International Journal of Advanced Technology and Engineering Exploration, Vol 9(86)
ISSN (Print): 2394-5443 ISSN (Online): 2394-7454
http://dx.doi.org/10.19101/IJATEE.2021.874739

Research Article

Feature-driven label generation for congestion detection in smart cities under big data

Aamish Izhar, Ajay Rastogi, Syed Shafat Ali*, S. M. K. Quadri and S. A. M. Rizvi
Department of Computer Science, Jamia Millia Islamia, New Delhi, India

Received: 06-September-2021; Revised: 19-January-2022; Accepted: 20-January-2022

Abstract
Due to rapid urbanization and the emergence of smart cities, the problem of traffic congestion has materialized into a major issue for smart city planners. Therefore, traffic congestion prediction is needed to effectively reduce traffic congestion and enhance the road capacity. There have been various studies which have tried to solve the problem of traffic congestion. However, it is difficult to properly judge the effectiveness of such studies given the absence of properly labeled datasets. Additionally, current studies use datasets with relatively lesser number of data instances, which does not correctly reflect the big data nature of the traffic data. Motivated by these problems and challenges, in this paper, we aim to study the problem of traffic congestion with respect to effective label-generation under big data perspective. Essentially, we provide two sound and intuitive techniques for label generation which help in the correct annotation of unlabeled data. One of the techniques is based on the number of vehicles plying on the road and the other is based on the amalgamation of average speed and number of vehicles. For this purpose, we consider a publicly available CityPulse traffic dataset with 13.5 million data instances. Using our techniques, we generate "congested" and "not-congested" labels depicting whether there is congestion on the road or not. To tackle the class imbalance problem, besides using random undersampling and oversampling techniques, we also introduce a mixture of the two techniques to negate any bias inherent to two individual sampling techniques. To test the effectiveness of our label generation approaches, we make the extensive use of various machine learning techniques and for performance evaluation we use all the standard classification evaluation metrics. Finally, we compare our techniques with a previous work which only considered average speed for label generation. Our results demonstrate the effectiveness of the proposed approaches against the comparing method. For example, in random undersampling the F1-score of every classifier under the proposed techniques is close to 1, whereas that under the comparing method, F1-score is as low as 0.70 in multinomial naïve Bayes (MNB) classifier and 0.88 in support vector machine (SVM). Similarly, in oversampling, our approaches have a close F1-score of 1 across all the classifiers, whereas the comparing method gets as low as 0.70 in MNB. The same trend can be seen in the mixture of both the sampling techniques.

Keywords
Smart cities, Big data, Label generation, Classification, Traffic congestion.

1.Introduction
Advancements in technology, especially in the field of information and communications technology (ICT), have paved the way for the introduction of the idea of "smart city" [1]. In a smart city, various technologies are utilized to improve efficacy in social services, environmental protection, public safety, transport system, and other related areas. This in turn leads to higher comfort levels of its citizen. However, fast urbanization poses various problems in which the problem of road traffic congestion is a serious one. This congestion, especially in big cities, can lead to the complete immobility of traffic and the related consequences tend to be even more damaging. Hence, it is of immense significance to timely prevent traffic congestion in smart cities.

The prediction of road traffic congestion has an important role in managing the transport system in smart cities. Informed travel decisions can be made with the help of such predictions by improving and providing better traveller information services. Moreover, precise traffic predictions can help improve road safety by preventing accidents and help to improve transportation costs and decrease air pollution [2]. With the recent availability of a large amount of transportation data from various sources,

*Author for correspondence

researchers have been able to examine different traffic patterns and solve various transportation problems. As far as traffic data sources are concerned, researchers have classified them into 6 different categories, as depicted in *Figure 1* [3]. The traffic data provided by these sources can be used for different purposes such as smart city planning, traffic forecasting, preventing accidents, congestion detection, etc.

In recent years, several approaches have been utilized to predict road congestion [4−13]. Some techniques involve statistical methods [14−16], some studies employ clustering techniques [17, 18], while others bank on classical machine learning [4, 5, 7]. More recently researchers have tried deep learning to solve the problem of traffic congestion prediction. However, comparative examination of different techniques is an open challenge as it is hard to decide which method has superiority over the other. The first argument to explain this difficulty in judging the superiority of various models would be that the efficiency of prediction models depends on the attributes of the traffic flow present in the dataset [19]. Furthermore, these models are constructed using specific traffic datasets of relatively smaller sizes. The second argument explaining this difficulty would be the limited availability of traffic datasets in general, especially the labeled datasets, wherein the ground-truth has been maintained and agreed upon. That is to say that if we have a properly labeled traffic congestion dataset (congested or not-congested), any method using such a dataset can be properly judged in its efficiency and can be compared against any other method without any bias. This would, thereby, essentially eradicate the problem explained above. The second argument, therefore in particular, forms the crux of the current study.

To overcome the problem of label generation, a variety of methods have been proposed to generate accurate labels according to different needs. However, sometimes critical attributes are missed while picking an attribute set for the purpose of the label generation. For example, authors in [20], have considered an attribute i.e., vehicle_avg_speed alone for the generation of labels in their dataset. While their results indicate one hundred percent precision score, this should not be the proper reflection of the correctness of their approach. We shall be explaining this in detail in section 3.1. Besides the problem of label generation, which is the main concern of this study, there are various problems in the current state of the literature as far as the problem of traffic congestion prediction is concerned. One of such problems is that of big data. There have been studies which primarily try to solve the traffic congestion problem, while only using very small datasets to train their mode [4−6] [12, 21]. However, given how the traffic system in smart cities tend to work, traffic congestion problem is a big data problem (huge amounts of data generated every minute [22, 23]). Another issue is the small number of classifiers used for model training, which can point towards classifier-bias [4−6] [12, 21]. Finally, studies generally focus on accuracy alone to evaluate the performance of their traffic congestion prediction models [4, 8, 12, 13].

Motivated by the above-mentioned problems and challenges prevalent in the present-day literature, the current study tries to overcome the same and provide a feasible solution. In this study, we propose two methods for label generation which highlight and consider some crucial features to generate the labels for an unlabeled dataset, logically and intuitively. To this end, we utilize the publicly available dataset provided by the CityPulse[1], which contains over 13.5 million instances (big data). For label generation, we utilize *vehicle_load* (number of vehicles) and *vehicle_avg_speed* (average speed of vehicles) as important features, on the basis of 1) *vehicle_avg_speed* (utilized in [20], comparing method), 2) *vehicle_load* (proposed method), and 3) *vehicle_avg_speed* ×*vehicle_load* (proposed method), a combination of *vehicle_avg_speed* and *vehicle_load*. The generated labels for each instance for all of the above 3 cases are assigned as *congested/not-congested* (will be explained in section 3.4). Moreover, we observe that there is a high probability of class imbalance in our label generation approaches. Therefore, we employ three sampling techniques-random undersampling, synthetic minority oversampling technique (SMOTE), and the combination of two (will be explained in section 3.5).

Compared to the other label generation methods, to test the efficacy of our proposed methods, we employ five well-known classifiers, namely, support vector machine (SVM), multinomial naïve Bayes (MNB), logistic regression (LR), random forest (RF), and multilayer perceptron (MLP), where MLP belongs to the class of artificial neural network (ANN). For thorough evaluation, standard classification evaluation metrics, i.e., accuracy, precision, recall and F1-score have been used. The obtained results show that the

[1] http://iot.ee.surrey.ac.uk:8080/datasets.html#traffic

models which are trained on our proposed methods of label generation i.e., *vehicle_load* and *vehicle_avg_speed* ×*vehicle_load* perform better in classifying *congested* and *not-congested* instances against the comparing method [20] which uses *vehicle_avg_speed* to generate labels. It is of importance to note that this study does not only solve the problem of label-generation, thereby allowing any traffic congestion prediction technique to be judged fairly but goes a bit farther. This study goes on and performs an extensive evaluation of various classification methods while using big data and makes use of all the standard classification evaluation metrics to negate any form of bias. Previous studies, as explained above, did not exhibit such endeavour.

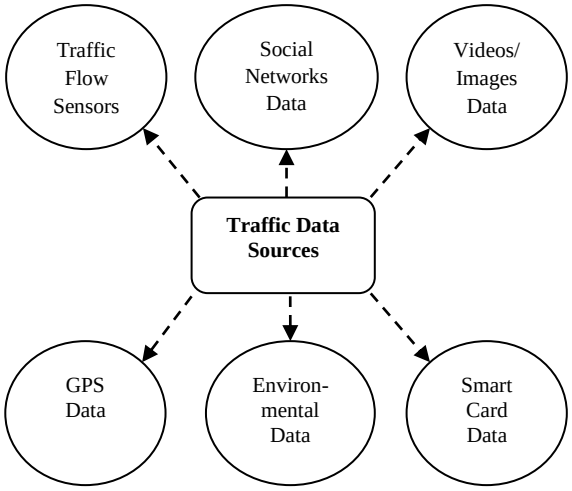

Figure 1 Classification of traffic data sources

The main contributions of this paper are as follows:

- Two novel methods for label generation i.e., *vehicle_load* and *vehicle_avg_speed* ×*vehicle_load* have been proposed in this work.
- To deal with data imbalance problem in generated labels, we have employed random undersampling and SMOTE.
- To overcome the inherent drawbacks of both under and oversampling, a mixture of both the sampling technique has been utilized.
- Big data approach using massive traffic dataset having 13.5 million traffic instances has been utilized for accurate prediction of traffic congestion.
- Extensive experiments using five well-known classifiers, namely, SVM, MNB, LR, RF, and MLP have been performed for label-generation evaluation.
- For thorough evaluation of the models, standard classification evaluation metrics, i.e., accuracy, precision, recall and F1-score have been used.
- Our experimental results show that both proposed methods outperform the existing approach [20] irrespective of the sampling techniques, classifiers and evaluation metrics used.

