2024 IEEE 27th International Conference on Intelligent Transportation Systems (ITSC)

IncidentNet: Traffic Incident Detection, Localization and Severity Estimation with Sparse Sensing

Sai Shashank Peddiraju[1,*], Kaustubh Harapanahalli[2,*], Edward Andert[2] and Aviral Shrivastava[2]
Arizona State University, Tempe
{speddira,kharapan,eandert,aviral.shrivastava}@asu.edu

***Abstract*— Prior art in traffic incident detection relies on high sensor coverage and is primarily based on decision-tree and random forest models that have limited representation capacity and, as a result, cannot detect incidents with high accuracy. This paper presents IncidentNet - a novel approach for classifying, localizing, and estimating the severity of traffic incidents using deep learning models trained on data captured from sparsely placed sensors in urban environments. Our model works on microscopic traffic data that can be collected using cameras installed at traffic intersections. Due to the unavailability of datasets that provide microscopic traffic details and traffic incident details simultaneously, we also present a methodology to generate a synthetic microscopic traffic dataset that matches given macroscopic traffic data. IncidentNet[1] achieves a traffic incident detection rate of 98%, with false alarm rates of less than 7% in 197 seconds on average in urban environments with cameras on less than 20% of the traffic intersections.**

I. Introduction

In 2019, traffic accidents alone caused approximately 28 million incidents, risking people's safety [1]. According to the study [2] conducted across 2268 US counties, a 5-minute delay in emergency response increased fatality rates by 46%, while response times under 7 minutes reduced fatality rates by 58% in urban and rural areas. Along with traffic accidents, cargo spills, stalled vehicles, road maintenance, and other emergency scenarios are also traffic incidents. Traffic incidents are generally defined as non-recurring events that reduce the roadway's capacity [3]. These incidents lead to secondary issues such as road congestion and delayed emergency support [4]. This motivates the need to work towards detecting traffic incidents quickly, improving emergency response time, and improving traffic re-routing time.

Faster and more accurate incident detection presents two main challenges. (i) Need for an algorithm to detect, locate, and estimate the severity of incidents in urban regions: Most existing traffic incident detection algorithms, such as [5], are tailored for highways. However, the existing algorithms for urban regions, like [6], introduced an algorithm that compares current traffic conditions, including travel times, to a predefined threshold, and [7] proposed a pattern-matching algorithm that uses a database of GPS trajectories to identify incidents. However, the performance of such comparative and pattern-matching algorithms heavily depends on thresholds, requiring continuous adjustment due to traffic's dynamic nature. (ii) Non-availability of microscopic datasets: Existing well-known public datasets like PEMS [8], San Francisco I-880 [9], and METR-LA [10] primarily use inductive loop detectors to capture macroscopic data focusing on highways by aggregating metrics like average vehicle speed and average flow-rate density obtained through these sensors without any vehicle distinguishing features. This level of aggregation makes it challenging to get high accuracy in dynamic urban settings. Alternatively, datasets using GPS sensors like NYC Taxi Data [11] and Bluetooth sensors like Highway 99-W [6] offer microscopic features but suffer from issues like signal loss and interference and data latency [12], [13], hindering their use in time-critical traffic incident detection tasks.

Owing to the vast development of Computer Vision and the quality of cameras over the last decade, the deployment and utility of cameras for traffic use cases have increased in urban environments and highways [14]. They can capture microscopic data like speed, location, timestamp, direction, and unique vehicle identifiers for each vehicle. Due to this, developments have focused on traffic incident detection approaches within the camera's field of view [15]. However, incidents outside their field of view remain undetected. Deploying cameras to increase coverage to 100% is challenging and not desirable. So, in this paper, we develop methods to identify incidents outside the camera's field of view using existing infrastructure, even with sparse coverage of roads in urban regions. We address these challenges through our two key contributions:

- A repeatable approach for generating realistic fine-grain synthetic datasets using traffic flow data within a microscopic traffic simulator, facilitating researchers with more realistic data. Our method takes readily available coarse-grain public traffic flow data. It generates a synthetic dataset using traffic data within a simulator that closely matches the coarse-grain distributions of the public traffic flow real-world dataset.
- A novel technique that can detect and localize a traffic incident without the incident being directly in the field of view of a visual sensor. Localization of the incident is achievable without knowing the precise distance

This work was partially supported by funding from National Science Foundation grants CPS 1645578 and Semiconductor Research Corporation (SRC) project 3154.

