DataFITS: A Heterogeneous Data Fusion Framework for Traffic and Incident Prediction

Philipp Zißner, Paulo H. L. Rettore, Bruno P. Santos, Johannes F. Loevenich, and Roberto Rigolin F. Lopes, *Member, IEEE*

***Abstract*— This paper introduces DataFITS (Data Fusion on Intelligent Transportation System), an open-source framework that collects and fuses traffic-related data from various sources, creating a comprehensive dataset. We hypothesize that a heterogeneous data fusion framework can enhance information coverage and quality for traffic models, increasing the efficiency and reliability of Intelligent Transportation System (ITS) applications. Our hypothesis was verified through two applications that utilized traffic estimation and incident classification models. DataFITS collected four data types from seven sources over nine months and fused them in a spatiotemporal domain. Traffic estimation models used descriptive statistics and polynomial regression, while incident classification employed the k-nearest neighbors (k-NN) algorithm with Dynamic Time Warping (DTW) and Wasserstein metric as distance measures. Results indicate that DataFITS significantly increased road coverage by 137% and improved information quality for up to 40% of all roads through data fusion. Traffic estimation achieved an R^2 score of 0.91 using a polynomial regression model, while incident classification achieved 90% accuracy on binary tasks (incident or non-incident) and around 80% on classifying three different types of incidents (accident, congestion, and non-incident).**

***Index Terms*— Intelligent transportation systems, heterogeneous data fusion, traffic estimation, incident classification.**

I. Introduction

DATA availability is a critical aspect in the design of modern Intelligent Transportation Systems (ITSs), which implement models to understand better various patterns of the transportation system [1], thus improving mobility and safety for people and goods. With modern society depending heavily on efficient and reliable transportation, the importance of these systems has seen a rapid increase in significance over recent years. In Germany alone, both the number of registered cars and the number of carried passengers using public transportation have shown a substantial increase, reaching their all-time highs of 48.5 million cars (2022) and 12.7 billion carried passengers (2019, before the pandemic) [2], [3]. As a result, urban areas experience an increasing number of traffic-related incidents (e.g., congestion and accidents), increasing time delays, emissions, and fuel consumption [4].

For this reason, academia and industry have driven efforts to create the next generation of transportation systems that are eco-friendly, cost-efficient, and powered by data analysis and communication technology. We hypothesize that a heterogeneous data fusion framework can enhance the coverage and quality of information serving as input for traffic models, thus increasing the efficiency and reliability of ITS applications. Therefore, we propose the Data Fusion on Intelligent Transportation System (DataFITS) framework, providing a spatiotemporal fusion of data used to train models for two ITS applications, traffic estimation, and incident classification. DataFITS collects and combines real heterogeneous data (e.g., weather, traffic, incident) from various sources (e.g., open databases, map applications), preparing them by fixing errors, adapting the data structure, and finally fusing them in the exact location and point in time. Our hypothesis is verified using data characterization to quantify the benefits of combining heterogeneous data sources and the proposal of two ITS applications. The performance of the two applications ratifies the benefits of larger data coverage/quality while estimating traffic and classifying incidents. Thus, the main contributions of this investigation are:

- An open-source framework DataFITS for heterogeneous spatiotemporal data fusion, covering the acquisition, processing, and fusion of data, available in a public code repository.[1]
- The characterization of a heterogeneous dataset combining real traffic data from two cities in Germany, collected from seven sources over nine months and provided together with the repository.
- Two traffic estimation models, one using descriptive statistics and another using polynomial regression with different parameters such as time, road type, and weather, and a comparison between single and fused datasets.
- An incident classification model trained and evaluated on heterogeneous fused data using k-nearest neighbors (k-NN), with Dynamic Time Warping (DTW) and Wasserstein as distance methods.

The rest of the paper is organized as follows. Section II reviews recent literature using data fusion to design applications like traffic estimation and incident classification and

Manuscript received 7 February 2023; revised 20 April 2023; accepted 25 May 2023. This work was suppported by the Bundeswehr through Federal Office of Bundeswehr Equipment, Information Technology, and In-Service Support (BAAINBw) and Bundeswehr Technical Center for Information Technology and Electronics (WTD81). The Associate Editor for this article was T. Tettamanti. *(Corresponding author: Paulo H. L. Rettore.)*

Philipp Zißner and Paulo H. L. Rettore are with the Communications Systems Department, Fraunhofer FKIE, 53177 Bonn, Germany (e-mail: philipp.zissner@fkie.fraunhofer.de; paulo.rettore.lopes@fkie.fraunhofer.de).

Bruno P. Santos is with the Department of Computer Science, Federal University of Bahia, Salvador 40170-110, Brazil (e-mail: bruno.ps@ufba.br).

Johannes F. Loevenich is with the Communications Systems Department, Fraunhofer FKIE, 53177 Bonn, Germany, and also with the Department of Mathematics/Computer Science, University of Osnabrück, 49074 Osnabrück, Germany (e-mail: johannes.loevenich@fkie.fraunhofer.de).

Roberto Rigolin F. Lopes is with the Secure Communications and Information (SIX), Thales Deutschland, 71254 Ditzingen, Germany (e-mail: roberto.rigolin@thalesgroup.com).

Digital Object Identifier 10.1109/TITS.2023.3281752

[1] https://github.com/prettore/DataFITS

compares them against our solution. The design of DataFITS and the traffic data applications are described in Section III. Section IV evaluates the performance of our framework and the effectiveness of our traffic estimation and incident classification models using the heterogeneous fused data, verifying our hypothesis. Finally, we conclude this paper in Section V, highlighting open problems for future investigations.

II. Related Work

This section reviews the literature on three main topics related to our proposed solution: (i) data collection and fusion, (ii) traffic estimation, and (iii) incident classification. Finally, we summarize and compare the literature with our proposal.

A. Data Collection and Fusion

To develop ITS applications, significant data is required from real or virtual sensors [5]. Vitor et al. [4] present a platform to collect, process, and export heterogeneous data from smart city sensors, providing different statistics and visualizations. However, their platform concentrates on securing data. Similarly, [6] proposes a smart city data platform containing information from various cities. In contrast to our framework, we focus on improving the quantity and quality of the information by fusing data, and we assess the advantages of using fused data through two ITS applications.

Data fusion combines data from multiple sources, enriching spatiotemporal information [7], [8], [9], [10]. Several applications benefit from data fusion, such as emergency management [11] and path planning [12]. However, fusing heterogeneous data requires additional preprocessing to combine various data types and features [13], [14]. This investigation focuses on two applications supported through data fusion: traffic estimation and incident classification, and the methods to achieve their goals, such as data acquisition, fusion, machine learning, correlation, and different data types.

B. Traffic Estimation

Traffic estimation is a crucial smart city application for better transportation management. This review focuses on data fusion, spatiotemporal correlation, and machine learning techniques to achieve accurate and reliable traffic estimation using historical data. The increasing availability of open databases (kept by governmental authorities) and Application Programming Interfaces (APIs) to commercial applications (Bing, Google Maps, etc.) results in a vast collection of traffic-related data, making big data an opportunity for heterogeneous data fusion [15]. The challenge is to combine stationary sensor data (e.g., traffic cameras or loop detectors) and probe vehicle information (e.g., cameras, GPS, cellular data, or vehicular sensors). Anand et al. [16] used a Kalman filter to fuse traffic flow values (from cameras) and travel time (from GPS), improving a traffic estimation approach.

Many recent traffic estimation models use Machine Learning (ML) [17], [18], [19], [20], [21], [22], [23], [24], [25]. Reference [17] proposes an auto-regressive model that uses data from a traffic simulator and adapts to events like accidents. Their results showed that estimation up to 30 minutes ahead has an error of 12%. Meanwhile, [18] employs deep learning algorithms for traffic estimation, showing an improvement of accuracy and efficiency. These approaches discuss the usage of ML to create accurate models for traffic estimation, but do not consider further methods, such as data fusion, correlation, etc.

Some ML approaches use spatiotemporal correlation to improve traffic estimation quality. In [19], a neural network(NN)-based estimation using Graph Convolutional Network (GCN) and Gated Recurrent Unit (GRU) models is proposed with full public access. The GCN captures spatial dependencies from the road network, and GRU detect dynamic changes in traffic data and captures temporal dependencies. Other NN-based approaches, such as [20] and [21], show similar improvements in accuracy using data correlation. Wang et al. [22] propose an open-source deep learning framework using GCN to estimate network-wide traffic multiple steps ahead in time. Zheng et al. [23] introduce another open-source solution, the Graph Multi Attention Network (GMAN), using an encoder-decoder architecture to provide long-term traffic estimation up to one hour ahead. These approaches also include correlation to improve the discussed models and offer access to their data but do not propose a solution for collecting or fusing data. Limited literature combines data fusion, spatiotemporal correlation, and ML to estimate traffic, similar to our solution. In [26], the authors fuse traffic data from stationary and dynamic sensors, considering the spatiotemporal correlation between traffic levels of road segments. A Multiple Linear Regression (MLR) model processes the fused information to enhance traffic estimation accuracy. Unlike our solution, this approach relies solely on traffic data from sensors but does not consider different data types and sources. Zhao et al. [24] propose a general platform for spatiotemporal data fusion to enhance traffic estimation. The approach introduces a fusion method to improve accuracy by combining direct and indirect traffic-related data as input for two different ML models. The indirect traffic-related data features contain information about weather and points of interest and are used to improve the estimation quality. However, their model uses pre-existing datasets, offering no solution for data collection, and our study focuses on incident-related data, while the authors in [24] consider points of interest and weather conditions.