The remaining paper has been organized as follows. The summary of related work is given in section 2. The proposed label generation approaches are explained in section 3 (subsection 3.1). The overall experimental setup which includes the description of the dataset, the label generation approach, the training of models, and the evaluation metric, has also been covered in section 3 (subsection 3.2 to subsection 3.7). Section 4 covers our results against comparing method. Section 5 discusses the key observations from our results and the limitations of our work. Finally, section 6 summarizes our work with the conclusion and possible future research directions.

2.Literature review

In recent years, several research attempts have been undertaken to address the issue of traffic congestion prediction by providing comprehensive and effective prediction models. A number of prediction approaches, for example, machine learning, statistical, and the combination of both (hybrid) are utilized for prediction. In [24], a kalman filter (KF) based model was used on the real-world traffic data from the social network (Twitter data) for the prediction of vehicle arrival time. The results indicated that the prediction accuracy was increased when more information corresponding to data was fed into the model. An autoregressive integrated moving average (ARIMA) model, which uses the correlation between sequences, was utilized for traffic flow prediction and indicated decent accuracy [14]. Moreover, for comparatively smaller datasets, the ARIMA methods provide good prediction results. Authors in [15] utilized traffic data of three days for the prediction of the traffic flow of the following day by making the use of seasonal ARIMA. This study also showed the advantage of the small amount of data to predict traffic flow.

In [25], a technique for forecasting and classifying for taxi-hailing was proposed, where the results of the K-means clustering were utilized in a neural network (NN) for making the prediction [18]. The results obtained after implementing this system on a real-world dataset, offered by a Shanghai-based taxi

company, verified the reliability of the proposed system. Again in [26], a neural network-based mechanism was employed to predict road congestion. Authors in [11] used SVM to predict bus arrival time using data collected from global positioning system (GPS). For this model, three factors, namely, weather, time, and holidays, were chosen as references. The results demonstrated SVM's ability to produce satisfactory predictions while being adaptive and stable. Furthermore, a prediction model based on RF classifier showed a prediction accuracy of 87.5% [12]. It is important to note that in [12], a very small dataset comprising 1124 instances was used. The aforementioned approaches are only performed under certain scenarios, i.e., the prediction efficiency of these methods is highly affected when the situations are altered, which in turn reduces the prediction accuracy.

Later on, the researcher utilized hybrid prediction models which utilize additive advantages of machine learning and statistical approaches and produce better performance. However, hybrid models use large storage and are computationally expensive. As for hybrid models, the study in [14] proposed a traffic prediction model which combined SVM and ARIMA and performed feature analysis to find out the characteristics of the given data. The results showed improvements in predicting traffic flow. In [27], researchers proposed a hybrid approach using a K-nearest neighbor (KNN) classifier together with an Adelson-Velsky and Landis (AVL) tree for the prediction of short-term traffic flow. In [28], the advantages of the periodic moving average (PMA) model and the ARIMA model were incorporated into a hybrid model. The model assessment was performed employing both the historical and real-time data, while the results indicated better performance. In [16], radial basis function-artificial neural network **(**RBF-ANN) and ARIMA were combined for the prediction of traffic flow. The obtained results showed how ARIMA and RBF-ANN may be outperformed by a hybrid model when compared individually.

More recently, authors in [4] used a very small set of data instances generated from sensor devices to predict the traffic condition a day ahead based on weather conditions. The experiments were performed using RF classifier. The researchers argue that since there was no class imbalance, only one evaluation metric, i.e. accuracy was used. The results showed that model could achieve average accuracy of more than 50%. Also, the results depend upon the accurate data collection from the sensors. Furthermore, authors in [5] used Python, SQL, machine learning, geo information system (GIS) and big data technologies to develop a decision tree (DT) model for traffic prediction and compared it with LR model. The results showed that DT model was more accurate than LR model. Moreover, when the same dataset of around five thousand instances was analyzed using big data technologies, an increase in the accuracy was observed. It shows that size and quality of input data plays a significant role in the model's results and accuracy. In [7], authors have used estimated time of arrival (ETA) approach to predict traffic congestion. They have also considered various features such as time period, weather and holidays. The traffic state was bifurcated into 5 different categories which ranged from smooth to blockage. The data was collected from Google Maps application programming interface (API. The results showed that RF had the highest accuracy of 92% followed by extreme gradient boosting (XGBoost) which took more training time than RF.

Long short term memory (LSTM) models, in both univariate and multivariate scenarios, have been used to predict congestion over a 5 minute time duration considering the observed and predicted value of vehicle speeds [6]. The results showed that both univariate and multivariate models were able to detect congestion with a varied accuracy between 84% and 95%, with the accuracy depending on the road network layout. The experiments were performed on a small dataset having 1015 samples per week. In [8], authors also developed an LSTM model for traffic prediction and compared it with 2 classic models, namely, the back propagation neural network (BPNN) model and ARIMA model. They have used the actual traffic data from OpenITS[2]. The results showed that their proposed model was able to achieve better prediction accuracy than the classic models. Furthermore, a multi-dimensional LSTM (MDLSTM) model was proposed based on multi-dimensional data considering the factors such as weather, holidays and working day [9]. The results showed that mean absolute error (MAE) of MDLSTM got reduced by 5.528 and 3.018, and root mean square error (RMSE) reduced by 6.827 and 2.154 respectively, in comparison to BPNN and LSTM models. In [10], authors have applied LSTM model on clustered data for traffic prediction. The data was taken from CityPulse. Three models were constructed: 1 to 1, many to 1, and many to many.

[2] https://www.openits.cn/

After evaluation the results showed that (many to many) model was the best model among the three where the mean square error (MSE) and MAE were reduced by 30% and 21% respectively.

In [13], three recurrent neural networks (RNN) (a standard RNN, LSTM, and a gated recurrent unit (GRU)) were implemented using real-world data, with varying number of layers. Besides, a new performance metric was also proposed by authors, namely, standardized accuracy and time score (STATS), to evaluate the models. Based on this metric, the results outcome showed that GRU model had the best overall performance with good accuracy score. Authors in [21] adapted convolutional neural networks (CNN) and used CNN-GRU-attention model and CNN-LSTM-attention model in a combined form, namely, combined deep learning prediction (CDLP) model for short-term traffic flow prediction. Real world traffic dataset of three consecutive months was used having 25920 instances. The results clearly indicated a higher accuracy for CDLP model than the other baseline models. Various machine learning and deep learning models, consisting of ridge regression, vector auto regression (VAR), LSTM and convolutional LSTM were implemented for traffic prediction using publicly available traffic dataset from the cities of U.K. and New York [29]. The results showed that convolutional LSTM performed better on both the datasets than the other models. A hybrid approach was used in [30], where boosted long short-term memory ensemble (BLSTME) and CNN were integrated to predict congestion. The model was implemented and evaluated in a simulated environment using SUMO[3] and OMNeT++[4]. The results showed 10% better precision, recall and accuracy than the comparing methods.

Additionally, a real-time traffic congestion prediction system was proposed for a metropolitan city [17]. The data, collected from the Google Maps, was fed into the system and K-means clustering was implemented to detect traffic condition. The results showed that the vehicle moving towards the destination, can automatically sense the traffic conditions and was able to change the path accordingly, using the proposed system. In [31], for short-term traffic prediction, authors used various state-of-the-art models through hyperparameter optimization. Many different classifiers, namely, MLP, decision jungles (DJ), local deep support vector machine (LD-SVM), and CN2 rule induction were used. Data used in the study was obtained through 'VISSIM'[5], a traffic simulator. The results showed that DJ outperformed and was more robust than other classifiers. In [32], authors used the data from Google Map's traffic layer and implemented various models for traffic prediction, which included historical averages (HA), ARIMA, support vector regression (SVR) and SVR-Graph. The results showed that these simple models were able to detect congestion effectively on the collected data.