∗: Equal Contributions from the authors

1: School for Engineering of Matter, Transport and Energy, 2: School of Computing and Augmented Intelligence

[1] https://github.com/MPSLab-ASU/IncidentNet

between sensors. This incident detection technique is also robust to sparse sensor placement in urban regions.

We generated a synthetic dataset for Tempe, for 12 separate urban backbone roads for an area of about 4 square miles [16] with a traffic approximation model and confirmed by the Kolmogorov-Smirnov test [17]. TabNet [18] models were trained on 31 days of simulated data. In an urban region, IncidentNet detection rate of traffic incidents was 98%, the mean time to detect incidents was 197.44 seconds, and the false alarm rate was a mere 6.26% with a sensor sparsity of 81.4%. Furthermore, applied to a highway scenario, IncidentNet achieved a detection rate of 99% with a false alarm rate of 4.17%, making it suitable to tackle both environments.

II. Related Works

A. Challenges of Macroscopic Datasets

The PEMS [8] Bay dataset collects traffic data using inductive loop detectors placed throughout the highways in the Bay area and other parts of California. Traffic metrics like average speed, occupancy, and vehicle count are gathered and aggregated at 5-10 minute intervals without distinguishing information about individual vehicles. I-880 [9] and METR-LA [10] also capture macroscopic data through inductive loop detectors, similar to the PEMS dataset. These datasets (i) don't capture the nuance details essential for better accuracy detection in urban areas, and (ii) primarily originate from highway and freeway sensors, not reflecting urban-level traffic dynamics, making it difficult to build accurate incident detection systems. We address these challenges by simulating fine-grained traffic using microscopic traffic simulation built on real-world coarse datasets.

B. Limitations of Existing Incident Detection and Localization Methods

Work	Region	Dataset	DR	FAR	MTTD
[5]	Highway	Macroscopic	88.09 %	2.80 %	26.80 sec
[19]	Highway	Macroscopic	99.33 %	6.50 %	NA
[7]	Urban	Microscopic	86.40 %	8.69 %	61 sec
[20]	Urban	Macroscopic	86.6 %	5.12 %	NA
[21]	Both	Macroscopic	80 %	4.68 %	450 sec
[22]	Highway	Macroscopic	74 %	7.6 %	300 sec
Ours	**Both**	**Microscopic**	**98 %**	**6.26 %**	**197.4 sec**

TABLE I: Summary of incident detection works and their observed metrics. Given our interest in urban regions, [20] has shown the best detection and false alarm rates.

Various incident detection algorithms and their metrics like Detection Rate (DR), False Alarm Rate (FAR), Mean time to detect (MTTD), region (type of road), and type of data (microscopic and macroscopic) have been summarized in Table I. [5] used multiple highway cameras to detect incidents via spatial trajectory anomalies but did not address complex scenarios like ramps or lane closures. [19], [23] used the XGBoost algorithm for highway incident detection with [23] also calculating incident severity. However, they make predictions every 5 minutes, introducing increased incident detection time. In urban settings, [6] and [7] detected incidents using comparative and pattern-matching approaches with thresholds but failed to work well in dynamic traffic conditions, and they also require the installation of additional infrastructure to enable communication. Alternatively, [20] utilized a deep learning approach using traffic volume data from inductive loop detectors to detect incidents. However, its reliance on an adjacency matrix representing a sensor network and using macroscopic data raises scalability and efficiency concerns. Also, similar to [23], they predict incidents at 5-minute intervals, leading to delayed incident detection.