In [27], the authors introduce a model to estimate traffic within a small urban area in Zurich, with data acquired as part of a video measurement campaign. Their solution fuses information from Loop detectors, traffic lights, and other sensors (e.g., video plus license plate recognition, thermal cameras) and trains different MLR models with this data. Finally, they evaluate the various sensors' accuracy and robustness. In contrast to our solution, they investigate the quality of a regression model using different sensor data fused to stationary data. Furthermore, their data is acquired using sensors that are not publicly available, covering only a small urban area.

Finally, [25] proposes a traffic speed prediction by integrating heterogeneous data from various sensors, including exogenous data like weather, into a hybrid spatiotemporal features space. The main contributions are a hybrid model

TABLE I
RELATED WORK COMPARISON

App	Key Aspects	Lit	Data Acq.	Data Fus.	ML	Corr.	Data Types		
							Station.	Probe	LBSM
Smart City	Platforms to provide het. data	[4] [6]	✓				✓	✓	
Emerg.	Emerg. managem.	[11]		✓	✓		✓		✓
	Path planning	[12]			✓			✓	
Traffic Estimation	Spatiotemporal fusion of het. data	[16]		✓		✓	✓		
	Correlation of data features	[36] [37] [38]				✓		✓	
	ML to improve accuracy	[17] [18]			✓		✓		
	Train ML models using data with corr. features	[20] [21] [22] [23]			✓	✓	✓		
		[19]			✓	✓	✓	✓	
	Combination of data fusion, ML and data with corr. features	[26] [24]		✓	✓	✓	✓	✓	
		[27]	✓	✓	✓		✓	✓	
		[25]	✓	✓	✓	✓	✓	✓	
Inc. Clas.	Classify incidents by traffic patterns	[28] [29] [30] [31]		✓	✓	✓	✓		
	Social media data	[35] [39] [34]	✓	✓	✓		✓		✓
Prop. Sol.	Data acq. data fusion and two data applications	This paper	✓	✓	✓	✓	✓	✓	⧖

✓covered; ⧖ supported, not yet implemented; public access: no, limited, yes

using Long short-term memory (LSTM) and GRU, comparing the model against other well-known classical deep learning models, showing the highest efficiency and lowest error metric. In contrast to our study, this investigation focuses on the prediction using only vehicle speed and has no open access to their solution and the data.

C. Incident Classification

Numerous ML and deep-learning models are also used for incident classification [28], [29], [30], [31]. These models improve road safety in urban areas by facilitating traffic management, warning systems, and emergency rescue operations. Other applications, such as incident detection, are proposed in [32] and [33], which provide additional traffic management enhancements, including the ability to control traffic lights from emergency vehicles.

In [28], the authors introduce a Convolutional Neural Network (CNN) model to predict traffic accidents using a state matrix with influencing traffic features. Their solution achieves high prediction accuracy, but limited training data affects CNN model quality, which could be improved by using data fusion. Park et al. [29] propose a big data approach using the *Hadoop framework* to combine incident-related and other traffic data. The study classifies data into groups of traffic incidents. Data fusion benefits the approach, but incorporating spatiotemporal aspects could further increase model accuracy.

In [30], the authors propose a recurrent neural network to predict traffic accident risk by combining incident data with a spatiotemporal traffic correlation. The model has high accuracy and can be used for accident prevention and integrated into traffic control systems. However, its main limitation is the consideration of only directly-related incident data. Other traffic-related features (e.g., traffic flow, weather, vehicular data, etc.) could be fused to improve accuracy. Shang et al. [31] propose a hybrid approach for automatic incident detection using random forest-recursive feature elimination and a LSTM network with Bayesian optimization. Their approach provides an accurate binary classification of incidents, outperforming other state-of-the-art solutions, but does not classify them into different types.

Other approaches use location-based social media (LBSM) to improve the detection and classification of incidents. Rettore et al. [34] propose a framework containing two data services, one to detect traffic-related events. The framework collects data from social media platforms (e.g., Twitter), which is used in a road incident detection model based on heterogeneous data fusion to provide more descriptive transportation system data. The free access of user data through Twitter's API improves the availability of incident data, an essential aspect in developing ITS solutions. Also, in [35], the authors describe a real-time traffic event detection solution using Twitter posts. Their solution is based on a text classification algorithm to identify traffic-related tweets with their location and classify the information into different classes of events.

D. Comparison & Summary

Table I summarizes the reviewed literature, categorizing them into five applications: smart city, emergency, traffic estimation, incident classification, and our solution. The second and third columns list the key aspects and the corresponding references. The remaining columns denote the following labels: *Data Acquisition*, *Data Fusion*, *ML*, *Correlation*, *Stationary*, *Probe*, and *LBSM*. These labels indicate whether the approach collects data, uses data fusion techniques, utilizes ML and deep-learning models, incorporates data correlation, employs stationary sensor data, uses probe vehicle data, or utilizes georeferenced social media data. Moreover, we classify the availability of the source code and data of all solutions into three categories using different colors *no*, *limited*, or *yes* public access. A paper labeled with *no public access* does not offer access to their data or solution, unlike solutions that provide *full public access* to source code and data. *Limited public access* describes the usage of datasets that are not accessible anymore or solutions that plan to offer open access in theory but currently do not fulfill this aspect.

The last row of Table I compares our investigation with the literature, highlighting the coverage and contributions of our proposed solution. Compared with most of the literature, we provide a methodology that covers four of five stages of the data cycle (acquisition, preparation, processing, use) [13], providing an open-source framework,[1] and access to the collected datasets. Making the models and datasets available, or the means to acquire and process them, is crucial to enable a fair comparison between models/methodologies, which we did not find in most literature. Moreover, the DataFITS framework is designed to support multiple data types, including stationary and probe data, and can potentially incorporate additional types of information like LBSM. We perform spatiotemporal data fusion to provide enriched information used as an input for two, but not limited to, data applications showing the benefit of using fused heterogeneous data. In contrast, other approaches in the literature focus on specialized solutions that combine only a subset of the listed features in the context of ITS.

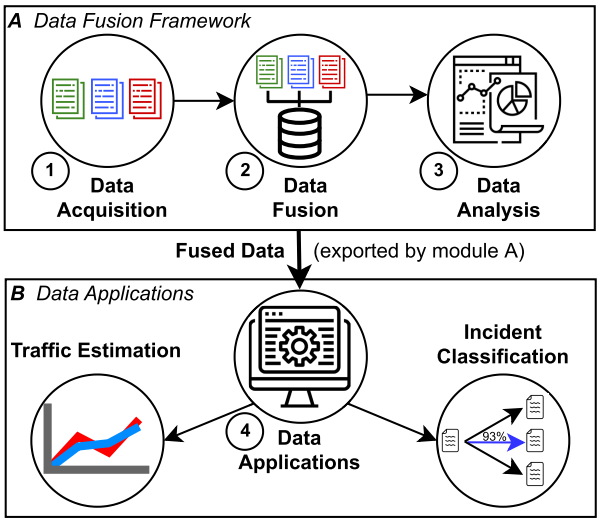


Fig. 1. The general workflow.

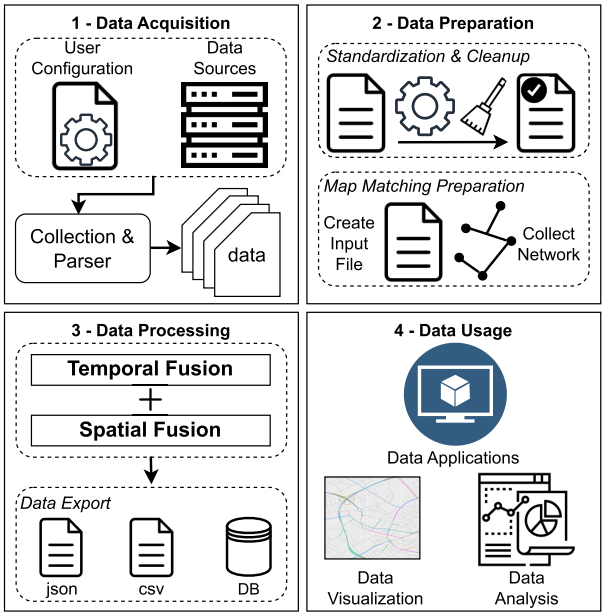


Fig. 2. Workflow of DataFITS.

III. The Design

This research proposes a solution including two different modules: A data fusion framework DataFITS and two data applications traffic estimation and incident classification. The DataFITS design follows a three-stage workflow, as presented in Fig. 1-A. It starts by gathering data from heterogeneous transportation-related data sources using APIs and web crawlers (1). In sequence, all acquired data are fused geographically by mapping them to road segments and aligned temporally (2). After fusing the data, we can perform data analysis to identify and visualize specific data characteristics (e.g., traffic and incident statistics) (3). DataFITS can export data which then can be used as input for different applications, depicted in Fig. 1-B. In this article, we use the fused data in two applications: traffic estimation and incident classification that can benefit from fused data (see Section III-B) providing a more comprehensive perspective of the results (4).