Having studied and discussed state-of-the-art literature relevant to this study, we can see that various techniques in different contexts have been employed for effective traffic congestion prediction. Some techniques involve statistical methods [14−16], some use clustering [17, 18], others resort of classical machine learning [4, 5, 7]. More recently researchers have tried deep learning involving the same problem with relatively smaller traffic datasets [6, 9, 10, 13]. Different situations or contexts involve various traffic conditions in varied places [4, 7, 9, 11]. Having said that, one can easily see the corresponding shortcomings. Such shortcomings range from the small number of related evaluation metrics used for evaluating the performance of the models to the biased selection of classifiers and very small size of the datasets. For example, [4, 8, 12, 13] only use accuracy as the evaluation metric to judge the performance of their models. It is needless to say that accuracy alone can be a misleading metric in a multi-class problem where one has to take care of false negatives and false positives. Furthermore, studies [4−6, 12, 21] use very small datasets to train their models whereas, given the way traffic operates, traffic congestion prediction is a big data problem. Additionally, there are studies [4, 5, 12] which only use a few classifiers for the prediction of traffic congestion, where the problem of classifier-bias cannot be ignored. Besides, while some studies [5] make use of big data technologies to predict traffic congestion, the data they operate on is nevertheless small in nature. In this study, therefore, we recognize and overcome all these shortcomings. We employ as many as 5 classifiers, i.e., SVM, MNB, LR, RF and MLP, use all the standard evaluation measures, i.e., accuracy, precision, recall and F1-score, and use big data technologies on an actual big data (as many as ~13.5 million data instances). Besides these, we propose label generation techniques using big data

[3] https://www.eclipse.org/sumo/
[4] https://omnetpp.org/
[5] https://www.ptvgroup.com/en/solutions/products/ptv-vissim/

while keeping class imbalance in check, which forms the crux of this study as explained in section 1.

3.Methods

In this section, we provide a detailed explanation of the proposed approaches in section 3.1, experimental setup in section 3.2, dataset used in this study in section 3.3, label generation process in section 3.4, countering the class-imbalance problem in section 3.5, choice of classifiers in section 3.6, and model training procedure, and evaluation in section 3.7.

3.1Proposed approaches

The issue with the annotation of traffic data is that the traffic data of a particular road is provided together with vehicles running on it alone. Therefore, there is no clear understanding of the handling of such data in context to traffic congestion. As mentioned in section 1, there are some methods that attempt to label data using certain feature(s). However, such methods have significant flaws [12, 20]. Although these methods may seem to perform well, we cannot consider these results to be the correct reflection of what happens in the real world. This is because the assumptions for label generation and the process of label generation itself are dubious. For example, [20] utilizes the feature *vehicle_avg_speed* which indicates the average speed of vehicles running on a road in order to produce labels. The shortcoming of their approach is that a very low average speed can indicate congestion. However, if we consider another parameter i.e., vehicle count, to be very low, intuitively we can understand that the probability of congestion can be considered as low.

Inspired by these shortcomings, this paper proposes a novel approach for label generation that takes into account the combined effect of two important features i.e., *vehicle_load* and *vehicle_avg_speed* for generating labels logically and intuitively. In addition, we also propose another label generation method based on *vehicle_load* where we generate the labels independently.

It is understood that the probability of data imbalance (in terms of instance labels) in our approaches is quite high (as we shall see later in section 3.4). To circumvent that, we employ three sampling techniques - random undersampling, SMOTE and the combination of two (will be explained in section 3.5). We do so in order to deal with data imbalance on the generated labels, that is, roughly the same no. of the *congested* and *not-congested* instances of traffic data.

Afterwards, several machine learning techniques are employed for the classification and prediction of traffic congestion on the basis of several discriminating features and the effectiveness of our approaches are analyzed. *Figure 2* shows the logical framework of the employed methodology.

Data processing, model training, and evaluation have been performed using Apache Spark[6], which is a distributed framework and is designed for cluster computing while being open-source. Spark offers the ability to do the processing in-memory which is more efficient and much faster than MapReduce's disk-based processing as done in Hadoop[7] [33]. Spark is considered highly accessible as it provides simple application programming interfaces (APIs) in Python, Java, Scala, and structured query language (SQL), and many rich built-in libraries. It is easy to integrate Spark with other big data platforms like Hadoop. Furthermore, Spark supports rich functionality features, such as Spark SQL, used for processing structured data, Spark Streaming, used for live data streams processing, MLlib, used for machine learning, and GraphX, used for graph processing.

3.2Experimental setup

All the experiments have been performed on a standalone machine, having Intel Xeon Silver 4114 CPU and 64 GB of RAM.

3.3Dataset description

This study utilizes the publicly available unlabeled traffic dataset of Aarhus city in Denmark. This dataset comes in a CSV format in its raw form provided by the CityPulse. This dataset has already been used in various studies [34−36].

For the present study, the raw dataset available from Feb 2014 - June 2014 has been used. This dataset consists of several observations collected between two points for the aforementioned period. The dataset consists of a total of 9 features, namely, "*avgSpeed*", "*vehicleCount*", "*avgMeasuredTime*", "*status*", "*REPORT_ID*", "*exitID*", "*medianMeasuredTime*", "*X_id*" and "*TIMESTAMP*". Among the 9 features, the following 4 features are utilized - "*avgSpeed*" (*vehicle_avg_speed*) in km/h, "*vehicleCount*" (*vehicle_load*), "*medianMeasuredTime*" and "*avgMeasuredTime*", for experimental analysis. The dataset statistics is provided in *Table 1*, which

[6] https://spark.apache.org/

[7] http://hadoop.apache.org/

includes the no. of instances, the no. of features, and no. of features selected for the given dataset.

Table 1 Dataset description

Number of instances	13,577,132
Number of features	09
Features selected	04

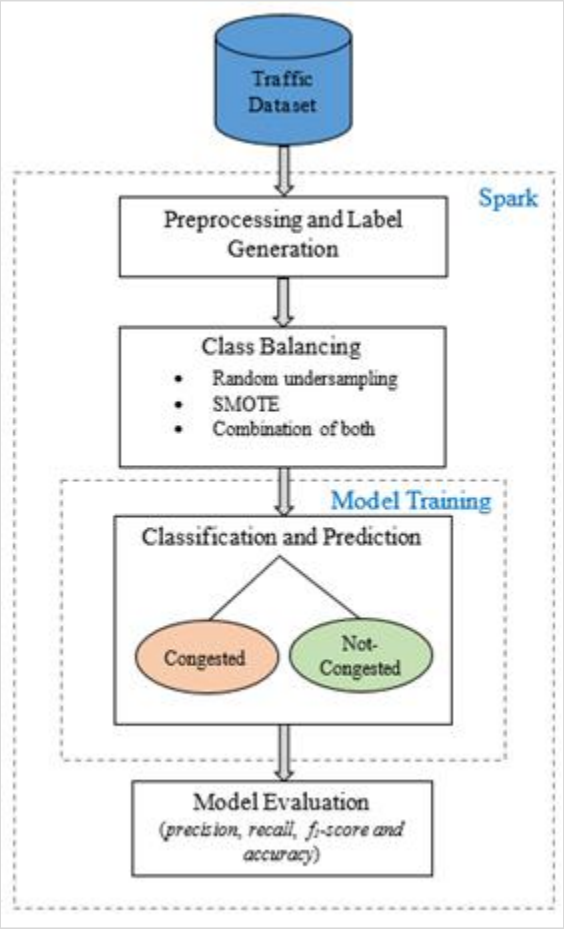

Figure 2 Framework of proposed methodology

3.4Label generation process

As for the current study, among the above 4 features, only 2 features that is, *vehicle_load* and *vehicle_avg_speed* are utilized for label generation. As these 2 features can collectively discriminate congestion from not-congestion, they are of significant importance, taken together. Intuitively, suppose there are *N* number of vehicles on a road, and given that *N* is quite high, we are to understand that if these vehicles have a higher average speed *S*, we can safely suppose that traffic congestion probability is quite low. Conversely, the probability of road congestion would be high if *vehicle_avg_speed S* low. In the same way, when we have a low *N* and low *S* as well, congestion could be possible, while if *S* is high (keeping *N* low), the probability of road congestion is low.

Therefore, it should be noted, keeping the discussion above in perspective, that while there are certain assumptions in play here, our assumptions are logical, sound, and quite safe as compared to [20].

Formally, for each given vector of a data instance, $V_i = f_{1i}, f_{2i}, f_{3i}, f_{4i}$, first, we take the hybrid of features *vehicle_avg_speed* and *vehicle_load*, which results in a new feature i.e., *vehicle_avg_speed* ×*vehicle_load*. To normalize this feature, we divide each feature value by the maximum feature value, so that the values fall in the interval [0, 1]. After normalization, the data instance will be considered *congested (1)* when the resulting normalized feature value is ≤ 0.5, and *not-congested (0)* otherwise, as mentioned in Equation 1.

$$Label(V_i) = \begin{cases} 1, normalized(vehicle\_avg\_speed \times vehicle\_load) \\ \qquad\qquad \leq 0.5 \\ 0, \qquad otherwise \end{cases} \quad (1)$$

After implementing the process above on the whole dataset, we get a total of 4,677 *congested* and 13,572,455 *not-congested* instances, which gives rise to the data imbalance issue. In order to solve this problem, we use several sampling techniques which are explained in section 3.5.

In addition to this, we also generate two more types of labels for our dataset – one using *vehicle_avg_speed* feature directly (implemented in [20]) and the other using *vehicle_load* feature (another proposed approach of this study). The process for generating labels based on *vehicle_avg_speed* is similar to *vehicle_avg_speed*×*vehicle_load*. For *vehicle_load*, the data instance will be considered *congested (1)* when the resulting normalized feature value is ≥ 0.5, and *not-congested (0)* otherwise, as mentioned in Equation 2.

$$Label(V_i) = \begin{cases} 1, & normalized(vehicle\_load) \geq 0.5 \\ 0, & otherwise \end{cases} \quad (2)$$

Table 2 provides the label statistics for all 3 methods of label generation.