Incident detection algorithms reliant on data from all sensors during inference face efficacy challenges as some sensors may become non-functional over time. This was shown in the report [24], which highlighted that about 25% of New York's traffic sensors were nonfunctional during the survey. This has not been a focus area in previous studies, making it a crucial problem to be addressed.

III. Microscopic Traffic Dataset Generation

Most real-world traffic flow information is macroscopic, but we need microscopic data to detect incidents accurately in urban environments. We can obtain microscopic data through simulators such as SUMO [25], VSIM [21], and AIMSUN [26]. We do this in three parts: (i) Microscopic traffic flow simulation from macroscopic data, (ii) Traffic incident simulation, and (iii) Dataset generation.

A. Microscopic Traffic Flow Simulation from Macroscopic Data

It's essential to model macroscopic data such as publicly available vehicle counts to create realistic traffic simulations, as simulators don't have this capability inherently. The city of Tempe provides vehicle count data aggregated and reported every 15 minutes for multiple days. We use a 24-hour period of data as shown in Fig. 1 and generate microscopic traffic information that can produce vehicle counts for every second, allowing simulators to use this data to simulate the traffic. We start by computing the average vehicle counts across all roads of interest at every time step in an urban region.

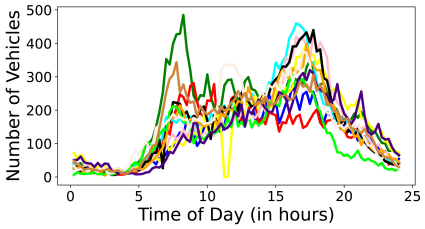

Fig. 1: The plot of the vehicle counts for a 24-hour period from the Department of Transportation of Tempe for the 12 roads between the placed sensors of interest from the selected Tempe region shown in Fig. 3.

We then apply Fast Fourier Transforms (FFT) [27] to the averaged vehicle count data points as shown in Fig. 2 and obtain the top two frequencies to build a non-linear equation that can approximately model the average traffic behavior over time, represented by the Equation 1.

$$f(t) = A_1 \sin(B_1 t + C_1) + A_2 \sin(B_2 t + C_2) + D + \alpha \quad (1)$$

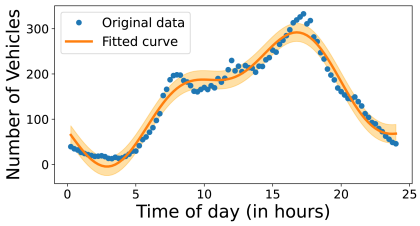

Fig. 2: Representation of averaged ground-truth vehicle counts and generated traffic flow model. To ensure variance in generated vehicle counts, we add a small deviation $alpha$.

To determine the parameters that best represent the original vehicle counts, we use the Levenberg-Marquardt algorithm (Equation 2) to tune the parameters, which results in minimizing the difference between the original vehicle counts and traffic flow model predictions.

$$\delta = \frac{J^T[y - f(t)]}{(J^T \cdot J + \lambda I)} \tag{2}$$

In this equation, λ represents the damping factor ($= 0.01$); δ represents the amount by which the parameters are updated in each step; J is the Jacobian matrix of partial derivative of the Equation 1 with respect to its parameters; $f(t)$ represents the vehicle count that we obtain from Equation 1.

B. Traffic Incident Simulation

Traffic incidents are simulated by halting vehicle(s). Depending on the likelihood of incident occurrence per vehicle, we first determine if we must insert an incident. If we have to insert an incident, we select one of the two incident types, halted vehicle and multi-vehicle crashes, for a duration also picked randomly based on the probability of the incident's severity depending on the two types of incidents. Once an incident is inserted, the radius of impact of the incident is calculated based on the severity of the incident. Inside the radius of impact, the vehicles are slowed down to emulate real-world crash behavior. It is challenging to categorize incidents as there is no direct access