A. Data Fusion Framework

1) Data Acquisition: Within the data acquisition, Fig. 2 (1), DataFITS collects information from different predefined data sources according to a set of user-defined parameters (e.g., geographical area and time interval). Currently, DataFITS supports multiple methods to collect traffic, incident, vehicular, and weather data. In addition, the framework parses heterogeneous information and stores them in standardized CSV files. The acquisition follows a modular application design, ensuring easy expandability of the framework functionalities and allowing the specification of additional data sources.

2) Data Preparation: The compiled data undergo an additional preparation step as illustrated in Fig. 2 (2). The key component of the preparation stage is data standardization, converting different feature names and types into a uniform representation and a set of user-customizable data mappings to deliver consistent data types. In sequence, the data is prepared to be mapped onto geographical locations. Leveraging OpenStreetMap (OSM), a free map database, DataFITS gathers shapefiles according to the bounding box parameter specified in the data acquisition stage, using OSMNX [40]. A shapefile stores road network information identified by the primary key (*fid*) for each road segment, which is used in the map-matching procedure conducted within the data processing. DataFITS can also extract the road type and speed limit from shapefiles. Finally, the collected data is converted into "trip files", a representation of the input used by the map-matching.

3) Data Processing:

a) Temporal fusion: Fig. 2 (3) displays the temporal data fusion. This process groups the complete data within an arbitrary time window aggregation (e.g., hourly, daily, or 10 minutes for the results in this paper), adapting the time interval from the collection process.

b) Spatial fusion: DataFITS leverages the map-matching technique, taking GPS points and aligning them to established coordinates under a predetermined degree of accuracy based on an underlying road network. This results in a balanced level of accuracy and associate all geo-located data with the same road network.

Among different strategies of map-matching, DataFITS integrates Fast Map Matching (FMM), an open-source tool, which provides two different algorithms for achieving optimal performance based on the given road network size [41]. To this end, FMM uses the trip and shapefiles created in the prior stage and connects all input data points to a corresponding road network. Each data entry within the trip file contains a *Linestring* representing the GPS coordinates (path) of a road segment, except for incident data entries, which only contain coordinates of a start and end point. In addition to the matched points of each input entry, the algorithm returns two arrays, *opath* and *cpath*, that contain a set of road identifiers (*fids*) from the OSM. The first array, *opath*, stores the *fid* for each matched point, representing a list of road segments that got matched to the input data entry (data source coordinates with OSM road map). The *cpath*, second array, stores the *fid* values that create a path between all matched road segments. This is

necessary for creating the road trajectories for data entries that are only represented by a start and an end point within the data.

This process is performed on each record of the vehicular and incident data sources, while the geo-location of the traffic data sources is only matched once for each area, as those are static and do not change between data acquisitions. This strategy significantly reduces the execution time and computation required for map-matching. Instead of processing all data points within each acquisition, the main amount of data points, namely the traffic-related information, is only matched once. On our nine-month data time frame and a 10-minute acquisition time, this reflects a single matching procedure instead of 38,800 procedures, significantly reducing the runtime and required computation power.

The spatial data fusion process combines the fused input dataset with the map-matching output, adding the enriched information about *opath*, *cpath*, and *matched GPS points*. To provide the data for all given road segments, the information is rearranged by extracting all *fid* values from the *cpath* array of each data entry. The amount of computing power and memory for grouping data points is a non-linear function of the input data size. Therefore, the framework splits the data into chunks, reducing the memory requirement and allowing multi-threading to speed up the process.

4) Data Usage: The last stage, (4) in Fig. 2, describes different use cases of the fused dataset, e.g., as an input to various data applications or being characterized through different types of statistics and visualizations for spatiotemporal data analysis. For example, DataFITS provides heat maps and density plots separated by each source and different features, such as the number of observations, traffic levels, speed, and types of incidents. In the scope of temporal analysis, DataFITS provides time-series statistics for a specific time window (e.g., by the hour, day of the week, month, and season) and shows the correlation between different features. Moreover, the fused data is exported in different data structures, allowing to be used by various data applications, such as our proposed models or other third-party tools (e.g., *ArcGIS*).

B. Applications

1) Traffic Estimation: The proposed traffic estimation application is organized into two phases, as shown in Fig. 3. Phase (1) prepares the data, groups it by intersecting areas, identifies similar traffic regions based on correlating traffic patterns and performs a train-test-split. A traffic region is defined as the set of connected paths (road segments) reported from a data source, represented through unique road identifiers (*fids*). By intersecting areas, we obtain a list of unique traffic regions and are able to measure the similarities between them. In phase (2), the prepared data is used to create and evaluate two traffic estimation models using: i) descriptive statistics (naive); and ii) polynomial regression. Each model estimates traffic values for a single area within an arbitrarily defined time interval and can also utilize data from correlating regions with similar traffic behavior. Furthermore, the process considers optional input parameters like weekday, weather, and road type to create more specific models for the given characteristics. This research mainly focuses on the regression-based model

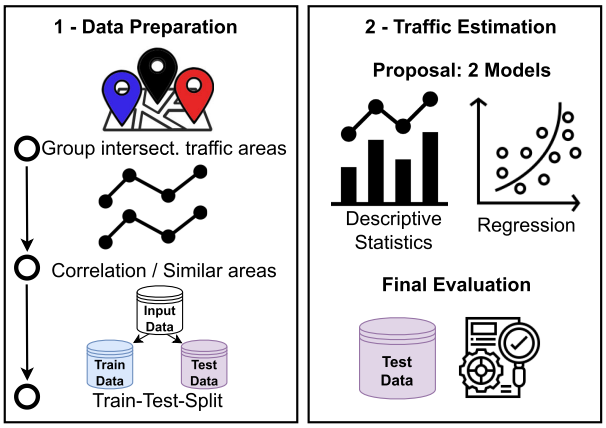


Fig. 3. Design of the traffic estimation application.

but also discusses the model based on descriptive statistics. and gives a comparative evaluation of both approaches in section IV.

a) Preprocessing: The fused data from DataFITS is cleaned, removing all incident-related information, as it is not required by the model, and grouped into traffic areas containing one or multiple road segments. Using a data aggregation over the array of road identifiers (*cpath*), we create a list of areas contributing traffic information to the dataset.

Due to the traffic area grouping, the data may contain overlapping areas due to the data fusion that merges traffic areas from different sources. Those intersecting areas describe the same spatial region but with minor differences in the covered road segments. Combining them removes potential duplicated areas, resulting in a final set of unique traffic areas. The underlying function iterates through all existing areas, calculates pairwise intersections, and combines them if the overlapping road segments exceed a predefined threshold $th_{overlap}$. Finally, the initial set of fused data is re-grouped according to the new set of combined traffic areas, resulting in an input dataset for the traffic estimation models that contains the combined information for each area.

Furthermore, the design covers a procedure to add data points from other regions that show similar traffic patterns based on correlation. The goal is to increase the volume of data points in areas with insufficient training data. Therefore, by correlating the traffic patterns (traffic level/relative speed) from different regions, the highly correlated areas can be merged, increasing the training dataset, thus, benefiting the accuracy of the traffic estimation. To identify such regions, we calculate data similarity based on the traffic values, aiming for a more precise representation of the traffic situation within the original area. The corresponding function implements a modified version of the Pearson Correlation and the DTW to identify correlated traffic areas with similar traffic patterns. The correlation between two time series was defined in [36] and adapted for our proposed methodology.

$$X_{i,j} = \frac{\sum_{t=1}^{L}(S_i(t) - \bar{S}_i)(S_j(t) - \bar{S}_i))}{\sqrt{\sum_{t=1}^{L-t}(S_i(t) - \bar{S}_i)^2} \cdot \sqrt{\sum_{t=1}^{L-t}(S_j(t) - \bar{S}_j)^2}} \quad (1)$$

Using Eq. (1), we calculate the respective correlation between two time series of any traffic data feature $S_i(t)$ for

two regions i and j at a given time t. We compute this value between all regions and define a correlation threshold th_{cor} to identify similarity. However, this type of correlation can solely describe a linear relationship between two variables, not considering the value variation. To overcome this issue, we use DTW to measure the distance between the two series and set a threshold th_{dtw} to ensure that both correlating areas have similar values. DTW measures the similarity between two time series that are not synchronized. More precisely, the algorithm can use a temporal alignment of the data pattern resulting in a more similar comparison than using, e.g., the Euclidean distance, comparing timestamps regardless of the feature values [42]. Calculating both correlation and DTW for the traffic and speed data, we can identify a set of areas that show similar patterns and satisfy Eq. 2:

$$\begin{aligned}&(cor_{traf} \geq th_{cor} \wedge cor_{speed} \geq th_{cor}) \wedge \\ &(dtw_{traf} \leq th_{dtw} \wedge dtw_{speed} \leq th_{dtw})\end{aligned} \tag{2}$$

Finally, the dataset is filtered according to the chosen parameters (i.e., time frame, weekday, road type, and weather) to generate the final model input data. We perform a *Train-Test-Split* and generate estimations for each individual area.

b) The model: Our initial model to estimate traffic values is based on descriptive statistics, with the goal to verify if basic statistics with low-computational costs can provide accurate predictions on a set of heterogeneous fused data. Eq. 3 describes the calculation of $Y(t)$, representing an estimated traffic value for a point in time t. Additionally to the mean of the original region at time t, $x(t)$, we also add the average value of all correlating regions $i \in 1, \ldots, n_{corr}$, represented by the second part of the equation. More details on the descriptive statistics approach can be found in [43].

$$Y(t) = \overline{x(t)} + \frac{1}{n_{corr}} * \sum_{i=1}^{n_{corr}} \overline{x_i(t)} \tag{3}$$

The second proposed traffic estimation model is based on ML and uses polynomial regression to estimate a continuous traffic value Y (e.g., traffic level or speed), as shown in Eq. (4).

$$Y = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \ldots + \theta_d x^d + \epsilon \tag{4}$$

The input feature x represents all traffic or speed values that are within the training dataset for a single traffic region, matching the parameters defined by the model. A higher-order polynomial, up to a degree d, can represent the dependent variable Y. The corresponding implementation calculates a least squares regression, resulting in an estimation that minimizes the sum of squares between the dependent and independent variables. The model is configured based on the input data created in the preprocessing phase, matching the previously defined thresholds and the following parameters: i) Required: data feature, polynomial degree, and time frame; ii) Optional: weekday, weather, and road type.

