Deep Learning Based IoT System for Real-time Traffic Risk Notifications

Sahidul Islam*, Seth Klupka †, Ramin Mohammadi†, Yu-Fang Jin†, Mimi Xie*
*Department of Computer Science, The University of Texas at San Antonio
†Department of Electrical and Computer Engineering, The University of Texas at San Antonio

2024 25th International Symposium on Quality Electronic Design (ISQED) | 979-8-3503-0927-0/24/$31.00 ©2024 IEEE | DOI: 10.1109/ISQED60706.2024.10528772

***Abstract*—Enhancing public safety by reducing traffic crashes due to human error is critical. A key strategy involves ensuring that drivers remain alert through the implementation of early safety notifications. Nonetheless, forecasting the risk of traffic crashes poses a complex challenge due to numerous factors including road conditions, traffic dynamics and weather patterns. In response to this problem, this study endeavors to develop an end-to-end traffic risk notification system, integrating IoT system with Deep Learning solution. This multi-node collaborative IoT system offers more pragmatic risk analysis by incorporating both static and dynamic factors which can contributes to a traffic crash. At the device level, strategically placed sensors capture dynamic data which is then transmitted to the decision node. Leveraging a deep learning-based model, the decision node processes both static and dynamic information to predict a crash severity risks, and the outcomes are seamlessly communicated to the display node for timely notification to drivers, fostering a safer and more responsive driving environment. The CNN model is developed via extensive training with accident history data in the state of Texas. The evaluation of our deep learning model was performed using key metrics, including accuracy, recall, precision, and F1 score. Finally, multi-node collaborative IoT system is evaluated based on prediction and communication latency.**

***Index Terms*—Deep Neural Network, Traffic Risk Notification**

I. Introduction

In the United States, a staggering 90% of the annual 36,000 traffic-related fatalities are attributed to human error [1], exemplified by the tragic toll of 4,480 lives lost in car crashes in Texas alone [2]. On a global scale, the World Health Organization (WHO) notes that traffic accidents cost about 3% of the world's GDP, with approximately 1.35 million people losing their lives each year in road traffic crashes [1]. Human error stands out as a significant contributing factor, encompassing behaviors like distracted driving and impaired driving, speeding due to alcohol or drugs, and failure to obey traffic signals.

Predicting traffic crashes is inherently complex, involving a multitude of factors beyond human error [3]–[5]. Road dynamics, infrastructure, passenger behavior, driver actions, pedestrian interactions, and varied weather and traffic conditions all contribute to the intricacies of potential crashes. While efforts such as public awareness campaigns, driver education, and law enforcement have been implemented globally, the advancement of technology, particularly artificial intelligence (AI), necessitates more potent solutions to effectively reduce human error. Alerting individuals to critical situations emerges as a promising approach, yet developing a notification-based alert system presents challenges. Determining the optimal timing for alerts, interpreting risky situations, and creating a sustainable, low-maintenance, and feasible system pose significant difficulties. The pursuit of proactive road safety solutions demands innovative methodologies to integrate advanced AI technologies effectively. The development of deep learning methods has shown promising results in addressing various traffic-related problems. Deep learning techniques, particularly convolutional neural networks (CNNs) [6]–[9] and recurrent neural networks (RNNs), have been applied to various aspects of traffic management and analysis.

Fig. 1: IoT-based crash notification system is strategically implemented in hot-spot region, identified by thorough analysis of historical data. This system dynamically assesses and presents real-time crash risk, incorporating both static and dynamic information. High-alert notifications are triggered in instances where (a) unexpected obstacles significantly elevate the risk of traffic crashes and (b) adverse weather conditions or poor road conditions pose an increased threat to traffic safety.

In this project, we have developed an innovative deep learning based multi-node collaborative crash notification system, offering a comprehensive end-to-end solution to address the complex challenges associated with predicting and mitigating traffic accidents. The system seamlessly integrates offline cloud training, edge based real-time inference, and device level sensing/receiving/display of data to facilitate real-time crash risk predictions and alerts.

The first step, we collect and analyze the traffic accident data to train a Conolutional Neural Network (CNN) model. This data is classified into two main groups: static and dynamic. Static data includes road dynamics and the history of traffic accidents, while dynamic data comprises variables such as weather conditions, human behavior, traffic congestion, and average traffic speed. Our training dataset spans accident history in the state of Texas. The historical data is stored in the cloud, serving as the foundation for training the CNN model to predict crash risks. Once trained, the model is deployed on the edge device, enabling real-time predictions.