Table 2 Congested & not-congested dataset instances

Label generation methods	*Congested*	*Not- congested*
vehicle_avg_speed	12,551,990	1,025,142
vehicle_load	2,699	13,574,433
vehicle_avg_speed ×*vehicle_load*	13,572,455	4,677

3.5Overcoming the problem of class-imbalance

The distribution of various features used in different label generation methods can be seen in *Figure 3*. We can see the heavy skew of *vehicle_load* feature and comparatively slighter skew of *vehicle_avg_speed* feature. Hence, there will be data imbalance in generated labels from *vehicle_load*. Therefore, combining it with *vehicle_avg_speed* will lead, as seen in *Figure 3*, to somewhat skewed data. This, again, poses the problem of class imbalance. In machine learning, the issue of class imbalance is a significant one, as it tends to be biased towards the majority class [37]. Therefore, to overcome this challenge, two well-known sampling techniques – random undersampling technique and SMOTE, which is a state-of- the-art oversampling technique, have been employed [38].

In case of random undersampling, let *k* be the no. of minority class instances, while *p* be the no. of majority class instances, *k* instances are randomly picked from the majority class, keeping in view the uniform probability of each pick, so that the number of minority class instances matches that of the majority class. Thus the balanced dataset will consist of roughly the same number (*k)* of *congested* and *not-congested* instances. SMOTE is an oversampling technique that generates synthetic samples from the minority class of the dataset. The sample data points' nearest neighbors are identified and based on the distance between a sample point and its neighbor, synthetic samples are generated.

In this paper, we have solved the data imbalance problem using three approaches. In our first approach, we undersample the data instances from the majority class using the random undersampling technique as explained above. In the second approach, we first scaled the minority class up using SMOTE such that the number of minority class instances are doubled and then scaled the majority class down to the level of minority class using random undersampling. Therefore, in this approach, we use the combination of oversampling and undersampling, and we simply refer to this approach as "oversampling with hundred percent minority", in the rest of the paper. In our third approach, we scale the minority class, using SMOTE, and right up to the level of the majority class, and we simply call this approach as "complete oversampling", in the rest of the paper.

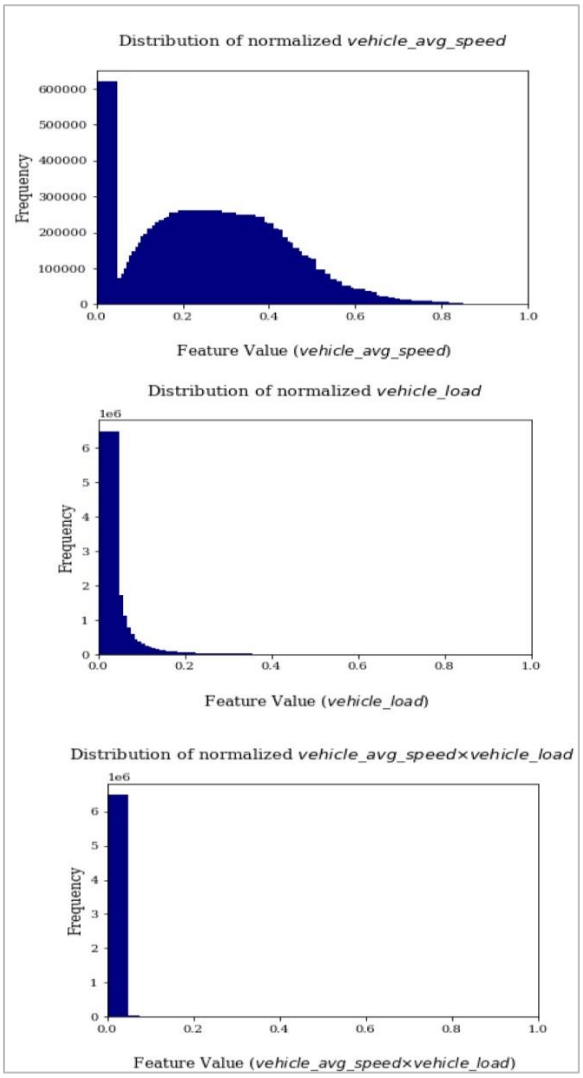

Figure 3 Features distribution on the dataset

The reason we choose three different approaches for sampling and data balancing is to negate any bias inherent to any of the sampling techniques. *Table 3*, *Table 4* and *Table 5* provide the summary of the number of instances generated by random undersampling, oversampling with hundred percent minority, and complete oversampling, respectively.

Table 3 Number of instances generated by random undersampling

Label generation methods	Congested	Not- Congested
vehicle_avg_speed	1,025,142	1,025,142
vehicle_load	2,699	2,699
vehicle_avg_speed ×*vehicle_load*	4,677	4,677

Table 4 Number of instances generated by oversampling with hundred percent minority

Label generation methods	Congested	Not- Congested
vehicle_avg_speed	2,050,284	2,050,284
vehicle_load	5,398	5,398
vehicle_avg_speed ×*vehicle_load*	9,354	9,354

Table 5 Number of instances generated by complete oversampling

Label generation methods	Congested	Not- congested
vehicle_avg_speed	12,551,990	12,551,990
vehicle_load	13,574,433	13,574,433
vehicle_avg_speed ×*vehicle_load*	13,572,455	13,572,455

3.6Classifiers used

In this paper, we use five well-known classification models, namely, SVM, MNB, LR, RF, and MLP, where MLP belongs to the class of ANN. We pick these classifiers to comprehensively evaluate the proposed methods of this study, besides the fact that these classifiers are well-known for their effectiveness towards binary classification problems. All these classifiers are implemented in Spark using *PySpark* which is the Python API for Spark and its *pyspark.ml* package. The classifiers are elaborated as below:

3.6.1Support vector machine

SVM is a supervised machine learning algorithm that is used to address the issues of classification and regression problems. It is widely used in classification analysis. SVM analyzes the data and classifies it into one of the two categories. The key purpose of SVM is to obtain the optimal hyperplane of separation such that the overall width of the margin between the training data is maximum. The obtained optimal hyperplane can then be used for the unseen training data as a generalized solution. Also, SVM can be applied for any number of dimensions [39]. SVM provides one of the most robust methods of prediction.

To address the nonlinear problem of traffic flow prediction, SVR can be used, which maps the problem from a low dimensional space to a high dimensional feature space. A function can be defined which on a given pattern can predict the future data. Let us say we have the training dataset:$\{(x_1, y_1), \ldots (x_n, y_n)\}$. A function can be defined through SVR which can show the relationship between *x* and *y* and predicts the new value of *x*, as in Equation 3:

$$f(x) = w.\varphi(x_i) + b \quad (3)$$

where *w* and *b* are the regression parameters needed to be determined, and $\varphi(x_i)$ is a mapping to high-dimensional space, which is nonlinear.

3.6.2Multinomial naïve bayes

MNB model is one of the naïve Bayes classification models which are based on Bayesian classification methods. These follow Bayes's theorem, which is an equation that describes the relationship of statistical quantities with conditional probabilities. A model of Naive Bayes assumes that, given any class, each of the features it uses is conditionally independent of each other. It gives probability $P(c|i)$ where *c* denotes the class of possible outcomes and *i* denotes the given instance which has to be classified, depicting some certain features [40] (Equation 4).

$$P(c|i) = P(i|c) * P(c)/P(i) \quad (4)$$

A particular instance of a naïve Bayes classifier using a multinomial distribution for each of the features is the MNB classifier.

3.6.3Logistic regression

LR is a technique for estimating the value of a dependent variable (usually the most significant variable) based on the value of an independent variable. Data is entered into the regression model, which is then used by a logistic function to estimate the categorical dependent variable of interest [41].

3.6.4Random forest

RF is a data mining methodology for dealing with classification and regression issues. Classification accuracy has improved dramatically as a result of increasing an ensemble of trees and determining the class category by voting. These ensembles are grown

through the construction of random vectors. Each tree is made up of one or more random vectors. Classification and regression trees make up RF. The output of trees is analyzed to solve classification problems. The RF prediction is determined by the majority of class votes [42].

3.6.5Multilayer perceptron

MLP is based on the feed-forward ANN, consisting of nodes arranged in layers. Each layer in the network is fully connected to the next layer. MLP consists of at least three layers: input, output, and one or more hidden layers. The input layer is made up of nodes equal to the input of the dataset. The output layer is a single node, the value of which is the output of the MLP [43].

3.7Training of models and evaluation metrics

This study examines the efficacy of 3 methods of label generations, that is, 1) *vehicle_avg_speed* feature used in [20], 2) *vehicle_load* feature which is one of the proposed approaches, and the combination of *vehicle_load* and *vehicle_avg_speed* features (*vehicle_avg_speed*×*vehicle_load*), which is another proposed approach (explained in section 3.4). To remove any potential bias in model training, exogenous or endogenous, due to the implementation of the various sampling techniques (section 3.5), 10 independent runs are executed, for *vehicle_avg_speed*, *vehicle_avg_speed* ×*vehicle_load*, and *vehicle_load*, separately. For all independent runs, the seeds are randomly chosen and 10-fold cross- validation (CV) is employed. Finally, the average of these 10 independent runs is reported.