For example, the model could be configured to estimate the traffic level for a traffic area on a Monday in a 10-minute time interval.

The regression is implemented, generating a polynomial feature matrix of a certain degree d, where the optimal degree depends on the input data used to train the model and is obtained during the model creation. Then, the data is transformed into a matrix of features to represent the given input in a higher-order feature space. For example, a 2-dimensional feature space (X_1, X_2) is transformed to $(1, X_1, X_2, X_1^2, X_1 \cdot X_2, X_2^2)$. Therefore, the newly created feature contains the bias value of 1, all values raised to the power for each degree $\in 0, \ldots, d$, and all combinations between every pair of features. Finally, the data points within the training dataset are fitted using polynomial regression, and the model is used to create traffic value estimations.

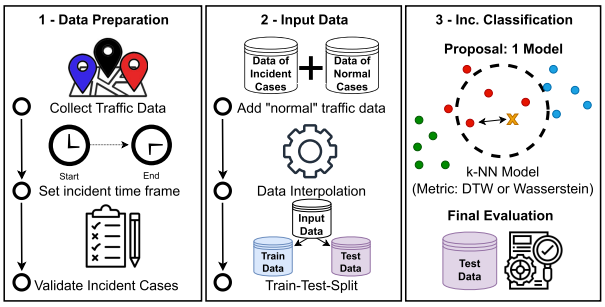


Fig. 4. Design of the incident classification application.

2) Incident Classification: Our second proposed application classifies traffic patterns to different incident types using a modified version of the k-NN algorithm. The application has three stages as shown in Fig. 4. It starts by applying preprocessing methods to prepare the input for the classification model, including collecting incident-related traffic data, defining a time frame for all incidents, and data validation mechanisms. Secondly, an input dataset is made by combining data from incidents and non-incident traffic situations. Also, a data interpolation process fills gaps in the traffic data and a train-test-split is performed. Finally, we train a k-NN-like algorithm to perform binary classification (incident and non-incident) or a multi-class classification (accident, congestion, and non-incident).

a) Preprocessing: First, all incident-related information is collected from the heterogeneous dataset, filtering the data by incident information and grouping it into unique reports with a specific duration. Because the incident-related data sources do not include information about traffic data features (e.g., traffic level and speed), this type of information is added through spatiotemporal fusion. To gather all data related to one specific incident, a method extracts all traffic areas that have a spatial intersection with the incident region and combines them in a temporal domain. The corresponding implementation uses the dataset of incidents, a list containing the unique traffic areas, and an overlapping threshold to calculate the intersection between the incident area and corresponding regions with available traffic information.

The incident duration is key information to define the right time window, which includes traffic data related to a particular incident. However, usually, this duration is not included in the data sources, requiring strategies to calculate it using another data source. Therefore, we developed two approaches: static and estimated incident start time. The static approach collects traffic data in a time interval of 90 or 120 minutes

before and after the reported incident start time from the data source, representing a time interval that includes data prior to and after the measurable incident effect. We chose the values of 90 and 120 minutes based on an exploratory data analysis (briefly indicated in Table IV), showing that the majority of incidents did not exceed 120 min impacting the traffic behavior, except for a few cases during congestion and disabled vehicle incidents. The second approach tries to estimate the incident start time, iterating over the traffic data, to find significant changes in the traffic pattern and setting the start point based on this observation. The estimated approach aims to provide a more realistic representation of the incident start time, which is evaluated later in Section IV.

Moreover, each incident report is validated by identifying samples with high noise in the corresponding traffic data. Noise has a negative contribution to the model and adds a potential bias to its accuracy. Therefore, it is detected and removed from the input dataset using three different strategies to validate each incident report, where at least one must be satisfied: i) Comparing the absolute difference between the traffic at the start and the end of the incident time interval, checking for a noticeable difference given by a deviation of more than a predefined threshold; or ii) Calculating the standard deviation over the entire incident time interval and comparing it to a certain threshold; or iii) Iterating over the data points close to the incident start and end time and checking for a point-wise traffic variation above a defined threshold. Based on these methods, we extract incidents that reflect a clear traffic pattern that shows a measurable impact of the incident on the traffic behavior and remove all other patterns that could be confused with a non-incident traffic pattern or a biased sensor report.

Lastly, to create the final dataset for the classification model, three further data processes are performed:

- *Add "normal traffic data"*: Adding non-incident samples to the input data is essential in the designing of our incident model. We add observations similar in time, weekday, and location to get the most comparable data. Therefore, these reports can accurately identify a non-incident situation for every incident in the dataset.
- *Data Interpolation*: As a result of measurement errors or other problems in the data collection, there is a possibility of missing traffic values within the incident duration. Therefore, we use linear data interpolation to fill gaps in the traffic data if required.
- *Train-Test-Split*: Finally, the input data is split into training and testing datasets. The former is used to train the ML model, allowing it to be generalized. Furthermore, the testing dataset evaluates the classification quality. The incident cases are randomly sampled and used within the training or testing dataset.

b) The model: We use k-NN algorithm, a well-known supervised learning approach to solve the classification problem. To train it, each incident entry has multiple features and a label referring to a particular incident type (accident, congestion, or non-incident). The data features represent a time series with the corresponding traffic level, speed (absolute and relative to the speed limit), and road type. k-NN was trained using two different distance metrics suitable to time series data. The DTW metric is used to measure the distance between two time series in the classification model. Furthermore, we train a model using the Wasserstein metric, a function to calculate the distance between two probability distributions μ and ν, defined in Eq. 5 [44].

$$W_p(\mu, \nu) := \left(\inf_{\gamma \in \Gamma(\mu,\nu)} \int_{\mathbb{R}x\mathbb{R}} d(x, y)^p d\gamma(x, y) \right)^{\frac{1}{p}} \quad (5)$$

We based our model on the k-NN-Implementation *K Nearest Neighbors with Dynamic Time Warping*,[2] modified and extended to support the Wasserstein metric. k-NN uses a parameter n (the number of neighbors) and the maximum warping window for the DTW, limiting the number of elements to compare and therefore reducing the execution time.

Moreover, data over and under-sampling are used to reduce the under-representation of accident samples in our imbalanced input dataset. To reduce the severity of this problem, two methods are used: i) *Oversampling*: Adds new samples of the minority class to the training data, using the information from the already existing data points. Our implementation includes *Random Oversampling* and *SMOTE Oversampling*; ii) *Undersampling*: Provides the contrary part by removing samples from a majority class. Our implementation includes *Nearmiss Undersampling*. These data sampling approaches are evaluated later in Section IV, further comparing the model quality using an imbalanced dataset.

Finally, the classification model is created using the parameters k (number of neighbors), warping window (comparison constraint), and metric (DTW or Wasserstein). Next, the model is trained, and each test data sample is classified using the trained model, returning a class label together with the corresponding probability. This method is implemented by calculating a distance matrix that contains the respective distance (DTW or Wasserstein) between all data samples regarding the chosen data feature. Using this matrix, our proposed algorithm can find the k closest neighbors and extract the most representative label for all test data samples.

IV. Evaluation

This section evaluates DataFITS by quantifying the improvements in data quality and quantity and presents a data characterization analysis from a real heterogeneous dataset. Finally, the enhanced fused data is used to evaluate the traffic estimation and incident classification models.