Recognizing the inherent uncertainty in determining if an accident will occur, our focus shifts to predicting the risk of a crash. This involves identifying hot-spots by analyzing crash severity history and assessing dynamic information. Leveraging historical accident data and rigorous statistical analysis, we pinpointed areas with the highest frequency of traffic crashes, known as hot-spots.

To enhance real-time data collection in these identified hot-spots, a collaborative multi-node IoT system is developed. This nodes consist of strategically placed sensors that capture dynamic information crucial for understanding changing conditions over time and varying situations. The collected data is transmitted to the nearest edge device, where the trained CNN model is deployed. This model performs crash severity-based risk predictions. The predicted results are promptly sent back to another display device, triggering notifications that display the current crash risk.

Here, all computations are performed in real-time as they involve in-situ processing of data and decision-making on edge servers, thereby avoiding communication with distant clouds. Moreover, crash risk notifications are transmitted to LED displays in hot-spot regions before drivers enter the area in real time. Lastly, the designed data sensing/pre-processing and CNN-based prediction models achieve real-time performance. These notifications serve as crucial alerts for drivers approaching these high-risk areas, enhancing situational awareness and fostering a proactive approach to accident prevention. This multi-node collaborative system represents an innovative and effective solution at the intersection of advanced computing technologies and real-world road safety challenges.

The rest of this paper is organized as follows. Section 2 describes the research related to deep learning techniques used for traffic crash prediction. Section 3 describes the proposed end-to-end framework. Section 4 describes the experimental evaluation and results. Lastly, Section 5 provides a conclusion.

II. Related work

A. Deep Learning for traffic data analysis

Deep learning stands out as one of the most popular subsets of machine learning, enabling researchers to delve into more complex, higher-dimensional data through automatic feature extraction. Within deep learning, different models and architectures are designed for different domains such as computer vision, language processing, or audio analysis. Despite the absence of specific models designed for traffic data analysis, researchers often leverage or adapt existing models for training and predicting traffic-related information. Numerous studies have explored the application of Deep Neural Networks (DNN) for tasks like traffic flow and pollution prediction.

In the context of traffic prediction, the work by Yisheng [10] introduces an auto-encoder-based model. Another notable contribution is from Yao [11], where Attention-based Graph Convolutional Networks are proposed for monitoring traffic pollution. Some studies focus on predicting the congestion evolution within large-scale transportation networks [12], while others [13] employ deep reinforcement learning to identify optimal traffic signal timing policies.

B. Machine Learning for Crash Severity analysis

Numerous machine learning methods are extensively employed for assessing traffic crash severity. In the study by [14], a Supervised Machine Learning (ML) algorithm and random forest decision tree-based algorithms are proposed and compared for predicting the severity level and future crashes based on road crash elements. Similarly, [15] addresses comparable issues by analyzing data from Louisiana states using a Convolutional Neural Network (CNN)-based deep learning algorithm. The model utilizes a customized loss function to optimize precision and recall directly. Additionally, crash analysis [16] is conducted on accident data history in Dhaka spanning from 2007 to 2011.

C. Real time traffic accident prediction

The real-time prediction of traffic accidents involves harnessing data from diverse sources to anticipate and notify authorities or drivers about the potential occurrence of accidents on the road. In a study conducted by Lee, an accident prediction model was developed using two machine learning methods: artificial neural networks (ANN) and k-nearest neighbor. The findings indicated that the ANN outperformed the k-nearest neighbor, delivering accurate predictions with less than 30Achu employed geospatial technology to investigate the temporal and spatial behaviors of traffic accidents. The study utilized various methods, including kernel density functions, Moran's-I, and Getis-Ord Gi hotspot analysis, to analyze the spatiotemporal patterns of traffic accidents [17]. Likewise, Park gathered extensive traffic accident data for highways in Seoul and constructed a prediction workflow based on k-means cluster analysis and logistic regression [18]. More recently, Chen [19] utilized human mobility data in Japan and developed a Stack Denoise Autoencoder to infer real-time traffic risk.