C. Dataset Generation

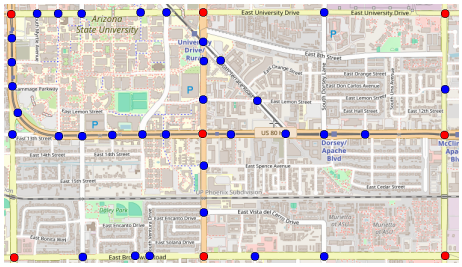

Fig. 3: Shows A Tempe, AZ region selected as the test area for our implementations. All the plotted points indicate the locations where cameras can be deployed for simulation. However, the deployed locations are highlighted in red to make the deployment of cameras similar to the real world.

Fig. 3 shows all the intersections at which traffic lights and, therefore, cameras can be placed. The red dots indicate the locations where the sensors are placed to capture simulation data, leading to an inherent sparsity in data capture. The simulation process is executed for multiple days, depending on the simulation configuration. We use an API service called Traci, provided by SUMO, to extract all the available features like vehicle counts, occupancy, vehicle speed, time of the day, and vehicle identifiers within a range of sensor locations similar to cameras for every second and consolidate them into a tabular format, generating huge raw microscopic traffic flow and incident dataset.

IV. Traffic Incident Detection, Localization and Severity Estimation

The captured raw dataset has (i) low variance as data is captured second, and traffic does not change significantly in such short intervals, leading to repeated data, (ii) frequent zero values, which are important from a data perspective but difficult to use from a deep learning perspective, like traffic counts, which makes sense for data, but acts as a sparse value for deep learning approaches and (iii) missing critical features such as vehicle travel time, limiting its effectiveness in training deep learning models. We consider data pre-processing approaches to overcome these challenges.

A. Feature Extraction from Raw Data

As travel time between intersections is an essential metric for incident detection, we used vehicle re-identification [28] to compute the travel time between all possible combinations of two contiguous intersections based on the sensor placements. Incorporating these travel times, junction mean speed, vehicle count, and vehicle occupancy into our dataset resulted in a feature-rich data source, significantly improving the dataset's utility and addressing the raw dataset's challenges.

Due to the presence of outlier data points, for example, when vehicles make unscheduled stops, we apply rolling window averages to reduce their impact. This technique involves averaging historical and current data, which allows us to smooth out anomalies in the dataset. If the current duration is labeled as an incident in the raw data, we label the rolling window average data points as incidents.

B. Model Selection

Despite these pre-processing efforts, we still observe missing data due to vehicles bypassing major intersections through interior roads and not getting re-identified. However, it still represents valuable information on traffic behavior. So, it is crucial to consider deep learning approaches that can better handle missing data.

Self-attention-based transformer models have worked exceptionally well to understand long-range sequences. TabNet [18] is an architecture designed for interpretable learning from tabular data. For training, the data is processed by the TabNet encoder, which uses a decision-making decoder to classify the results. Each TabNet encoder block comprises an attentive transformer block, a learnable mask, and a feature transformer. The learnable mask performs a soft selection of salient features, which are processed by the feature transformer, and the attentive transformer learns the

importance of each feature during training. Multiple layers of these encoder blocks form the TabNet Encoder. The authors claim that this instance-wise feature engineering and learning allows for a better performance than Decision Tree-based models like XGBoost, making it a significant factor for us to consider this as our model architecture.

C. IncidentNet's Model Architecture Design

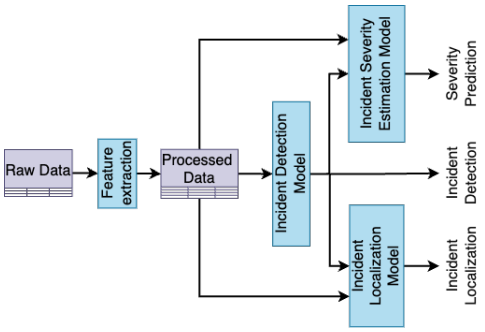

Fig. 4: The block diagram depicts IncidentNet's architecture. The raw data from the simulator is transformed into processed data. For training, all data points are used for the incident detection model, and data points with positive incident labels are used for incident localization and severity estimation models. During the prediction phase, localization and severity estimation models depend on the incident detection model's prediction.