A. The Data Fusion Framework

1) The Data: The data acquisition process started on December 1st, 2021, and covers nine months of heterogeneous data from Bonn and Cologne. The acquired dataset from Bonn contains 13,700,000 entries with a total size of 14 GB, whereas the dataset from the neighboring city Cologne has 28,700,000 entries with a total size of 31 GB. The data is structured

[2] github.com/markdregan/K-Nearest-Neighbors-with-Dynamic-Time-Warping

TABLE II
COVERED ROADS BY DATA SOURCE

Source	Bonn			Cologne		
	Total Roads	Unique Roads	Fusion Portion	Total Roads	Unique Roads	Fusion Portion
Traf. HERE	684	339	21.0%	2940	1379	27.1%
Traf. OD	581	195	12.0%	914	173	3.4%
Inc. HERE	206	53	3.3%	946	370	7.3%
Inc. BING	597	256	15.8%	1944	821	16.2%
Inc. OD	52	31	1.9%	193	86	1.7%
Envirocar	433	178	11.0%	905	245	4.8%
Overlap	**567**	**567**	**35.0%**	**2007**	**2007**	**39.5%**
Total		1619			5081	

TABLE III
GENERAL TRAFFIC DATA STATISTICS

Road Type	Traffic	Entries	Traffic	Speed	Rel. Sp.
Main Road	Low	2,670,574 (81%)	0.86	37.74	69%
	Normal	527,150 (16%)	2.21	31.88	57%
	Increased	61,169 (02%)	5.07	17.68	32%
	Jammed	44,047 (01%)	9.79	15.37	31%
Motorway	Low	2,029,821 (80%)	0.41	83.79	87%
	Normal	296,694 (12%)	2.11	79.72	81%
	Increased	166,481 (07%)	4.91	53.78	57%
	Jammed	28,643 (01%)	8.74	27.42	29%
Residential	Low	286,654 (94%)	0.88	27.04	54%
	Normal	285 (00%)	2.27	13.76	28%
	Increased	16,218 (05%)	5.00	13.26	27%
	Jammed	3,062 (01%)	10.00	3.31	03%

into four types of information acquired from seven different data providers: i) Traffic data from the commercial service HERE and the open service Open Data (OD), containing data features like speed, traffic, and GPS coordinates; ii) Incident reports from the commercial services HERE, BING and OD, containing data features like the type of incident, GPS coordinates, and additional information; iii) Vehicular probe data from the Envirocar platform, providing in-depth data about the vehicle such as speed, fuel consumption, CO2 emissions, torque, throttle position, and more; and iv) Weather data from the Meteostat providing the weather conditions.

Commercial map services like Google, HERE, and Bing are the leading traffic data providers. They offer limited or paid access per user. Contrasting, projects like Open Data (OD) provide open access to data from multiple information categories. The goal is to create a collaborative data infrastructure that can be used by industry, academia, government, and civilian people, to design intelligent data-driven systems.

2) The Fusion: Table II emphasizes the benefits of heterogeneous data fusion by tabulating the number of roads covered by all sources and the ones that are spatiotemporally covered by multiple data sources, labeled with *Overlap*. Thus, the overlapped data correspond to multiple sources reporting data in the same location and time. Each source covers a number of total roads and provides a portion of unique roads to the fused dataset of Bonn and Cologne. For instance, Traffic HERE presents a proportion of 21% to the fused data for Bonn and 27% for Cologne. In contrast, OD contributes a significantly lower amount of information to the fused dataset, especially within the Cologne data at only 3.4%.

Regarding the incident data, a significant amount of additionally covered roads was added to the fused dataset, contributing 20-25% of new information. Furthermore, there is a substantial amount of extra information provided by Envirocar, especially in Bonn, with the probe vehicles covering areas that are not equipped with sensors and committing a portion of 11% to the fused data. This can be explained by the fact that the users contributing to the platform also collect data in many residential areas that are usually not equipped with sensors. The amount of overlapping road segments reaches 35% for Bonn and 39.5% in Cologne, revealing the potential of information enrichment using heterogeneous data fusion. These numerical results show that we can utilize fused data to better describe the transportation system status and improve the amount of information compared to only using a single data source. For instance, the Traffic HERE solely covers 684 roads in Bonn and 2940 in Cologne, while the fused data covers 1619 and 5081 roads, respectively. This is a data enrichment of 137% in Bonn and 173% in Cologne.

3) Data Characterization: DataFITS presents general traffic statistics as a function of time, day, or road type. It helps in analyzing collected and fused data. The traffic values are grouped into levels: Low (0-1), Normal (>1-4), Increased (>4-7), and Jammed (>7-10) over three types of roads: Motorway, Main Road, and Residential. Additionally, it provides the characterization of incident data from Bonn, which includes four incident types, namely *Accident*, *Congestion*, *Disabled Vehicle*, and *Road Hazard*.

a) Traffic data: Table III lists the number of data entries, in Bonn, for each traffic level on different road types, including the average values for traffic, speed, and relative speed ($\frac{speed}{speed\ limit}$). One can observe a similar distribution of traffic for different road types. The low-traffic level presents more entries, but there is a variation between the distributions related to the various road types. On main roads, nearly 97% of all data entries represent a low or normal traffic level. On motorways, the amount is 92%, a difference of 5%, distributed in the other two levels (increased and jammed). The traffic in residential areas is mainly represented in a low (94%) or increased level (5%).

Table III also lists the traffic and speed (absolute and relative). The relative speed varies significantly on the three road types, decreasing from nearly 70% in a low traffic level to 30% in a jammed condition on a main road. A similar pattern is observed on motorways with higher relative speeds at the first two traffic levels. In residential areas, the speed reaches a maximum of 54% and significantly decreases to 3% when jammed. This characterization suggests that the city of Bonn has a low traffic level most of the time (>80% of all data entries). Higher traffic levels were observed in seven percent of all data entries, with the proportion of a jammed condition representing one percent. This depicts a realistic behavior, as congestion generally emerges during rush hours or in case of specific incidents. Moreover, Table III shows a significant reduction in speed for high traffic levels (seven or more), especially in the case of residential areas. A low average speed on the residential roads is also noticed, with less than $27\,\mathrm{km\,h^{-1}}$, representing a safe value to reduce noise, pollution, and the probability of fatal accidents.

In summary, the traffic data characterization suggests that the city demands different traffic management strategies based

TABLE IV
GENERAL INCIDENT DATA STATISTICS

Inc Type	Road Type	Entries	Dur. (avg)	Dur. (max)
Accident	Main Road	4 (01,48%)	10.00 min	10 min
	Motorway	260 (95.94%)	15.92 min	170 min
	Residential	7 (02.58%)	25.71 min	120 min
Congestion	Main Road	326 (07.37%)	12.98 min	90 min
	Motorway	3892 (88.03%)	19.15 min	730 min
	Residential	203 (04.59%)	11.43 min	50 min
Dis. Vehicle	Main Road	9 (02.58%)	10.00 min	10 min
	Motorway	328 (93.98%)	15.88 min	210 min
	Residential	12 (03.44%)	18.33 min	60 min
Road Hazard	Main Road	11 (02.49%)	10.91 min	20 min
	Motorway	412 (93.42%)	15.29 min	110 min
	Residential	18 (04.08%)	18.33 min	110 min

on road type and location to improve its quality and safety. Due to the limited space, more spatiotemporal data characterization is provided by the DataFITS on the git repository.[1]

b) Incident data: Table IV lists the number of reports for each type of incident, grouped by different road types. As expected, the amount of congestion reports surpasses the number of all other incident types within the dataset because congestion is a re-occurring event, generally emerging due to high traffic. Furthermore, a significant reduction in incidents is shown when comparing motorways to the other two discussed road types. Besides the higher traffic on motorways in general, this observation matches the results of the previous traffic data characterization, suggesting that there is a lower speed on main roads, especially in residential areas, reducing the probability of accidents. Finally, Table IV includes the average and maximum duration regarding each group of incident reports, showing significant differences, especially for the maximum duration of congestion on a motorway reaching 730 minutes, compared to not more than 210 minutes for the other groups.

By showing traffic and incident events together, it is possible to identify the effects of a single incident on the traffic levels in the surrounding area, as shown in Fig. 5. The accident (marked by 'X') was reported on a motorway at 17:30, and the traffic levels on the surrounding roads over time are shown with green to red colors, from no traffic (level 0) to high traffic (level 10), respectively. Before the accident, a low traffic level around the incident location can be observed, but the traffic increases in the directly connected areas 10 minutes before the incident is reported in the respective source. This condition can be observed further, escalating to a jammed road at 17:30 with very high traffic on the connected roads. It reverts to the initial state at 17:50, indicating that the incident lasts about 30 minutes. Therefore, the accident impacts the traffic pattern of many neighboring roads, especially between 17:30 and 17:40.

DataFITS includes further data characterization by combining incidents with different weather conditions and seasons, showing a more significant number of incidents reported in worse weather conditions (e.g., rain and snow) than in normal conditions. Moreover, analyzing the different road types, most incidents occur on motorways, mainly at two certain intersections that are important parts of the transportation system. For more data characterization, please see the DataFITS repository.[1]

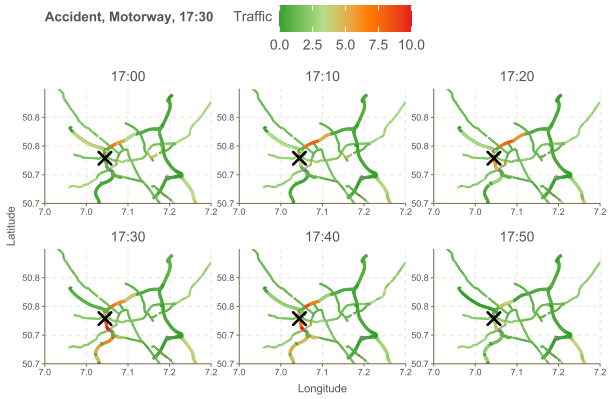


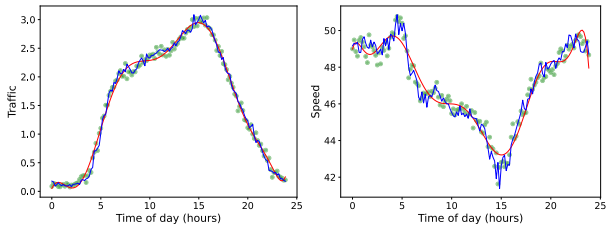
Fig. 5. Incident effect on the traffic level.

