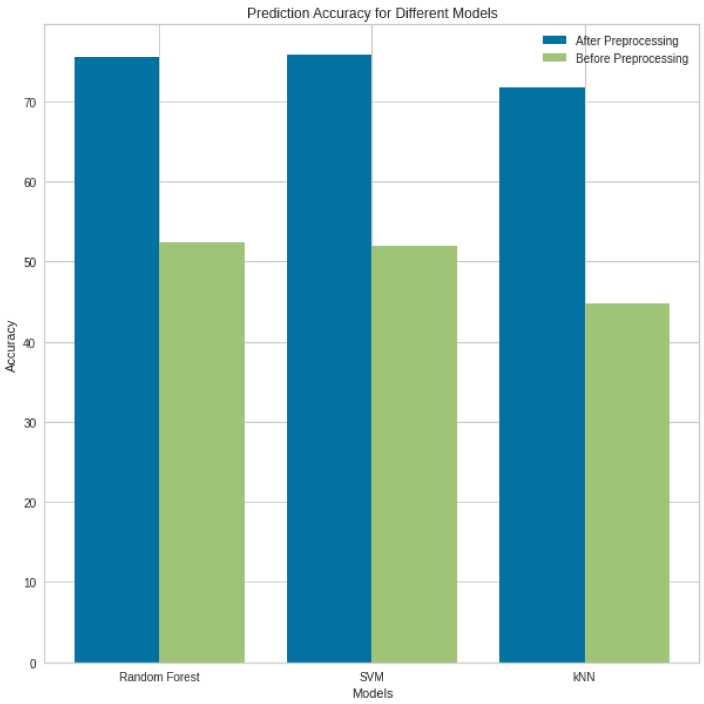

Fig. 2. Overall Performance Comparison between the raw data and the preprocessed data on three different Machine Learning models

The detailed performance metrics can be found in Table V. From Table V, we conclude that when using the correctly preprocessed data, by using simple machine learning models such as Random Forest and SVM can give us relatively acceptable performance compared to other deep learning approaches, while maintaining the property of being operated in real-time. On the other hand, we can also observe that the Macro F1 scores among all these three models are relatively low, indicating that the models are performing very well on the common classes while performing poorly on the rare classes. Fig. 3 reports the confusion matrix of the RF model, with label 1 referring to moderate jam, label 2 referring to heavy jam, and label 3 referring to stand still jam.

TABLE V
MODEL PERFORMANCE WITH PREPROCESSED DATA

Metric Name	Random Forest	SVM	kNN
Accuracy	75.55%	75.78%	71.62%
F1 Score (micro)	75.55%	75.78%	71.62%
F1 Score (macro)	28.69%	28.74%	31.05%
F1 Score (weighted)	65.24%	65.34%	64.49%

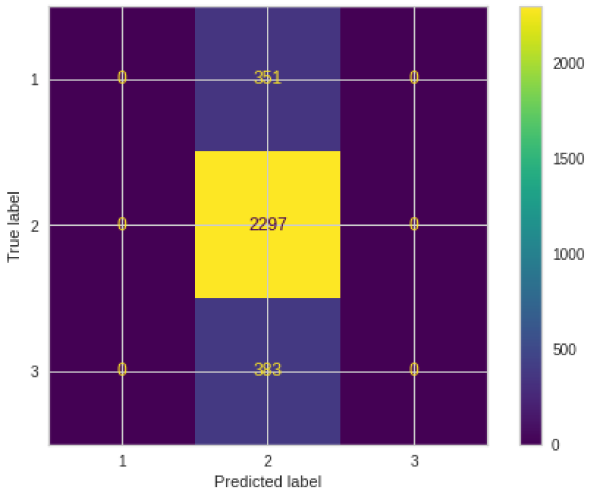

Fig. 3. Confusion Matrix for Random Forest Model

VI. Conclusion

This paper presents a platform for real-time crowdsourced traffic data combined with weather-related data. The proposed solution is a valuable aid, easily deployable, for the processing of crowdsourced data and the definition of scalable and efficient learning models, for the prediction of traffic events. The proposed solution is efficient, in terms of real-time data retrieval, combination and synchronization of different types of data, data reconciliation and elimination of spurious data, and efficient training of the predictive model. The results obtained make the platform in question a valid aid for public or private entities engaged in the prediction of traffic events, and can be easily extended to regional or national geographical contexts, with a benefit in terms of costs and installation time.

This solution can help people make better travel plans, and provide institutions a useful tool for dealing with emergencies, for example for fast alerting of emergency vehicles, or in the event of natural disasters such as storms, landslides, or high winds, which can cause problems in managing public transportation, and efficiently routing traffic in critical situations.