The performance of our models is evaluated by using four well-known evaluation metrics, i.e., accuracy, precision, recall, and F1-score.

Accuracy (acc): is the ratio of correct predictions over the total predictions made, given by Equation 5.

$$Accuracy\ (acc) = \frac{tp+tn}{tp+fp+fn+tn} \quad (5)$$

Precision (p): is the ratio of correctly predicted positives instances over the total positive predictions made, given by Equation 6.

$$Precision\ (p) = \frac{tp}{tp+fp} \quad (6)$$

Recall (r): is the ratio of correctly classified positive instances to the total correctly classified positive and negative instances, given by Equation 7.

$$Recall\ (r) = \frac{tp}{tp+fn} \quad (7)$$

Precision and Recall both are valuable indicators of prediction success for highly imbalanced classes.

F1- Score: is the harmonic mean between precision and recall values, given by Equation 8.

$$F1 - Score = \frac{2*p*r}{p+r} \quad (8)$$

where *tp* and *tn* represent the number of correctly classified positive and negative instances, while *fp* and *fn* represent the number of negative and positive instances that are misclassified.

4.Results

In this section, we present our results which are obtained after we train each of the five models, namely, SVM, LR, MNB, RF and MLP. The results are presented using 4 commonly used evaluation metrics - accuracy, precision, recall, and F1-score. *Figure 4*, *Figure 5*, and *Figure 6* show the results for random undersampling, oversampling with hundred percent minority, and complete oversampling (section 3.5), respectively.

It can be seen from *Figure 4(a)*, that SVM model trained on both the proposed methods of label generation, i.e., *vehicle_load* and *vehicle_avg_speed*×*vehicle_load*, outperforms the model trained on the *vehicle_avg_speed* [20] across all the metrics. While the precision for *vehicle_load* (1) is slightly better than that of *vehicle_avg_speed* ×*vehicle_load* (0.98), this is well compensated by recall where *vehicle_avg_speed*×*vehicle_load* (1) performs better than *vehicle_load* (0.99). The same pattern can be seen for MNB classifier (*Figure 4(b)*), where both of our proposed approaches have similar performances in terms of accuracy, precision, recall and F1-score. However, the F1-score seems to favor the higher precision, resulting in a better F1-score for *vehicle_load* (0.97) as compared to *vehicle_avg_speed* ×*vehicle_load* (0.92). Nevertheless, the comparing label generated technique of *vehicle_avg_speed* [20] is the worst performer among the three label generation methods across all the evaluation metrics in both SVM and MNB. These results and the results to follow suggest that *vehicle_load* is a critical feature in determining road congestion that has not been given any attention in the previous work [20].

From *Figure 4(c)* and *Figure 4(d)*, we observe that precision is again in favor of both the proposed methods for LR and RF classifiers (1) as compared to *vehicle_avg_speed* (0.88 in LR and 0.98 in RF). Although recall for *vehicle_avg_speed* (1 for both the classifiers) is as good as the proposed methods, its F1-score is considerably on the lower side (0.93 in LR and 0.99 in RF), again giving an edge to our

proposed methods. For MLP (as can be seen in *Figure 4(e))*, the trend is quite similar to SVM and MNB as both the proposed methods significantly outperform the comparing label generation technique, *vehicle_avg_speed*, across all the evaluation metrics. Therefore, from the above discussion, we can see that both *vehicle_load* and *vehicle_avg_speed*×*vehicle_load* thoroughly outclass *vehicle_avg_speed* [20] across all the classifiers, when the data was undersampled. Having said that, it is important to note that the comparing method has the best performance in RF classifier among all the classifiers.

Figure 5 presents the results when the data is oversampled with hundred percent minority (where both oversampling and undersampling are used). From *Figure 5(a)*, again for SVM classifier, interestingly a similar result has been obtained as was obtained for random undersampling for SVM (*Figure 4(a)*). All the evaluation metrics, i.e., accuracy, precision, recall and F1-score produce a score of 1 (100%) for both the proposed approached while that does not seem to be the case with the comparing method. Having said that all the three label generation approaches either have improved or similar scores for SVM when contrasted against the corresponding results found in random undersampling. Moreover, from *Figure 4(b)*, in the case of MNB classifier, our results significantly support the proposed method as compared to the *vehicle_avg_speed* [20], which is performing poorly across all the metrics. Again, this pattern was also seen in the corresponding results of random undersampling. However, there is no apparent improvement in the results with MNB unlike SVM.

As was found with undersampling (*Figure 4(c))*, LR model trained on oversampling with hundred percent minority data has a similar performance (*Figure 5(c)*), where averge_speed is as good as the proposed method in terms of recall (1), but performs comparatively poorly for the rest of the three metrics. Interestingly, for RF classifier in *Figure 5(d)*, all the three methods perform equally well and achieve significantly higher scores for all the four metrics. Having said that, the comparing method still does not outshine the proposed methods. As for MLP classifiers, from Figure *5(e)* the results clearly support *vehicle_load* and *vehicle_avg_speed*×*vehicle_load* against *vehicle_avg_speed*. Individually, the performance of both the proposed methods, *vehicle_load*, and *vehicle_avg_speed*×*vehicle_load* is quite similar across all the classifiers and evaluation metrics. One more important observation is that the comparing method persistently produces lower scores of accuracies and precision which in turn hampers its F1-score.

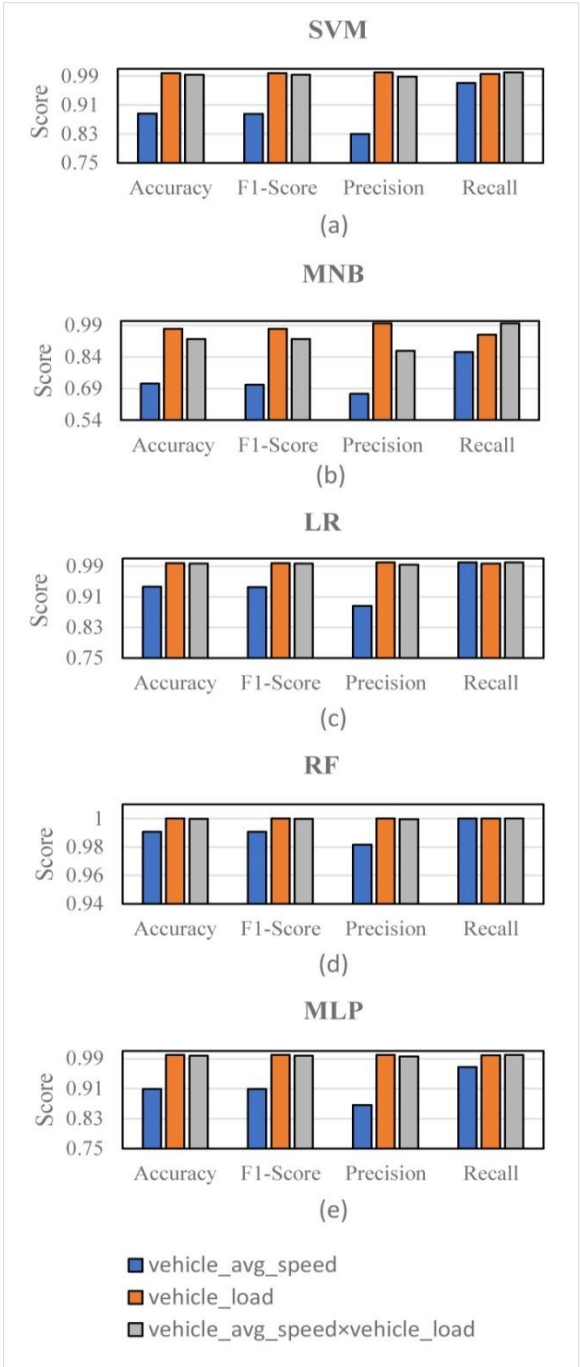

Figure 4 Results on prediction of traffic congestion using different classifiers with random undersampling

From the discussion above, again we can see how our proposed methods are better label generators as compared to *vehicle_avg_speed*, which has been validated by the prediction of traffic congestion results.

Figure 6 shows the results of complete oversampling using SMOTE [38]. For SVM, LR, and RF in *Figure 6(a), Figure 6(c)* and *Figure 6(d)* respectively, we observe a similar pattern which was seen with random undersampling (*Figure 4*) and oversampling with 100% minority data. While recall for *vehicle_avg_speed* seems to be as good as the recall of the proposed method, the latter completely outperforms *vehicle_avg_speed* in terms of precision, F1-score, and accuracy. It is interesting to note that, individually, the performance of both the proposed methods, *vehicle_load* and *vehicle_avg_speed*×*vehicle_load* is quite similar across all the classifiers and evaluation metrics. As for as MNB (*Figure 6 (b)*) and MLP (*Figure 6(e)*) are concerned, we observe a similar trend in both the classifiers, where both the proposed methods are outperforming *vehicle_avg_speed* across all the evaluation metrics. It is noteworthy, however, that for the comparing method, *vehicle_avg_speed*, the accuracy for RF is the highest in contrast to the accuracy found in undersampling and oversampling with 100% minority data. Also, it exhibits as good scores for the rest of the metrics as produced by the proposed techniques.