D. IoT Based Crash Notification System

Internet of Things (IoT)-based Crash Notification System is a technological solution designed to enhance road safety and emergency response by leveraging IoT devices and connectivity. The study in [20] introduces a rapid reporting system designed to promptly notify about road crashes. Utilizing various sensors, this system gathers vehicle data and signals abnormal driving situations, facilitating swift responses from rescue teams, insurance personnel, or relatives for easy navigation to the incident location. In a parallel investigation, a research initiative proposed an IoT-based Post Crash Assistance system with the aim of eliminating dependency on smartphones [21]. This system aims to provide support after a crash, streamlining the process for reporting and assistance without relying on mobile devices. Furthermore, in the study referenced as [22], researchers developed an automatic accident detection and

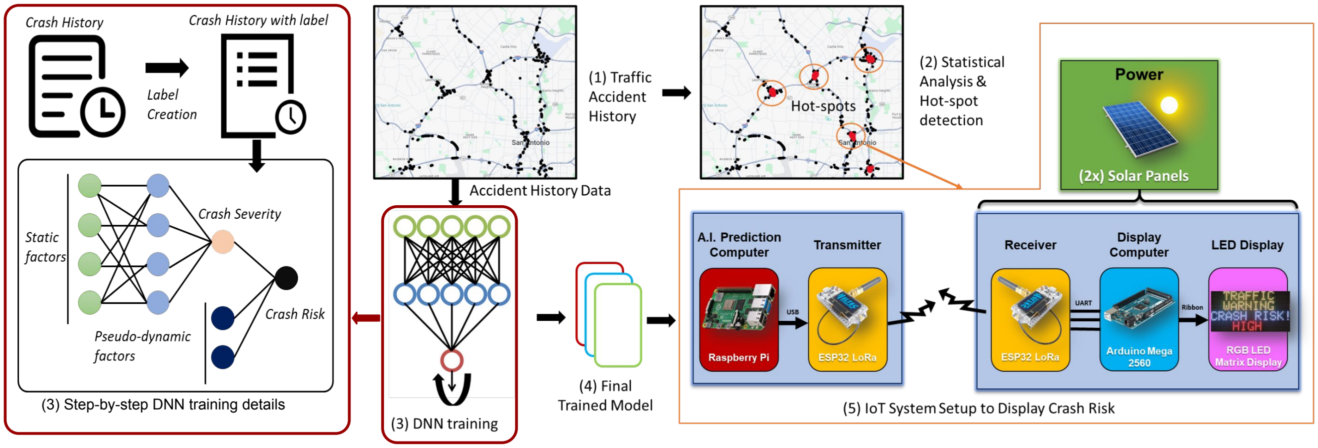

Fig. 2: System Overview

notification system within an IoT-based vehicular environment. In a different context, another research effort, detailed in [23], proposed a smart helmet designed to serve as both a means and apparatus for detecting and reporting accidents. This innovative approach integrates wearable technology to contribute to improved safety measures on the road.

Nevertheless, the existing IoT systems primarily concentrate on post-crash notification mechanisms. In contrast, our work addresses this gap by developing a comprehensive end-to-end crash notification system that proactively alerts the driver before an impending collision occurs.

III. Methodology

The end-to-end traffic notification system comprises two main components. Firstly, the development of a deep learning-based model capable of predicting crash risk constitutes the initial part. Secondly, an multi-node IoT-based network is established to alert drivers based on predictions generated by the AI model. This sustainable and low-maintenance IoT system is then deployed in each hot-spot region.

A. Analysis of Accident History

The foundation of our IoT-based notification system lies in data. To conduct a thorough analysis, we gathered five years of accident history data from the state of Texas. Our analysis comprises two primary components: predictive analysis and statistical analysis. In predictive analysis, we developed a deep learning-based model, as detailed in Section III-B, to forecast crash patterns. Following that, we conducted statistical analysis to pinpoint locations deemed as frequent crash hot-spots. These hot-spots serve as areas where the edge-device collaborative IoT system is deployed to provide real-time crash risk notifications.

B. CNN training and deployment

Our deep learning based crash prediction model went to rigorous training with crash history data. The training process has four phase. (1) Data Pre-processing (2) Label Creation. (3) Training (4) Psedo-dynamic factor incorporation.

1) Data Pre-processing: In this step, our aim is to improve the quality of the data, reduce noise, and prepare it for effective utilization by deep learning algorithms. To ensure data quality and enable meaningful pattern recognition for the CNN model, we quantified the string data using label encoding. In label encoding, string values are converted to several categories where each unique category is assigned a numerical label. The labels are typically assigned in ascending order starting from 0 or 1. This encoding allows algorithms to operate on categorical data by representing them as numerical values, which can be more easily processed by deep learning models. Furthermore, we pruned unnecessary features from the dataset, diminishing the total count from 82 to 62. Lastly, any null or empty values were substituted with appropriate numerical equivalents, commonly represented as zero.