Our incident detection architecture employs a stacked ensemble of three models dedicated to incident detection, localization, and severity estimation tasks. All three models are trained individually, with varying input data. For the incident detection model, the complete pre-processed dataset is provided as input and trained to predict if an incident has occurred in the complete selected urban region. We cannot predict the incident's class as they are not occurring in the sensor's field of view. For localization and severity estimation models, the data points with positive ground truth incident labels are considered for training. The localization model predicts the roads on which the incident occurred, and the severity estimation classifies if an incident is severe. Our ensemble model can localize and estimate severity only due to the microscopic dataset we generated.

A unique aspect of our architecture is its robustness in accommodating sparse sensor settings, a common challenge in real-world traffic monitoring scenarios. Unlike existing incident detection methods, our models are evaluated under various levels of sensor sparsity to assess the performance of each task under various degrees of sparsity.

V. Experimental Setup

A. Simulation Setup for Dataset Generation

We generate simulation files using the OSM Web Wizard for a continuous period of 30 days to simulate traffic flow for the selected Tempe region and generate the microscopic data using the process described in our approach.

B. Pre-processing Raw Dataset

We test with three variations: 300, 600, and 900 seconds to pre-process the raw data and select the rolling window size. We train our model using pre-processed data aggregated using different window sizes and observe F1 scores of 93.12%, 96%, and 96%, respectively. Given that more data increases the computational requirement, we choose 600 seconds as our window size choice, as the F1 score for 600 seconds and 900 seconds are similar.

C. Model Training and Evaluation Considerations

We used TabNet to evaluate the Tempe dataset. The model was trained on NVIDIA RTX 5000 GPU, and for TabNet, the hyperparameters used are mentioned in Table II.

Hyper-parameters	Value
Prediction Layer Dimension	64
Attention Embedding Dimension	64
Optimizer Momentum	0.3
Optimizer	Adam
Learning Rate	0.02
Epochs	80
Loss Function	Cross Entropy

TABLE II: Model training hyper-parameters for TabNet.

Table III shows the different metrics we use to evaluate the performance of our model. We used the three standard metrics, Detection Rate (DR), Mean Time to Detect (MTTD), and False Alarm Rate (FAR), to evaluate the performance of the incident detection algorithm.

Metrics	Definition
DR (Detection Rate)	$\frac{TP}{TP+FN}$
FAR (False Alarm Rate)	$\frac{FP}{FP+TN}$
Accuracy	$\frac{TP+TN}{TP+TN+FP+FN}$
Precision	$\frac{TP}{TP+FP}$
Recall	$\frac{TP}{TP+FN}$
F1 Score	$2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision}+\text{Recall}}$
Specificity	$\frac{TN}{TN+FP}$

TABLE III: Traffic Incident Detection Metrics and their definitions based on confusion matrix, where TP = true positives, TN = true negatives, FP = false positives, FN = false negatives.

VI. Results

A. Our Microscopic Data Matches Very Well with Real-World Macroscopic Data

To validate that our simulated data accurately reflects real-world conditions in the Tempe region, we aggregated the microscopic simulation data to match the time frame of Tempe's macroscopic real-world traffic count data. This produces a distribution similar to the original data represented in Fig. 2. To assess the similarity, we used the Kolmogorov-Smirnov (KS) test [17], which evaluates the similarity between two distributions by calculating two metrics: KS statistic and p-value. The KS statistic measures the maximum discrepancy

Algorithm	DR	MTTD	FAR	Accuracy	Precision	Recall	F1 Score	AUC-ROC	Specificity
Our approach (XGBoost)	96 %	94 secs	11.03 %	92.13 %	93.76 %	90.49 %	87.43 %	91.46 %	95.7 %
Our Approach (TabNet)	98 %	197.44 secs	6.26 %	93.85 %	94.27 %	91.17 %	92.70 %	93.51 %	95.95 %
Zhu et al. [20]	51 %	471 secs	35.42 %	60.06 %	40 %	50.45 %	44.62 %	51 %	64.57 %