B. Traffic Estimation

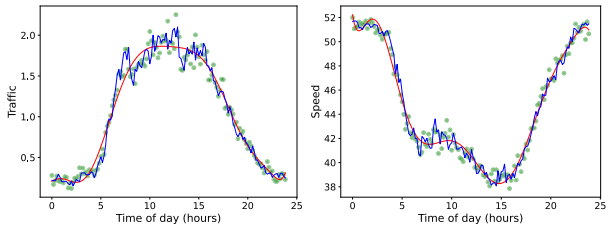
This section reports a comparative evaluation of the estimation model introduced in Section III-B.1 and the model introduced in a previous paper [43]. Both were trained with the heterogeneous fused dataset characterized in the earlier sections. To measure the performance of each model, we calculate three different performance metrics: i) Coefficient of Determination (R^2) represents the variance proportion between the true and estimated values. The optimal value is one, and zero is given to a model that always predicts the average of the true value y. ii) Mean Absolute Error (MAE) is an error metric that describes the sum of absolute errors between the real and estimated values, aiming for a value close to zero. iii) Root Mean Squared Error (RMSE) denotes the root of the Mean Squared Error (MSE), a measurement for the average squared distance between the estimated values by the model and the real values within the dataset, also having a desired value of zero. We compare the model's performances against each other, using different input parameters and a fused vs. non-fused dataset.

The proposed model estimated the traffic level for 181 different traffic areas in Bonn and was trained using a dataset of more than 7 million entries using the following thresholds: $th_{overlap} = 0.50, th_{cor} = 0.90$ and $th_{dtw} = 0.25$. Within the experimental setup, we used a train-test split of 60-40 and a polynomial degree of 10, reflecting the optimal model configuration based on extensive experiments using different polynomial degrees. All estimations shown here use a time frame of 24 hours. The remaining input parameters are stated within each result throughout this section.

Fig. 6 depicts four examples of traffic estimation for two different regions (A and B), for an entire day, with the time displayed in hours on the x-axis. It shows the estimated regression line (red) on the training dataset (green dots). For better visualization, we grouped the observations within the training data and showed the mean values. The blue line depicts the test data, representing real-world traffic data. Noticeably, the estimation in Fig. 6(a) shows a high precision on both traffic (left) and speed (right), suggesting that the model fits well on the prepared input dataset. The second area depicted in Fig. 6b shows a similar result, scoring an even higher performance and providing a very close estimation compared to the ground truth.



(a) Traffic (left) and speed (right) estimation on region A.



(b) Traffic (left) and speed (right) estimation on region B.

Fig. 6. Comparison: Estimation and real data.

TABLE V
PERFORMANCE OF THE REGRESSION MODEL ON VARIOUS STREETS AND WEATHER CONDITIONS

	Road Type	Weather	Feature	R^2	MAE	RMSE
↑	All	All	Traffic	0.76	0.07	0.09
			Speed	0.84	0.06	0.08
●	Motorway	All	Traffic	0.68	0.08	0.10
			Speed	0.71	0.08	0.11
↑	Main Road	All	Traffic	0.81	0.06	0.08
			Speed	0.91	0.05	0.07
↑	Residential	All	Traffic	0.75	0.07	0.10
			Speed	0.91	0.05	0.07
↑	All	Clear	Traffic	0.74	0.07	0.09
			Speed	0.82	0.06	0.09
↓	All	Rain	Traffic	0.55	0.08	0.11
			Speed	0.69	0.08	0.11
↓	All	Snow	Traffic	0.23	0.09	0.15
			Speed	0.28	0.11	0.16

Furthermore, the model can be configured to differentiate road types and weather conditions. Table V shows the overall performance of the traffic estimation from all 181 areas when using different configurations. The first line at the top represents the performance measures of the model using the entire dataset without any additional filter. In total, the model achieves a high R^2 score of 0.84, using speed or relative speed, while also performing well estimating the traffic value reaching a score of 0.76. Both error measures are low for each respective data feature, represented by values below 0.10. By differentiating the road types and weather conditions, it is noticeable that the model achieves the best performance estimating traffic on main roads with no specified weather conditions, achieving R^2 scores of 0.91 using the speed data and 0.81 using traffic. The general performance on motorways (orange circle) is slightly lower compared to both main roads and residential areas (green up arrow), reaching an R^2 score of up to 0.84.

The same measurements using different weather conditions are shown in the last three rows of Table V. Noticeably, the performance is significantly lower in estimating traffic in case of rain, and especially snow (indicated by the red down arrows). This is due to the low amount of traffic observations within our data during rain or snow, significantly reducing the size of the training dataset. The estimation uses many interpolated data points instead of real values, reducing the overall quality. However, the model can accurately estimate traffic features in clear weather conditions (green up arrow). This exploratory investigation suggests the model performs well on most input data parameters. However, using separate configurations to estimate the traffic based on a specific road type achieves the best results. In contrast, creating a model based on the weather conditions, rain, and snow reduces the quality of the model's estimations.

TABLE VI
PERFORMANCE OF THE REGRESSION MODEL ON VARIOUS INPUT DATASETS

	Feature	Dataset	Unique Area Coverage	R2	MAE	RMSE
●	Traffic	FUSED	**181** (36 ovlp.)	0.76	0.07	0.09
↑		HERE	94 (43.3%)	0.81	0.06	0.09
↓		OD	123 (56.7%)	0.66	0.08	0.10
↑	Speed	FUSED	**181** (36 ovlp.)	0.84	0.06	0.08
●		HERE	94 (43.3%)	0.81	0.07	0.09
●		OD	123 (56.7%)	0.83	0.06	0.08

TABLE VII
COMPARISON OF STATISTICAL AND REGRESSION MODEL

Metric	Dataset	Descriptive Statistics Model	Regression Model
R2	2 months	-0.9	0.15
	9 months	-0.57	0.68
MAE	2 months	0.3	0.43
	9 months	0.23	0.16

1) Single Vs. Fused Data: Due to the limited public access to source code and data of most of the literature as discussed in Section II, we compared the polynomial regression approach on the non-fused datasets, containing information that was obtained from a single source, with the fused dataset in order to compile quantitative evidence that data fusion shows the benefits in precision and coverage as listed in Table VI. Estimating traffic values, the single data source HERE scores the best results, indicating that the fused dataset is biased by the poor performance of the OD dataset. However, on the estimation of speed values, the fused dataset achieves the highest performance of 0.84 for the R^2 and error metrics of 0.06 and 0.08. By fusing multiple datasets, we can combine their individual benefits and achieve a minor improvement in the model quality. Furthermore, the area coverage within the combined data is much higher, comparing 94 unique areas covered by HERE, 123 by OD, and 217 areas covered by the fused information, represented through 181 unique and 36 overlapping areas from both datasets. These results demonstrate that although a single data source can perform better in some cases, it is still limited in spatiotemporal coverage, limiting the model's generalization. A similar result was achieved by comparing the statistical model using the fused vs. non-fused dataset in our previous study [43].

2) Descriptive Vs. ML-Based Model: When evaluating the use of data from correlating areas in the training dataset, we noticed no further improvement in the polynomial regression model. However, the descriptive statistics model showed a substantial improvement in the estimation quality using the correlation approach. In general, the model based on ML,

proposed in this article, achieves significantly better results compared to the descriptive statistics approach presented in [43], as shown in Table VII. On the dataset containing nine months of data, we could improve the average R^2 score from 0.15 to 0.68 and achieve significantly lower error metrics. However, the descriptive statistics model is not relying on a large amount of data and, therefore, may be useful in the case of small input datasets. In conclusion, we could identify that using the polynomial regression model on our entire heterogeneous fused dataset provides promising results, reaching an R^2 score of up to 0.91. However, a less complex and costly model (computation and time), such as using descriptive statistics, can be applied in the case of a reduced amount of data available.

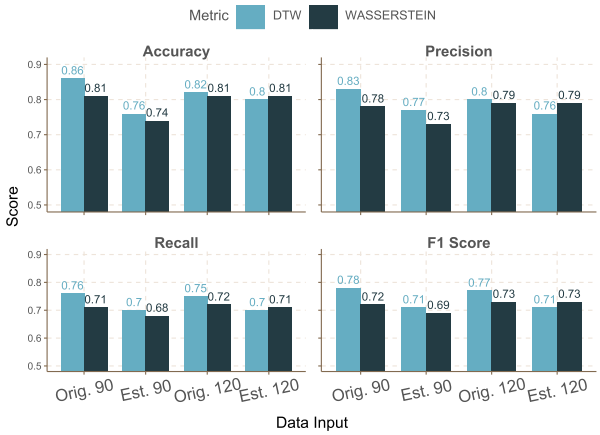


Fig. 7. Comparison: Different input strategies.

C. Incident Classification

We compare the performance and training of our proposed incident classification model for two different approaches: A binary classification (*Incident* or *Non-Incident*) and a multi-class approach (*Accident*, *Congestion*, and *Non-Incident*). The classification relies on the fused dataset combining the information on traffic and incidents. Therefore, all proposed results are generated on the heterogeneous fused dataset. First, we evaluate the performance of the presented data preparations, referring to ii) incident validation; iii) over- and undersampling; and iv) various time intervals.