5.Discussion

From the results discussed above, we happen to experience some of the very intriguing observations, which we shall now try to understand, argue with reasons and discuss briefly. This discussion can also serve as a means of comparative analysis of various factors observed above.

Observation 1. Firstly, we observe that the classification models trained on the labels generated using *vehicle_load* feature alone tend to produce better results than models trained on *vehicle_avg_speed* alone. Besides, when used together with *vehicle_avg_speed*, *vehicle_load* seems to improve the performance of traffic congestion prediction, essentially acting as a booster or a catalyst for *vehicle_avg_speed*. As was seen in the results observed above, this observation seems to be true irrespective of the sampling method or the classifier used. This in turn, means *vehicle_load* is an essential feature for label generation when it comes to traffic congestion prediction in smart cities. By extension, *vehicle_avg_speed*×*vehicle_load*, becomes an essential feature set to generate labels for an unlabeled traffic dataset. The reason for the same was argued in section 3.1 where we argued that if the vehicle count is high, it is intuitive to think that the probability of traffic congestion is high.

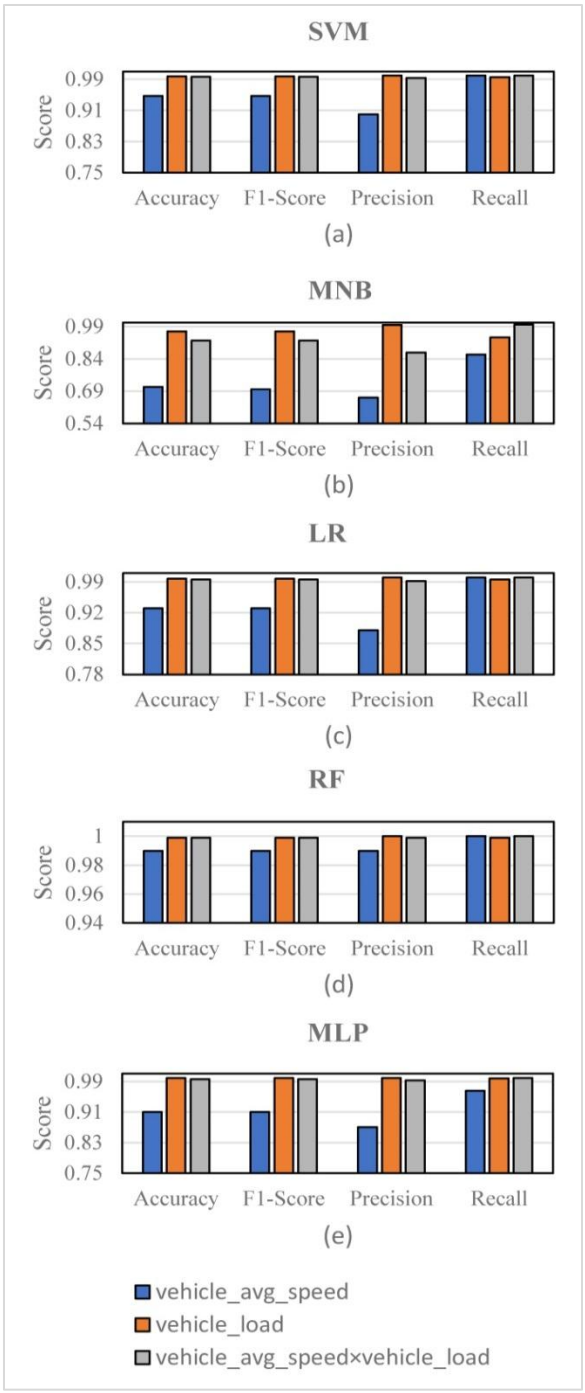

Figure 5 Results on prediction of traffic congestion using different classifiers by oversampling with hundred percent minority

Consequently, the significantly good classification results in classifying *congested* and *not-congested*, go on to verify our rationale and intuition behind our proposed approaches.

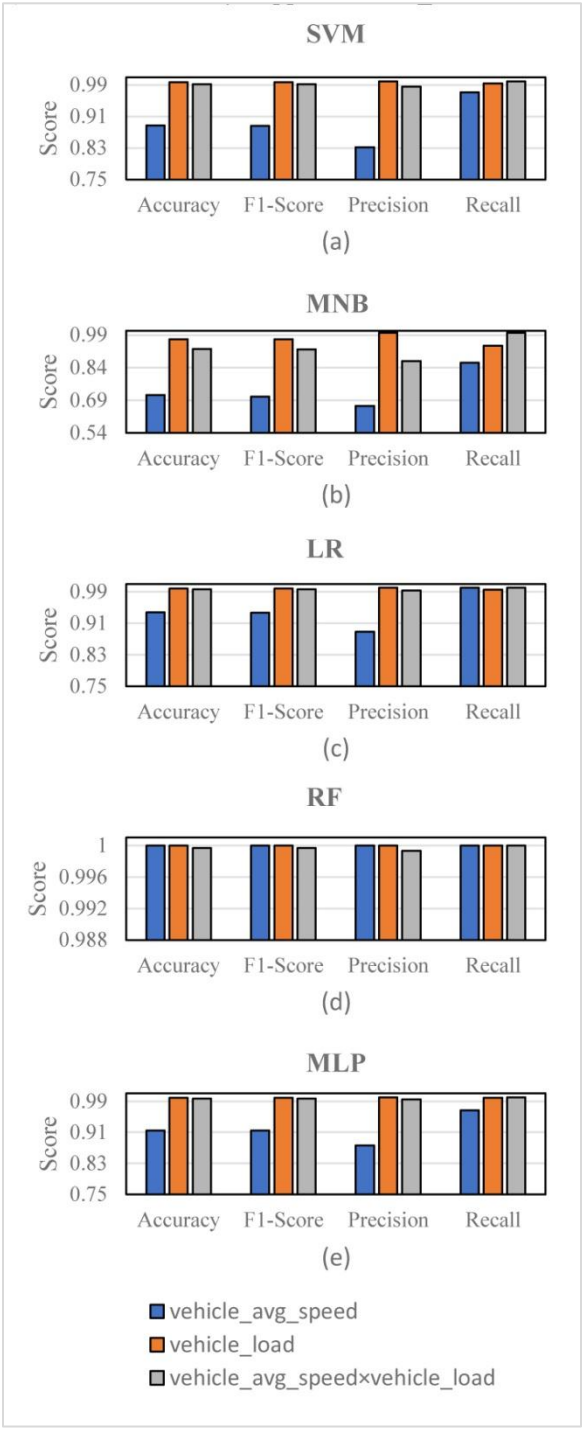

Figure 6 Results on prediction of traffic congestion using different classifiers by complete oversampling

Observation 2. Secondly, we observe that each of the three sampling techniques used to solve the class imbalance problem, produce somewhat similar results across all the classifiers and evaluation metrics. This means whether we randomly downscale the majority class to match the size of the minority class or upscale the minority to match the majority or try both upscaling and downscaling, the model training does not have much of an impact. There can be several reasons for the same. The first reason for this can be lesser variance in the data. The second reason can be the two classes being tightly-knit individual clusters of data, while the third reason can be embedded in the fact that minority classes are actually significantly smaller in number than the majority class.

Observation 3. Our third intriguing observation from the results above is that all the classifiers tend to follow a similar pattern for each of the three label generation method, again irrespective of the sampling technique employed. This essentially means that irrespective of the number of minority or majority class instances (albeit, balanced), models are trained equally well. The reason for this is closely related to one of the reasons for Observation 2. If the two classes are tightly-knit clusters individually, the size of the data should not matter for training the models, whether the data is upscaled or downscaled.

Observation 4. Fourthly, we observe that while the comparing method, i.e., *vehicle_avg_speed*, remains the worst performing label generation approach, it has a better performance (when compared against itself) in RF in contrast to the rest of the classifiers. It is noteworthy, we had raised a concern in the introduction section (section 1) regarding the classifier-bias present in the present-day literature. This observation further goes on to prove one of the important needs of this study in which we have been able to show how a certain classifier can produce better results while there is a danger of ignoring all the other relevant classifiers.

Observation 5. Fifthly, we observe that the models trained on the labels generated by the comparing method, *vehicle_avg_speed*, generally produces high recall and low precision, which in effect impacts its F1-score. This means that this method has the tendency to generally misclassify negative classes (*not-congested*) as positive classes (*congested*), i.e., low precision. Also, high recall means the method tends to generally misclassify positive classes (*congested*) as negative classes (*not-congested*). It is hard to argue which class is more important and such an argument is highly contextual, i.e., sometimes we might prefer precision more than recall and vice-

versa. Therefore, a balanced view, i.e., F1-score, is considered, which is far superior in the proposed methods. Furthermore, the same can be said of accuracy metric, which keeps on producing relatively and significantly low scores. This again goes onto show that *vehicle_avg_speed* alone is not a good label-generation feature/technique.