TABLE IV: The table compares the previous state-of-the-art, XGBoost and our approach for the microscopic dataset generated for urban traffic scenarios. Our approach performed exceptionally well when compared to the previous state-of-the-art. The other outcome we observed was that XGBoost performed better than the state-of-the-art, proving the importance of microscopic datasets. Our method predicted incidents every 30 seconds instead of every 5-minute interval, as in [20].

between the distribution functions of datasets. The p-value measures the probability of low discrepancy between the two datasets. The null hypothesis is true when both distributions are similar. We reject the null hypothesis if the p-value is below the accepted significance of 0.05.

The Tempe Department of Transportation provides the vehicle count data for just four days, and the days on which they were collected are randomly presented. Of the 30-day simulated data, we selected four days randomly for validation. We observed that, though there is variation in the KS Statistic and the p-value, all of them pass the cut-off according to the algorithm as shown in Fig. 5, indicating similarity between the simulated and original data.

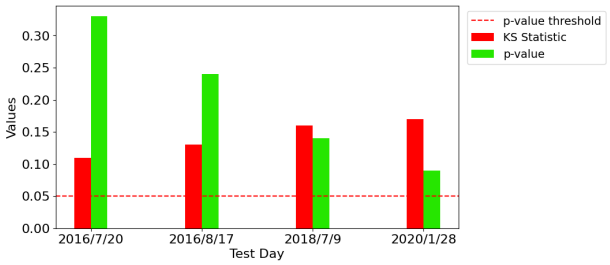

Fig. 5: The KS Statistic and the p-value obtained from the KS test for the four days of data made available by Tempe are shown. The p-value threshold is indicated as the red line.

B. IncidentNet is Better at Detecting Incidents in Urban Regions Compared to the Previous Works

As highlighted before, a fast and accurate traffic incident detection algorithm can reduce the impact of incidents economically and environmentally and, mainly, reduce fatality rates. Their impact can be evaluated using metrics such as DR, FAR, and MTTD, which are defined in V-C. We evaluated our work against the state-of-the-art by training a model using the architecture provided by [20], which we implemented to the best of our understanding as the official model implementation was not available, and the XGBoost model architecture, using our microscopic dataset. As XGBoost has proven to work well on tabular data [19], [23] due to its efficient selection of global features with high information value [29], to assess the impact of the microscopic dataset, we also evaluate with XGBoost as the model consideration in our approach.

We evaluated all the models on a newly generated evaluation dataset for the same region, consolidated in Table IV. We observe that XGBoost's performance improves drastically compared to the model's performance on microscopic data, showing the importance of considering microscopic datasets for traffic incident detection. Our TabNet approach is more accurate than XGBoost, with a DR of 98% and FAR of 6.26%. The downside we observed is that the MTTD is 197.44s, almost 100s higher than XGBoost. However, this is offset by the much lower FAR, which indicates that our model has the ability to report incidents more accurately while remaining fast enough to be within the 7-minute mark, as defined in [2]. The inference time of the TabNet model on the Intel Xeon W-2555 CPU was 5 ms, providing timely insights and supporting real-time decision-making.

C. IncidentNet Works Even In Sparse Sensing Condition

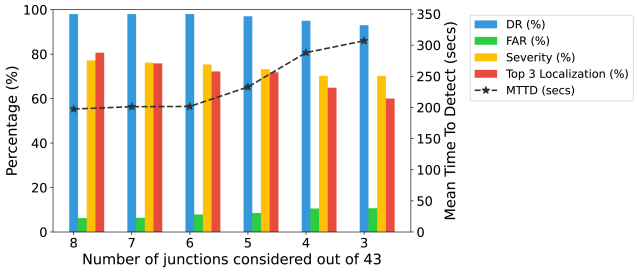

Fig. 6: Our approaches' performance with consideration for different sparsity levels. Notice that the incident detection rate is still high for sparsity, as high as 93%. Tabnet performs better in incident detection with a low false alarm rate.