The proposed incident validation approach, which is a part of the data preparation presented earlier in Section III-B.2, provides a major accuracy improvement. Precisely, it improves the overall performance by 26%, e.g., increasing the accuracy from 0.7 on a not validated input dataset to 0.86 after the validation.

Regarding the imbalanced dataset problem, we use *SMOTE Oversampling* to improve the precision of classifying accident data samples from 0.35 to 0.77 while keeping high precision at all other data classes. Using SMOTE improves the precision by 15% from 0.72 to 0.83. In contrast, using the *Nearmiss Undersampling* method also showed a minor benefit on the precision score of accident data samples but reduced the total score from 0.72 to 0.59. Based on this investigation, we apply the incident validation approach and the SMOTE oversampling in the final configuration of the classification model.

We compare the model's performance using the originally reported start time of each incident against the idea of estimating a more realistic start time, obtained by iterating over the time series, as presented in Section III-B.2. Furthermore, we evaluated two different time intervals of 90 or 120 minutes before and after the incident start time (original start time or estimated start time), respectively. Fig. 7 shows all performance metrics over the different time interval strategies. Generally, the traffic patterns during the 90-minute time intervals achieve the highest overall scores, with an accuracy of 0.86 for the DTW and 0.81 for Wasserstein distance metrics. Furthermore, we noticed an advantage of using the (original) reported start time, in the 90-minute time interval, compared to our estimated incident start time approach. In contrast, the larger time interval (120 minutes) shows only minor performance differences between using the original and estimated starting point. Moreover, the Wasserstein metric achieves a better performance on the larger input intervals, compared to the DTW in the case of the estimated incident start time.

In conclusion, our final model uses input data covering a 90-minute time frame before and after the originally reported incident time, as this provides the overall best results. The conducted evaluation shows that using the iterative approach to estimate a more realistic start point of an incident does not benefit the model's accuracy. Moreover, increasing the input time interval to 120 minutes shows a minor decrease in the overall performance but benefits the model in all test setups that use the Wasserstein metric, especially when working with the estimated start time.

Finally, we compare the performance of the three-class classification model with the binary classification. Evaluating the binary classification that distinguishes between *Incident* and *Non-Incident*, we achieve an overall accuracy and an F1 score of 90%, with no differences between the number of samples in both classes, due to a balanced input dataset. Furthermore, DTW shows a minor advantage compared to using the Wasserstein metric, improving 5% on all metrics on average. The three-class model adds complexity by classifying the types of incidents (*Accident*, *Congestion*, and *Non-Incident*). and achieves an average accuracy of 86%, slightly lower than the binary model. When considering the performance of each class, the problem of data imbalance remains even after using data oversampling. This is reflected by the significantly lower performance to identify the *Accident* data as shown by comparing the F1 scores of 0.56 to 0.9 on other classes. To obtain a more generalized and accurate model, we proposed a completely balanced dataset, combining over- and undersampling methods, achieving better accuracy scores of 80% and an F1 score of 0.78 on all data classes.

V. Conclusion

In this paper, we introduce DataFITS, an open-source data fusion framework that integrates diverse data by collecting, analyzing, and fusing it. We hypothesize that heterogeneous data fusion increases data quantity and quality, thereby improving datasets for ITS applications. To verify this, we developed two ITS applications: one used polynomial regression to estimate traffic levels, while the other combined

traffic and incident data to classify events into accident, congestion, or non-incidents.

Using real heterogeneous data from two German cities, we quantified the advantages of DataFITS by compiling a fused dataset. Our results indicate that DataFITS integrated data from multiple sources for 40% of all roads, thereby increasing the overall road coverage by 137%. In addition, the traffic estimation model, which uses polynomial regression, outperformed our previous approach based on descriptive statistics, achieving a high R^2 score of 0.91, low error metrics of 0.05, and provides accurate traffic estimations using the fused dataset. Compared to using a single sources dataset, the fused dataset estimation showed minor accuracy improvements but drastically improved the spatiotemporal coverage of the estimated areas. Our incident classification model relies on the fusion of traffic and incident data, achieving a 90% binary classification accuracy rate within our evaluation. Preprocessing the data, such as removing unclear traffic patterns, improved accuracy by an average of 29%. The classification of incidents into different categories resulted in a slightly lower accuracy of 86%, with unequal performance among classes indicated by F1 scores. To mitigate this problem, we oversampled the training dataset to create a more uniform representation of the data, resulting in an 80% accuracy for each class. Collecting more accident data can also solve this problem.

We plan to expand the DataFITS framework by collecting and fusing more data types, improving its performance and data quality, and expanding its data analysis. We focus on data types such as social media and images, which require methods such as Natural Language Processing (NLP) and image processing. For ITS applications, we aim to use automated machine learning to explore different models and hyper-parameters and compare them with our current models. We also plan to analyze the correlation between traffic and incidents and incorporate it into the traffic estimation models. In addition, we intend to explore the use of big data in military scenarios, combining information from the civilian and military fields to support strategic operations in urban warfare. To this end, our framework can be enhanced to collect and combine different types of information (image, text) to create common operational pictures and verify/authenticate information, thereby avoiding misinformation that may influence political decisions.

REFERENCES

[1] L. Zhu, F. R. Yu, Y. Wang, B. Ning, and T. Tang, "Big data analytics in intelligent transportation systems: A survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 1, pp. 383–398, Jan. 2019.

[2] Umweltbundesamt. (2022). *Verkehrsinfrastruktur und fahrzeugbestand*. Accessed: Dec. 12, 2022. [Online]. Available: https://www.umweltbundesamt.de/daten/verkehr/verkehrsinfrastruktur-fahrzeugbestand

[3] German Federal Statistical Office (Destatis). (2022). *Passengers Carried in Germany*. Accessed: Jul. 12, 2022. [Online]. Available: https://www.destatis.de/EN/Themes/Economic-Sectors-Enterprises/Transport/Passenger-Transport/Tables/passengers-carried.html

[4] G. Vítor, P. Rito, and S. Sargento, "Smart city data platform for real-time processing and data sharing," in *Proc. IEEE Symp. Comput. Commun. (ISCC)*, Sep. 2021, pp. 1–7.

[5] A. B. Campolina, P. H. L. Rettore, M. Do Val Machado, and A. A. F. Loureiro, "On the design of vehicular virtual sensors," in *Proc. 13th Int. Conf. Distrib. Comput. Sensor Syst. (DCOSS)*, Jun. 2017, pp. 134–141.

[6] S. Jeong, S. Kim, and J. Kim, "City data hub: Implementation of standard-based smart city data platform for interoperability," *Sensors*, vol. 20, no. 23, p. 7000, Dec. 2020. [Online]. Available: https://www.mdpi.com/1424-8220/20/23/7000

[7] L. Zhang, Y. Xie, L. Xidao, and X. Zhang, "Multi-source heterogeneous data fusion," in *Proc. Int. Conf. Artif. Intell. Big Data (ICAIBD)*, May 2018, pp. 47–51.

[8] P. H. L. Rettore, B. P. Santos, A. B. Campolina, L. A. Villas, and A. A. F. Loureiro, "Towards intra-vehicular sensor data fusion," in *Proc. IEEE 19th Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2016, pp. 126–131.

[9] P. H. L. Rettore, A. B. Campolina, L. A. Villas, and A. A. F. Loureiro, "A method of eco-driving based on intra-vehicular sensor data," in *Proc. IEEE Symp. Comput. Commun. (ISCC)*, Jul. 2017, pp. 1122–1127.

[10] P. H. L. Rettore, A. B. Campolina, A. Souza, G. Maia, L. A. Villas, and A. A. F. Loureiro, "Driver authentication in VANETs based on intra-vehicular sensor data," in *Proc. IEEE Symp. Comput. Commun. (ISCC)*, Jun. 2018, pp. 00078–00083.

[11] G. L. Foresti, M. Farinosi, and M. Vernier, "Situational awareness in smart environments: Socio-mobile and sensor data fusion for emergency response to disasters," *J. Ambient Intell. Humanized Comput.*, vol. 6, no. 2, pp. 239–257, Apr. 2015.

[12] H. Wen, Y. Lin, and J. Wu, "Co-evolutionary optimization algorithm based on the future traffic environment for emergency rescue path planning," *IEEE Access*, vol. 8, pp. 148125–148135, 2020.

[13] P. H. Rettore, G. Maia, L. A. Villas, and A. A. F. Loureiro, "Vehicular data space: The data point of view," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 3, pp. 2392–2418, 3rd Quart., 2019.

[14] S. A. Kashinath et al., "Review of data fusion methods for real-time and multi-sensor traffic flow analysis," *IEEE Access*, vol. 9, pp. 51258–51276, 2021.

[15] W. Jiang and J. Luo, "Big data for traffic estimation and prediction: A survey of data and tools," *Appl. Syst. Innov.*, vol. 5, no. 1, p. 23, Feb. 2022.

[16] R. A. Anand, L. Vanajakshi, and S. C. Subramanian, "Traffic density estimation under heterogeneous traffic conditions using data fusion," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2011, pp. 31–36.

[17] A. Abadi, T. Rajabioun, and P. A. Ioannou, "Traffic flow prediction for road transportation networks with limited traffic data," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 2, pp. 653–662, Apr. 2015.

[18] G. Meena, D. Sharma, and M. Mahrishi, "Traffic prediction for intelligent transportation system using machine learning," in *Proc. 3rd Int. Conf. Emerg. Technol. Comput. Eng., Mach. Learn. Internet Things (ICETCE)*, Feb. 2020, pp. 145–148.