2) Label Creation: In supervised learning, a "label" refers to the output or target variable that the model is trained to predict. In a dataset used for supervised learning, each example consists of input features and their corresponding labels. The goal of the model is to learn a mapping from the input features to the correct output labels. In the first step, we created a label called "crash severity." This severity level has six outcomes (*no-injury, fatal, serious, minor, possible, unknown*). The outcome is calculated based on factors such as injury count, damage intensity, airbag deployment status, and so on. In this process, each accident is assigned a crash severity level with one of these six outcomes.

3) Training: Next, a Convolutional Neural Network (CNN) model is trained to predict crash severity using crash history data. The objective is to learn forecasting the crash severity level based on static parameters. Following that, the CNN model is expanded with an additional final layer. The purpose of this layer is to integrate the crash severity level with dynamic factors to predict the final risk level.

4) Pseudo-dynamic factor: As dynamic factors influencing crash risks change over time and are only accessible in a real-time setup, we incorporate pseudo-dynamic factors and rules in the concluding segment of the trained model to forecast the risk level. This process is illustrated in Figure 2-(3).

After completing the training process, the finalized predic-

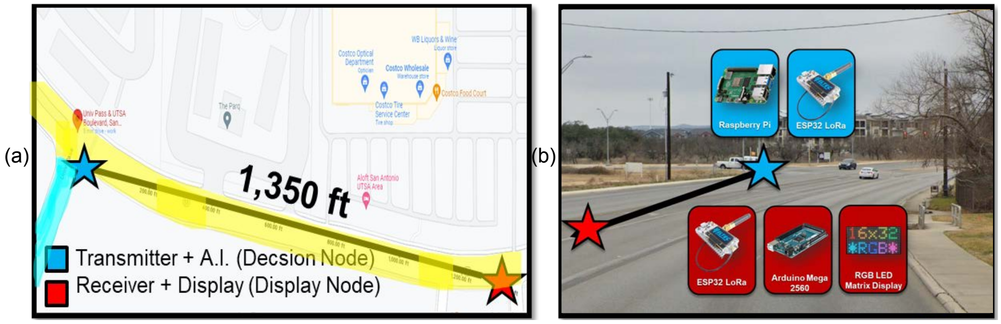

Fig. 3: Multi-node IoT System Setup to Display Crash Notification

TABLE I: Description of the proposed Convolutional Neural Network (CNN) based risk prediction model

Model Detail	Layer (type)	Weight Shape	Output Shape	Param #
CNN for	Conv2d-1	50x1x3x1	1x50x31x1	200
Risk Prediction	ReLU-2	50x1x3x1	1x50x31x1	0
	BatchNorm2d-3	50x1x3x1	1x50x31x1	100
Input Shape: [1*62]	Conv2d-4	50x50x4x1	1x50x14x1	10,050
Accuracy: 99.49%	ReLU-5	50x50x4x1	1x50x14x1	0
Model Size: 155 KB	BatchNorm2d-6	50x50x4x1	1x50x14x1	100
Total params: 38,751	Linear-7	40x700	1x40	28,040
Latency: 7ms	Linear-8	6x40	1x6	246
	Linear-9	3x6	1x3	21

TABLE II: Evaluation Metrics

Accuracy	Precision	Recal	F1_score	Fb_score	Latency (P/C)
99.49%	94.71%	86.29%	87.63%	87.63%	7ms / 5000ms

tion model is deployed on the edge device to facilitate real-time predictions, as detailed in the following Section III-C.

C. IoT System Setup

The safety notification IoT system comprises multiple nodes collaborating within each hot-spot region. This collaborative setup involves two primary nodes: the decision node and the display node.

Decision Node: In this setup, a trained CNN model is deployed for real-time risk prediction. The model runs on a Raspberry Pi and is connected to strategically deployed sensors. These sensors capture dynamic information such as changing weather conditions (e.g., rainy, icy, sunny), traffic conditions (e.g., average speed, traffic density), and date-time information (e.g., day of the week, hour of the day). This dynamic data is fed into the CNN model to perform accurate risk analysis, as discussed in Section III-B. In Figure 3, blue node represents the decision node.

Display Node: Within this node, a display system has been developed to showcase the risk level analyzed by the decision node. Typically, the display node is positioned ahead of the decision node with the aim of alerting drivers well in advance of any hazardous regions. As depicted in Figure 2-(5) and Figure 3, our multi-node IoT system includes this display node (red color), which is often situated approximately 2000 feet away from the decision node. Notably, the display node is powered by energy harvesting technology, ensuring sustainability and minimal maintenance for the system.