Sensing hardware is fallible and degrades over time, thus it is reasonable to assume that not all cameras will be working at all times. It is vital for a model to be able to work even in such conditions. So, we test our model's performance with increasing levels of sparsity. We start with a realistic sensor deployment at 8 of the 43 possible intersections and scale down to just 3 intersections.

In Fig. 6, we observe with increased sensor sparsity that our model still retains the capability to detect if an incident occurs, but the accuracy of localization and severity predictions is reduced. Interestingly, with only six sensors, the MTTD does not increase much. However, the MTTD increases more drastically with fewer than 6 sensors. The FAR also increases with an increase in sparsity. Although we observe this increase, we show our model is still capable of predicting metrics, even during infrastructure anomalies.

D. IncidentNet can Detect Incidents on Highways

Given that our model works in urban regions, we test if our approach works in a highway scenario. We used an 8-mile highway stretch, inserted the sensors on every available

Algorithm	DR (%)	FAR (%)	MTTD (secs)
Our Approach (XGBoost)	98	6.02	45
Our Approach (TabNet)	99	4.17	70

TABLE V: Highway performance of IncidentNet our approach compared against the XGBoost model on our microscopic dataset. Results demonstrate that the performance of XGBoost improves because of the microscopic dataset, and IncidentNet performs better than XGBoost.

ramp, and simulated the microscopic dataset. We trained and evaluated using XGBoost and our model. The metrics obtained are shown in Table V. XGBoost model performed better than the previous works shown in Table I. Our model performed better than XGBoost in terms of DR and FAR, with a very minimal increase in the MTTD, proving that our approach works in both urban regions and highways.

VII. Conclusion

In this paper, we have shown that IncidentNet can successfully detect traffic incidents with a high detection rate in urban roads using microscopic sensor data. In particular, the results confirm that using just 3 instrumented intersections of the 43 possible IncidentNet can accurately detect, localize, and classify incidents in a large area, marking a significant advancement in traffic management technologies. Building upon this supervised model, a promising next step is implementing a semi-supervised version of IncidentNet. This would allow the model to continually improve and handle recurring congestion when deployed in real-world settings. This work also highlights the importance of sensor placement in sparse sensing scenarios, highlighting the need for an algorithm to efficiently place sensors while maximizing the incident detection rate in sparse sensing. Further investigation could extend to categorizing incidents into more classes and enhancing localization accuracy, possibly including rough estimation of distances of incidents from the intersections. Additionally, extending to other regions is part of the future scope of this work.

References

[1] Lawrence Blincoe et al. The economic and societal impact of motor vehicle crashes, 2019. Technical report, 2022.

[2] James P Byrne et al. Association between emergency medical service response time and motor vehicle crash mortality in the united states. *JAMA Surgery*, 154:286–293, 2019.

[3] PB Farradyne Inc., editor. *Traffic incident management handbook*. Nov. 2000.

[4] Unaiza Alvi et al. A Comprehensive Study on IoT Based Accident Detection Systems for Smart Vehicles. *IEEE Access*, 8:122480–122497, 2020.

[5] Haoxiang Liang et al. Traffic incident detection based on a global trajectory spatiotemporal map. *Complex & Intelligent Systems*, 8(2):1389–1408, Apr. 2022.

[6] Wooyeon Yu, SeJoon Park, David S. Kim, and Sung-Seok Ko. Arterial Road Incident Detection Based on Time-Moving Average Method in Bluetooth-Based Wireless Vehicle Reidentification System. *Journal of Transportation Engineering*, 141(3):04014084, Mar. 2015.

[7] Xiaolin Han et al. Traffic incident detection: A trajectory-based approach. In *2020 IEEE 36th international conference on data engineering (ICDE)*, pages 1866–1869. IEEE, 2020.