Given the characteristics of the crowdsourced data, we believe several further developments can be made. First of all, given the power of recently developed deep learning frameworks, we are planning to find an efficient way to integrate them into our platform while making sure they can work with real-time streaming data. In addition, we are also planning to study and apply new techniques for dealing with outliers (or the refinement of the existing ones) and new techniques for dealing with highly imbalanced data reported by the users which we cannot control, in order to better clean the dataset from noise and improve the overall accuracy of the prediction model.

Acknowledgment

The authors would like to thank the city of Centennial (CO, USA) for providing the Waze data used in this study.

References

[1] Tafidis, P., Teixeira, J., Bahmankhah, B., Macedo, E., Coelho, M. C., and Bandeira, J., "Exploring crowdsourcing information to predict traffic-related impacts," in *2017 IEEE International Conference on Environment and Electrical Engineering and 2017 IEEE Industrial and Commercial Power Systems Europe (EEEIC / I CPS Europe)*, 2017, pp. 1–6.

[2] Tran Minh, Q., Pham-Nguyen, H.-N., Mai Tan, H., and Xuan Long, N., "Traffic congestion estimation based on crowd-sourced data," in *2019 International Conference on Advanced Computing and Applications (ACOMP)*, 2019, pp. 119–126.

[3] Liu, Z., Chen, L., and Tong, Y., "Realtime traffic speed estimation with sparse crowdsourced data," in *2018 IEEE 34th International Conference on Data Engineering (ICDE)*, 2018, pp. 329–340.

[4] Bogaerts, T., Masegosa, A. D., Angarita-Zapata, J. S., Onieva, E., and Hellinckx, P., "A graph cnn-lstm neural network for short and long-term traffic forecasting based on trajectory data," *Transportation Research Part C: Emerging Technologies*, vol. 112, pp. 62–77, 2020.

[5] Xiangxue, W., Lunhui, X., and Kaixun, C., "Data-driven short-term forecasting for urban road network traffic based on data processing and lstm-rnn," *Arabian Journal for Science and Engineering*, vol. 44, no. 4, pp. 3043–3060, Apr 2019.

[6] Zhang, X., Onieva, E., Perallos, A., Osaba, E., and Lee, V. C., "Hierarchical fuzzy rule-based system optimized with genetic algorithms for short term traffic congestion prediction," *Transportation Research Part C: Emerging Technologies*, vol. 43, pp. 127–142, 2014, special Issue on Short-term Traffic Flow Forecasting.

[7] Zaki, J. F., Ali-Eldin, A., Hussein, S. E., Saraya, S. F., and Areed, F. F., "Traffic congestion prediction based on hidden markov models and contrast measure," *Ain Shams Engineering Journal*, vol. 11, no. 3, pp. 535–551, 2020.

[8] Yang, S., "On feature selection for traffic congestion prediction," *Transportation Research Part C: Emerging Technologies*, vol. 26, pp. 160–169, 2013.

[9] Liu, Y. and Wu, H., "Prediction of road traffic congestion based on random forest," in *2017 10th International Symposium on Computational Intelligence and Design (ISCID)*, vol. 2, 2017, pp. 361–364.

[10] Tseng, F.-H., Hsueh, J.-H., Tseng, C.-W., Yang, Y.-T., Chao, H.-C., and Chou, L.-D., "Congestion prediction with big data for real-time highway traffic," *IEEE Access*, vol. 6, pp. 57 311–57 323, 2018.

[11] Zafar, N. and Ul Haq, I., "Traffic congestion prediction based on estimated time of arrival," *PloS one*, vol. 15, no. 12, p. e0238200, 2020.

[12] Jia, Y., Wu, J., and Xu, M., "Traffic flow prediction with rainfall impact using a deep learning method," *Journal of advanced transportation*, vol. 2017, 2017.

[13] Xu, X., Su, B., Zhao, X., Xu, Z., and Sheng, Q. Z., "Effective traffic flow forecasting using taxi and weather data," in *International Conference on Advanced Data Mining and Applications*. Springer, 2016, pp. 507–519.

[14] Hou, Y., Deng, Z., and Cui, H., "Short-term traffic flow prediction with weather conditions: based on deep learning algorithms and data fusion," *Complexity*, vol. 2021, 2021.

[15] Lee, J., Hong, B., Lee, K., and Jang, Y.-J., "A prediction model of traffic congestion using weather data," in *2015 IEEE International Conference on Data Science and Data Intensive Systems*. IEEE, 2015, pp. 81–88.

[16] Gustavo Niemeyer, "geohash.org," 2008, http://geohash.org/site/tips.html.

[17] Yuan, H. and Li, G., "A survey of traffic prediction: from spatio-temporal data to intelligent transportation," *Data Science and Engineering*, vol. 6, no. 1, pp. 63–85, Mar 2021.

[18] Chawla, N. V., Bowyer, K. W., Hall, L. O., and Kegelmeyer, W. P., "Smote: Synthetic minority over-sampling technique," *J. Artif. Int. Res.*, vol. 16, no. 1, p. 321–357, jun 2002.