On a concluding note, we can say that both *vehicle_load* and *vehicle_avg_speed* are crucial factors when it comes to traffic congestion prediction in smart cities. While the *vehicle_avg_speed* feature was used in [20], our results suggest that *vehicle_load* (proposed approach) can be more efficient to predict congestion of traffic against *vehicle_avg_speed*. Furthermore, another proposed hybrid approach that utilizes the above two features, i.e., *vehicle_avg_speed*×*vehicle_load* performs better than *vehicle_avg_speed* alone in predicting traffic congestion. We should emphasize that we arrived at this conclusion while using different sampling techniques, thereby negating any particular bias inherent to any sampling technique used. Therefore, the present study suggests that both of the factors should be considered for a better traffic congestion prediction in smart cities.

Moreover, we should emphasize the fact that while 4 features are utilized - "*avgSpeed*" (*vehicle_avg_speed*) in km/h, "*vehicleCount*" (*vehicle_load*), "*medianMeasuredTime*" and "*avgMeasuredTime*", to train the classification model. Only 2 features, that is, *vehicle_avg_speed* and *vehicle_load* were used to produce labels. Consequently, the significantly good classification results in classifying *congested* and *not-congested*, emphasize the crucial role of the 2 features, which goes on to verify our rationale and intuition behind selecting only these 2 features.

5.1Limitations of the study

While the current study solves several challenges and problems as explained in the section 1, there are certain limitations of our study which should be presented before the reader. Firstly, due to the limitations of the dataset in question, our study does not consider the impact of external factors affecting traffic congestion prediction, like weather conditions and holidays. However, we advise the readers interested in future scope of our work, to consider such factors for better, more applicable and effective prediction. Secondly, this study does not implement deep learning techniques where we believe, our proposed methods can work even better. Thirdly, we focus on only one dataset, albeit of big data nature (13.5 million instances). Furthermore, this study works on historical data, i.e., no real-time data has been used. A complete list of abbreviations is shown in *Appendix I.*

6.Conclusion and future work

This work studied the issue of prediction of traffic congestion in transportation system of a smart city from a big data perspective. Although a number of studies have been conducted to deal with this problem of traffic congestion, the literature suggests that most of the existing studies suffer from improper label generation problems. Besides, generally, researchers have used relatively smaller datasets, however, the problem of traffic data is that of big data. To this end, in this study, we proposed intuitive and rational approaches considering multiple factors for effective traffic congestion prediction in big data scenario.

In particular, two important factors, namely, number of vehicles (*vehicle_load*) and average vehicle speed (*vehicle_avg_speed*) were focused upon using a well-known and publicly available CityPulse traffic dataset. We went on to generate labels by using 1) *vehicle_avg_speed*, 2) *vehicle_load* (the proposed method) and 3) *vehicle_avg_speed* ×*vehicle_load* (the proposed method), while keeping the data balanced using different sampling techniques. Afterwards, to examine the efficacy of the proposed approaches of label generation, we used five commonly used classification models, namely, SVM, LR , MNB, RF, and MLP. The obtained results show the effectiveness of different classification models under *vehicle_load* and *vehicle_avg_speed*×*vehicle_load*. Both the proposed approaches are effective in distinguishing *congested* from *not-congested* scenarios when compared to *vehicle_avg_speed*, even when the traffic data is massive (big data).

In future, we aim to build a traffic data optimizer framework where the safe reduction of the size of big data will be performed without compromising the performance. Additionally, we can also consider multiple datasets from various cities and include some external factors such as weather for effective traffic congestion prediction. Also, it will be interesting to see how deep learning can enhance the effectiveness of the proposed approaches of the current study. Furthermore, readers can also consider the problems of traffic flow and occupancy prediction.

Acknowledgment
None.

Conflicts of interest
The authors have no conflicts of interest to declare.

Authors contribution statement
Aamish Izhar: Conceptualization, methodology, data curation, formal analysis, investigation, validation, writing- original draft preparation. **Ajay Rastogi:** Conceptualization, methodology, data curation, formal analysis, investigation, visualization, writing- original draft preparation, writing- review and editing. **Syed Shafat Ali:** Conceptualization, methodology, data curation, formal analysis, investigation, writing- original draft preparation, writing- review and editing. **S. M. K. Quadri:** Validation, **s**upervision and project administration. **S. A. M. Rizvi:** Validation, **s**upervision and project administration.

References
[1] Negara JG, Emanuel AW. A conceptual smart city framework for future industrial city in Indonesia. International Journal of Advanced Computer Science and Applications. 2019; 10(7):453-7.
[2] Nour MK, Naseer A, Alkazemi B, Jamil MA. Road traffic accidents injury data analytics. International Journal of Advanced Computer Science and Applications. 2020; 11(12):762-70.
[3] Dabiri S, Heaslip K. Transport-domain applications of widely used data sources in the smart transportation: a survey. arXiv preprint arXiv:1803.10902. 2018.
[4] Christantonis K, Tjortjis C, Manos A, Filippidou DE, Mougiakou E, Christelis E. Using classification for traffic prediction in smart cities. In IFIP international conference on artificial intelligence applications and innovations 2020 (pp. 52-61). Springer, Cham.
[5] Mystakidis A, Tjortjis C. Big data mining for smart cities: predicting traffic congestion using classification. In international conference on information, intelligence, systems and applications 2020 (pp. 1-8). IEEE.
[6] Majumdar S, Subhani MM, Roullier B, Anjum A, Zhu R. Congestion prediction for smart sustainable cities using IoT and machine learning approaches. Sustainable Cities and Society. 2021.
[7] Zafar N, Ul HI. Traffic congestion prediction based on estimated time of arrival. PloS One. 2020; 15(12):1-19.
[8] Zheng J, Huang M. Traffic flow forecast through time series analysis based on deep learning. IEEE Access. 2020; 8:82562-70.
[9] Yu J, Yan Y, Chen X, Luo T. Short-term road traffic flow prediction based on multi-dimensional data. In international conference on intelligent transportation, big data & smart city 2021 (pp. 43-6). IEEE.
[10] Wang Z, Thulasiraman P. Foreseeing congestion using LSTM on urban traffic flow clusters. In 6th international conference on systems and informatics (ICSAI) 2019 (pp. 768-74). IEEE.
[11] Li Y, Huang C, Jiang J. Research of bus arrival prediction model based on GPS and SVM. In Chinese control and decision conference 2018 (pp. 575-9). IEEE.
[12] Liu Y, Wu H. Prediction of road traffic congestion based on random forest. In 10th international symposium on computational intelligence and design 2017 (pp. 361-4). IEEE.
[13] Bartlett Z, Han L, Nguyen TT, Johnson P. Prediction of road traffic flow based on deep recurrent neural networks. In smartworld, ubiquitous intelligence & computing, advanced & trusted computing, scalable computing & communications. 2019 (pp. 102-9). IEEE.
[14] Wang Y, Li L, Xu X. A piecewise hybrid of ARIMA and SVMs for short-term traffic flow prediction. In international conference on neural information processing 2017 (pp. 493-502). Springer, Cham.
[15] Kumar SV, Vanajakshi L. Short-term traffic flow prediction using seasonal ARIMA model with limited input data. European Transport Research Review. 2015; 7(3):1-9.
[16] Li KL, Zhai CJ, Xu JM. Short-term traffic flow prediction using a methodology based on ARIMA and RBF-ANN. In Chinese automation congress 2017 (pp. 2804-7). IEEE.
[17] Singh M, Srivastava VM. Prediction and avoidance of real-time traffic congestion system for Indian metropolitan cities. International Journal of Vehicle Information and Communication Systems. 2020; 5(1):109-18.
[18] Ali SS, Anwar T, Rastogi A, Rizvi SA. EPA: exoneration and prominence based age for infection source identification. In proceedings of the international conference on information and knowledge management 2019 (pp. 891-900).
[19] Lv Y, Duan Y, Kang W, Li Z, Wang FY. Traffic flow prediction with big data: a deep learning approach. IEEE Transactions on Intelligent Transportation Systems. 2014; 16(2):865-73.
[20] Devi S, Neetha T. Machine learning based traffic congestion prediction in a IoT based smart city. International Research Journal of Engineering and Technology. 2017; 4(5):3442-5.
[21] Ren C, Chai C, Yin C, Ji H, Cheng X, Gao G, et al. Short-term traffic flow prediction: a method of combined deep learnings. Journal of Advanced Transportation. 2021.
[22] Saddad E, Mokhtar HM, El-bastawissy A, Hazman M. Lake data warehouse architecture for big data solutions. International Journal of Advanced Computer Science and Applications. 2020; 11(8):417-24.
[23] Petalas YG, Ammari A, Georgakis P, Nwagboso C. A big data architecture for traffic forecasting using multi-source information. In international workshop of algorithmic aspects of cloud computing 2016 (pp. 65-83). Springer, Cham.
[24] Trovati M. Big-data analytics and cloud computing. Theory, Algorithms and Applications. 2015.