[8] California Department of Transportation. Performance measurement system (pems).

[9] Alexander Skabardonis et al. I-880 field experiment: Data-base development and incident delay estimation procedures. *Transportation Research Record*, 1554(1):204–212, 1996.

[10] Yaguang Li et al. Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting, Feb. 2018. arXiv:1707.01926 [cs, stat].

[11] Zhiyu Ren et al. Graph autoencoder with mirror temporal convolutional networks for traffic anomaly detection. *Scientific reports*, 14(1):1247, 2024.

[12] Fadi Al-Turjman and Joel Poncha Lemayian. Intelligence, security, and vehicular sensor networks in internet of things (iot)-enabled smart-cities: An overview. *Computers and Electrical Engineering*, 87:106776, 2020.

[13] Bozheng Pang et al. A Study on the Impact of the Number of Devices on Communication Interference in Bluetooth Low Energy. In *2020 XXIX International Scientific Conference Electronics (ET)*, Sept. 2020.

[14] Yanwei Yu et al. Citywide traffic volume inference with surveillance camera records. *IEEE Transactions on Big Data*, 7(6):900–912, 2021.

[15] Ankit Shah et al. Cadp: A novel dataset for cctv traffic camera based accident analysis. *arXiv preprint arXiv:1809.05782*, 2018. First three authors share the first authorship.

[16] Traffic Counts and Maps | City of Tempe, AZ.

[17] Kolmogorov–Smirnov Test. In *The Concise Encyclopedia of Statistics*, pages 283–287. Springer New York, New York, NY, 2008.

[18] Sercan Ömer Arik and Tomas Pfister. TabNet: Attentive Interpretable Tabular Learning. *CoRR*, abs/1908.07442, 2019. arXiv: 1908.07442.

[19] Junyu Chen et al. More robust and better: Automatic traffic incident detection based on XGBoost. In *Advances in Traffic Transportation and Civil Architecture*, pages 267–274. CRC Press, 2023.

[20] Lin Zhu et al. A deep learning approach for traffic incident detection in urban networks. In *2018 21st international conference on intelligent transportation systems (ITSC)*, pages 1011–1016. IEEE, 2018.

[21] Huan Yang et al. Traffic Incident Generation And Supervised Learning Based Detection Via A Microscopic Simulation Platform. In *2023 IEEE International Conference on Cybernetics and Intelligent Systems (CIS)*, pages 146–151, June 2023. ISSN: 2326-8239.

[22] Irem Atilgan et al. Towards accurate traffic accident detection: Developing deep learning strategies with distant past, recent past, and adjacency features. In *2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC)*, pages 6120–6125. IEEE, 2023.

[23] Miao Xu, Hongfei Liu, and Hongbo Yang. Ensemble learning based approach for traffic incident detection and multi-category classification. *Engineering Applications of Artificial Intelligence*, 132:107933, 2024.

[24] Edward W Bikowitz and Scott P Ross. Evaluation and Improvement of Inductive Loop. *Transportation Research Record*.

[25] Pablo Alvarez Lopez et al. Microscopic traffic simulation using sumo. In *The 21st IEEE International Conference on Intelligent Transportation Systems*. IEEE, 2018.

[26] Aimsun. *Aimsun Next 23 User's Manual*. Barcelona, Spain, aimsun next 23.0.0 edition, Accessed on: Month, Day, Year 2023. [Online].

[27] James W Cooley and John W Tukey. An algorithm for the machine calculation of complex Fourier series. *Mathematics of computation*, 19(90):297–301, 1965.

[28] Xiaohui Huang et al. Machine-learning-based real-time multi-camera vehicle tracking and travel-time estimation. *Journal of imaging*, 8(4):101, 2022.

[29] Krzysztof Grabczewski and Norbert Jankowski. Feature selection with decision tree criterion. In *Fifth International Conference on Hybrid Intelligent Systems (HIS'05)*, pages 6–pp. IEEE, 2005.