[19] L. Zhao et al., "T-GCN: A temporal graph convolutional network for traffic prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 9, pp. 3848–3858, Sep. 2020.

[20] J. Tang, L. Li, Z. Hu, and F. Liu, "Short-term traffic flow prediction considering spatio-temporal correlation: A hybrid model combing type-2 fuzzy C-means and artificial neural network," *IEEE Access*, vol. 7, pp. 101009–101018, 2019.

[21] X. Di, Y. Xiao, C. Zhu, Y. Deng, Q. Zhao, and W. Rao, "Traffic congestion prediction by spatiotemporal propagation patterns," in *Proc. 20th IEEE Int. Conf. Mobile Data Manage. (MDM)*, Jun. 2019, pp. 298–303.

[22] X. Wang, X. Guan, J. Cao, N. Zhang, and H. Wu, "Forecast network-wide traffic states for multiple steps ahead: A deep learning approach considering dynamic non-local spatial correlation and non-stationary temporal dependency," *Transp. Res. C, Emerg. Technol.*, vol. 119, Oct. 2020, Art. no. 102763. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0968090X20306756

[23] C. Zheng, X. Fan, C. Wang, and J. Qi, "GMAN: A graph multi-attention network for traffic prediction," in *Proc. AAAI*, vol. 34, no. 1, 2020, pp. 1234–1241.

[24] B. Zhao, X. Gao, J. Liu, J. Zhao, and C. Xu, "Spatiotemporal data fusion in graph convolutional networks for traffic prediction," *IEEE Access*, vol. 8, pp. 76632–76641, 2020.

[25] N. Zafar, I. U. Haq, J.-U.-R. Chughtai, and O. Shafiq, "Applying hybrid LSTM-GRU model based on heterogeneous data sources for traffic speed prediction in urban areas," *Sensors*, vol. 22, no. 9, p. 3348, Apr. 2022. [Online]. Available: https://www.mdpi.com/1424-8220/22/9/3348

[26] Z. Shan, Y. Xia, P. Hou, and J. He, "Fusing incomplete multisensor heterogeneous data to estimate urban traffic," *IEEE MultimediaMag.*, vol. 23, no. 3, pp. 56–63, Jul. 2016.

[27] A. Genser, N. Hautle, M. Makridis, and A. Kouvelas, "An experimental urban case study with various data sources and a model for traffic estimation," *Sensors*, vol. 22, no. 1, p. 144, Dec. 2021.
[28] L. Wenqi, L. Dongyu, and Y. Menghua, "A model of traffic accident prediction based on convolutional neural network," in *Proc. 2nd IEEE Int. Conf. Intell. Transp. Eng. (ICITE)*, Sep. 2017, pp. 198–202.
[29] S.-H. Park, S.-M. Kim, and Y.-G. Ha, "Highway traffic accident prediction using VDS big data analysis," *J. Supercomput.*, vol. 72, no. 7, pp. 2815–2831, Jul. 2016.
[30] H. Ren, Y. Song, J. Wang, Y. Hu, and J. Lei, "A deep learning approach to the citywide traffic accident risk prediction," in *Proc. 21st Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2018, pp. 3346–3351.
[31] Q. Shang, L. Feng, and S. Gao, "A hybrid method for traffic incident detection using random forest-recursive feature elimination and long short-term memory network with Bayesian optimization algorithm," *IEEE Access*, vol. 9, pp. 1219–1232, 2021.
[32] Z. Liu and C. Wang, "Design of traffic emergency response system based on Internet of Things and data mining in emergencies," *IEEE Access*, vol. 7, pp. 113950–113962, 2019.
[33] K. R. Sanjana, S. Lavanya, and Y. B. Jinila, "An approach on automated rescue system with intelligent traffic lights for emergency services," in *Proc. Int. Conf. Innov. Inf., Embedded Commun. Syst. (ICIIECS)*, Mar. 2015, pp. 1–4.
[34] P. H. L. Rettore, B. P. Santos, R. Rigolin F. Lopes, G. Maia, L. A. Villas, and A. A. F. Loureiro, "Road data enrichment framework based on heterogeneous data fusion for ITS," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 4, pp. 1751–1766, Apr. 2020.
[35] A. Salas, P. Georgakis, and Y. Petalas, "Incident detection using data from social media," in *Proc. IEEE 20th Int. Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2017, pp. 751–755.
[36] S. Guo et al., "Identifying the most influential roads based on traffic correlation networks," *EPJ Data Sci.*, vol. 8, no. 1, pp. 1–17, Dec. 2019.
[37] Z. Liu, Z. Li, M. Li, W. Xing, and D. Lu, "Mining road network correlation for traffic estimation via compressive sensing," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 7, pp. 1880–1893, Jul. 2016.
[38] Y. Zhu, Z. Li, H. Zhu, M. Li, and Q. Zhang, "A compressive sensing approach to urban traffic estimation with probe vehicles," *IEEE Trans. Mobile Comput.*, vol. 12, no. 11, pp. 2289–2302, Nov. 2013.
[39] B. P. Santos, P. H. L. Rettore, H. S. Ramos, L. F. M. Vieira, and A. A. F. Loureiro, "Enriching traffic information with a spatiotemporal model based on social media," in *Proc. IEEE Symp. Comput. Commun. (ISCC)*, Jun. 2018, pp. 00464–00469.
[40] G. Boeing, "OSMnx: New methods for acquiring, constructing, analyzing, and visualizing complex street networks," *Comput., Environ. Urban Syst.*, vol. 65, pp. 126–139, Sep. 2017.
[41] C. Yang and G. Gidófalvi, "Fast map matching, an algorithm integrating hidden Markov model with precomputation," *Int. J. Geographical Inf. Sci.*, vol. 32, no. 3, pp. 547–570, Mar. 2018.
[42] R. Tavenard. *An Introduction to Dynamic Time Warping*. Accessed: Sep. 14, 2022. [Online]. Available: https://rtavenar.github.io/blog/dtw.html
[43] P. Zißner, P. H. L. Rettore, B. P. Santos, R. R. F. Lopes, and P. Sevenich, "Road traffic density estimation based on heterogeneous data fusion," in *Proc. IEEE Symp. Comput. Commun. (ISCC)*, Jun. 2022, pp. 1–6.
[44] S. Kolouri, S. R. Park, M. Thorpe, D. Slepcev, and G. K. Rohde, "Optimal mass transport: Signal processing and machine-learning applications," *IEEE Signal Process. Mag.*, vol. 34, no. 4, pp. 43–59, Jul. 2017.

Philipp Zißner received the B.Sc. and M.Sc. degrees in computer science from Rheinische Friedrich-Wilhelms-Universität Bonn in 2020 and 2022, respectively. He is currently a Scientist with the Communication Systems Department (KOM), Fraunhofer FKIE, Bonn, Germany. His research interests include intelligent transportation systems, smart mobility, the Internet of Things, and tactical networks.

Paulo H. L. Rettore received the B.Sc. and M.Sc. degrees in computer science in 2009 and 2012, respectively, and the Ph.D. degree in computer science from the Federal University of Minas Gerais (UFMG) in 2019. He is currently a Scientist with Fraunhofer FKIE, Bonn, Germany. Sitting with the Communication Systems Department (KOM), he has been focused on measuring the performance bounds of tactical systems over ever-changing scenarios. His research interests include computer networks, mobile ad-hoc networks, tactical networks, software-defined networking, ubiquitous computing, the Internet of Things, intelligent transportation systems, and smart mobility.

Bruno P. Santos received the bachelor's degree from Universidade Estadual de Santa Cruz (UESC) and the M.S. and Ph.D. degrees in computer science from Universidade Federal de Minas Gerais (UFMG). He is currently a Professor of computer science with the Federal University of Bahia (UFBA). His research interests include computer networks, distributed systems, ubiquitous computing, the Internet of Things, intelligent transportation systems, and smart mobility.

Johannes F. Loevenich received the B.Sc. degree in computer science and the B.Sc. degree in mathematics from Rheinische Friedrich-Wilhelms-Universität Bonn. He is currently pursuing the Ph.D. degree in computer science/mathematics with the Distributed Systems Department, University of Osnabrück. He is a Scientist with the Communication Systems Department (KOM), Fraunhofer FKIE, Bonn, Germany. His research interests include computer systems, computer networks, distributed systems, data science, optimization theory, artificial intelligence, and game theory.

Roberto Rigolin F. Lopes (Member, IEEE) received the B.Sc. degree in computer science from UFMT, Brazil, the M.Sc. degree in computer science from UFSCar, Brazil, and the Ph.D. degree in computer science from USP, Brazil. During his Ph.D., he also visited Twente, The Netherlands, and Ottawa, Canada. After his Ph.D., he got a post-doctoral scholarship from the European Research Consortium for Informatics and Mathematics (ERCIM) to join NTNU, Norway, for four years, and a Scientist with Fraunhofer FKIE, Germany, for six years. He is currently a Scientist with Thales Deutschland, Ditzingen, Germany. Sitting with the Secure Communications and Information Systems (SIX), he has been attacking problems in computer networks and distributed systems with a particular interest in the performance bounds of tactical systems over ever-changing communication scenarios. His academic life triggered interesting life experiences, but he has been rebuilding his own education following curiosity freely by reading books on physics, mathematics, and philosophy.