[25] Yin C, Lin Y, Yang C. A classification and predication framework for taxi-hailing based on big data. In international conference on intelligent computing 2017 (pp. 747-58). Springer, Cham.
[26] Florido E, Castaño O, Troncoso A, Martínez-alvarez F. Data mining for predicting traffic congestion and its application to Spanish data. In international conference on soft computing models in industrial and environmental applications 2015 (pp. 341-51). Springer, Cham.
[27] Meng M, Shao CF, Wong YD, Wang BB, Li HX. A two-stage short-term traffic flow prediction method based on AVL and AKNN techniques. Journal of Central South University. 2015; 22(2):779-86.
[28] Xie J, Choi YK. Hybrid traffic prediction scheme for intelligent transportation systems based on historical and real-time data. International Journal of Distributed Sensor Networks. 2017; 13(11):1-11.
[29] Kundu S, Desarkar MS, Srijith PK. Traffic forecasting with deep learning. In region 10 symposium 2020 (pp. 1074-7). IEEE.
[30] Joseph LL, Goel P, Jain A, Rajyalakshmi K, Gulati K, Singh P. A novel hybrid deep learning algorithm for smart city traffic congestion predictions. In international conference on signal processing, computing and control 2021 (pp. 561-5). IEEE.
[31] Zahid M, Chen Y, Jamal A, Memon MQ. Short term traffic state prediction via hyperparameter optimization based classifiers. Sensors. 2020; 20(3):1-22.
[32] Pramanik M, Rahman MM, Anam AS, Ali AA, Amin MA, Rahman AK. Modeling traffic congestion in developing countries using google maps data. In future of information and communication conference 2021 (pp. 513-31). Springer, Cham.
[33] Karau H, Konwinski A, Wendell P, Zaharia M. Learning spark: lightning-fast big data analysis. O'Reilly Media, Inc.; 2015.
[34] Ali MI, Gao F, Mileo A. Citybench: a configurable benchmark to evaluate rsp engines using smart city datasets. In international semantic web conference 2015 (pp. 374-89). Springer, Cham.
[35] Barnaghi P, Tönjes R, Höller J, Hauswirth M, Sheth A, Anantharam P. Citypulse: real-time iot stream processing and large-scale data analytics for smart city applications. In European semantic web conference (ESWC) 2014.
[36] Kolozali S, Bermudez-edo M, Puschmann D, Ganz F, Barnaghi P. A knowledge-based approach for real-time iot data stream annotation and processing. In international conference on internet of things (iThings), and IEEE green computing and communications (GreenCom) and IEEE cyber, physical and social computing (CPSCom) 2014 (pp. 215-22). IEEE.
[37] Rastogi A, Mehrotra M, Ali SS. Effective opinion spam detection: a study on review metadata versus content. Journal of Data and Information Science. 2020; 5(2):76-110.
[38] Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP. SMOTE: synthetic minority over-sampling technique. Journal of Artificial Intelligence Research. 2002; 16:321-57.
[39] Deshpande M, Bajaj PR. Performance analysis of support vector machine for traffic flow prediction. In international conference on global trends in signal processing, information computing and communication 2016 (pp. 126-9). IEEE.
[40] Schütze H, Manning CD, Raghavan P. Introduction to information retrieval. Cambridge: Cambridge University Press; 2008.
[41] Hosmer JDW, Lemeshow S, Sturdivant RX. Applied logistic regression. John Wiley & Sons; 2013.
[42] Breiman L. Random forests. Machine Learning. 2001; 45(1):5-32.
[43] Goodfellow I, Bengio Y, Courville A. Deep learning. MIT Press; 2016.

Aamish Izhar completed his bachelor's (BSc (H) Mathematics) and master's (MCA) degrees from Jamia Millia Islamia (A Central University), New Delhi. He has qualified National Eligibility Test (NET), and at present, pursuing PhD from the Department of Computer Science, Jamia Millia Islamia. His research interests include Smart Cities, Big Data, Machine Learning, IoT and Cloud Computing.
Email: aizhar22@gmail.com

Dr. Ajay Rastogi is presently working as a guest faculty in Department of Economics, Jamia Millia Islamia (A Central University), New Delhi. He received his Bachelor's Degree in Science and Mathematics (B. Sc.) from Mahatma Jyotiba Phule Rohilkhand University in 2010 and Master's Degree in Maths with Computer Applications (M. Sc. Tech.) from Jamia Millia Islamia in 2013 (Gold Medalist). Dr. Rastogi completed his PhD in Computer Science from Jamia Millia Islamia in 2020. He qualified University Grant Commission's National Eligibility Test (UGC-NET) and was awarded Junior Research Fellowship (JRF) in June 2013 and December 2013, respectively. Dr. Rastogi has reviewed papers from top publications like IEEE Transactions on Knowledge and Data Engineering (TKDE). His current research interests include Data Analytics, Machine Learning, Text Mining, Graph Mining and Social Computing.
Email: ajay148115@st.jmi.ac.in

Dr. Syed Shafat Ali is from Kashmir, India. He completed his bachelor's (BCA) and master's (MCA) degrees from University of Kashmir in 2011 and 2014, respectively, and completed his PhD in Computer Science from Jamia Millia Islamia in 2021. He qualified State Eligibility Test (SET) in 2018 and National Eligibility Test (NET) in 2019. He was awarded student travel grant by ACM-SIGIR and conference fee grant by IARCS/ACM-India to attend CIKM'19 conference in Beijing, China. Dr. Ali has been a reviewer for top science journals like PLOS One, Physica A: Statistical Mechanics and its Applications and IEEE Transactions on Knowledge and Data Engineering (TKDE). Dr. Ali is presently working as guest lecturer at the Department of Economics in Jamia Millia Islamia where he has taught courses in mathematics, statistics, data analytics and programming languages like Python, Prolog and C. His research interests Include Complex Networks, Graph Theory, Online Social Networks Analysis, Data Mining and Machine Learning.
Email: shafat159074@st.jmi.ac.in

Dr. S. M. K. Quadri received his M. Tech. in Computer Applications from Indian School of Mines and PhD in Computer Sciences from Kashmir University. Presently, he is a Professor in the Department of Computer Science at Jamia Millia Islamia (A Central University), New Delhi, and serving as Head of the Department. He is a Fellow of IETE and an Expert Member of National Accreditation and Assessment Council (NAAC), besides being a member of other well-recognized societies. His research interests include Software Testing, Software Reliability, Disk File systems and Artificial Intelligence. He has many research papers published in various reputed conferences and journals. He has so far supervised 18 PhDs in his academic career, and at present, 08 research scholars are researching under his guidance.

Email: quadrismk@jmi.ac.in

Dr. S. A. M. Rizvi is a Professor in the Department of Computer Science at Jamia Millia Islamia (A Central University), New Delhi, where he, recently, completed his second term in office as Head of the Department. He has more than 170 research publications and has authored 6 text books as well. So far in his academic career, he has supervised 25 PhDs in different areas of computer science. Besides various academic positions held by him and being the Founder-Director of many institutions, he has also served as Chief Manager (IT) in Goa Shipyard under the Ministry of Defense, Government of India. In addition to this, he has served as a Consultant-Programme Director (MIS) at Abu Dhabi University in UAE and Jawaharlal Institute of Technology (JIT) in India. Prof. Rizvi has a rich experience of academic management, evaluation/accreditation process and administration through the roles such as founder-director/dean/head of institutions/departments in various institutions.
Email: sarizvi@jmi.ac.in

Appendix I

S.No.	Abbreviation	Description
1	ANN	Artificial Neural Network
2	APIs	Application Programming Interfaces
3	ARIMA	Autoregressive Integrated Moving Average
4	AVL	Adelson-Velsky and Landis
5	BLSTME	Boosted Long Short-Term Memory Ensemble
6	BPNN	Back Propagation Neural Network
7	CDLP	Combined Deep Learning Prediction
8	CNN	Convolutional Neural Networks
9	CV	Cross-Validation
10	DJ	Decision Jungles
11	DT	Decision Tree
12	ETA	Estimated Time of Arrival
13	GIS	Geo Information System
14	GPS	Global Positioning System
15	GRU	Gated Recurrent Unit
16	HA	Historical Averages
17	ICT	Information and Communications Technology
18	KF	Kalman Filter
19	KNN	K-nearest Neighbour
20	LD-SVM	Local Deep Support Vector Machine
21	LR	Logistic Regression
22	LSTM	Long Short Term Memory
23	MAE	Mean Absolute Error
24	MDLSTM	Multi-Dimensional LSTM
25	MLP	Multilayer Perceptron
26	MNB	Multinomial Naïve Bayes
27	MSE	Mean Square Error
28	NN	Neural Network
29	PMA	Periodic Moving Average
30	RBF-ANN	Radial Basis Function-Artificial Neural Network
31	RMSE	Root Mean Square Error
32	RNN	Recurrent Neural Networks
33	RF	Random Forest
34	SMOTE	Synthetic Minority Oversampling Technique
35	SQL	Structured Query Language
36	STATS	Standardised Accuracy And Time Score
37	SVM	Support Vector Machine
38	SVR	Support Vector Regression
39	VAR	Vector Auto Regression
40	XGBoost	Extreme Gradient Boosting