TABLE III: Descriptive Statistics of Crash Data

Year	Crash Count	Crash Severity Count					
		Unknown	Serious	Minor	Possible	Fatal	No-injury
2017	619174	36045	14663	60426	100953	3373	403714
2018	623221	41203	12438	55635	103840	3335	406770
2019	648019	39298	13273	55999	109167	3353	426929
2020	544697	36932	12415	47639	85399	3577	358735
2021	633043	40705	16191	62546	89632	4108	419861

IV. Experimental Evaluation

A. Experimental Platform

1) Training: Offline CNN model training is performed on Intel Xeon Gold 5218 machine at 2.30 GHz using an Nvidia Quadro RTX 6000 GPU with CUDA 11.6, pytorch 1.12.1.

2) IoT System Hardware: Our IoT System is evaluated with multiple hardware device. In the decision node, a Raspberry Pi 4 Model B is used to host CNN model. The device is connected to transmitter LoRa-32. In the display node, another LoRa-32 device act as a receiver which also connects an Arduino Mega 2560 board. This device process the information receives from the wireless receiver and display to the LED broad. We use 64x32 RGB LED Matrix Display that is connected to the audrino board via Adafruit RGB Matrix Shield. There are also Raspberry Pi Sense Hat sensors collecting information such as environmental conditions, barometric pressure, temperature, humidity. For gathering visual information we use Arducam 16MP Autofocus Camera. The display node is powered by an energy harvesting module Keithley 2280S power supply with a power regulator Bq25570 (3.3V) and an energy buffer (100µF capacitor).

3) Datasets: We clean and reformat a dataset by collecting the traffic crash history in the state of Texas from the year 2017 to 2021 [24] and show the summary of the dataset in Table III. In total, the dataset contains 3,068,154 number of crashes. And among these crashes we used 62 total features, where most important features are described in Table IV. Here, the features are mostly categorical data which is later quantified by the data pre-processing steps described in Section III-B1.

B. Experimental Setup

The CNN model is trained offline with above dataset. At first, data are prepossess by replacing null value and quantifying the string values. Next the model is trained with 10 epochs, 0.001 learning rate and batch size of 100. The 3 million training data is spited into 3 different sets such

TABLE IV: Important Features and Descripton

Type	Factor	Feature	Description
Static	Road Statistics	Adt_Curnt_Amt	Average daily traffic amount for a given road segment and year.
		Adt_Curnt_Year	Year identifier for the average daily traffic crashes.
		Adt_Adj_Curnt_Amt	Adjusted average daily traffic for the current year.
		Pct_Single_Trk_Adt	Percentage of average daily traffic that is single-unit truck.
		Pct_Combo_Trk_Adt	Percentage of average daily traffic that is combo truck.
		Trk_Aadt_Pct	Adjusted average daily traffic percent for trucks.
		Cnty_ID	County Information.
		City_ID	City Information.
		Latitude	Latitude.
		Longitude	Longitude.
	Road Conditon	Road_Type_Id	Type of the Road. Such as Highway, Lane count, etc.
		Road_Align_Id	Straight or Curvy level of the road.
		Crash_Speed_limit	Maximum Speed Limit on the road.
		Active_School_Zone_Fl	Whether the road is in active school zone.
		Toll_Road_Fl	Whether it is a toll road or not.
	After Effect	Thousand_Damage_Fl	If the damage cost thousand dollar or more.
		Sus_Serious_Injry_Cnt	How many people injured seriously.
		Non_Injry_Cnt	How many people is not injured.
		Tot_Injry_Cnt	Total Injury count.
		Death_Cnt	Total Death Count.
		Medical_Advisory_Fl	If Medical adivsory is taken.
Dynamic	TimeStamp	Day_of_Week	Indicates week (Saturday[0]-Friday[6]).
		Hour_of_Day	Indicates hour of the day (0-23.)
	Weather	Wthr_Cond_ID	Whether the weather is rainy,sunny or icy.
		Light_Cond_ID	Whether it is daylight, dawn, dusk or night.
		Surf_Cond_Id	Whether the surface is dry,wet,icy or muddy.
	Traffic	Av_Car_Speed	Avergage speed of the traffic.
		Car_Density	Density of traffic.

as training set, testing set and validation set. 80% of the total dataset are used for training and rest 20% are used for validation and testing with 10% each. Which means the sample size of training, validation, testing is 2,454,523, 306,812 and 306,816 respectively. We also make sure that the training, validation and testing data are splited evenly for each of the class to avoid data imbalance. It took a total of 30 minutes to complete the training process. The loss function we have used is cross entropy loss and the optimization algorithm we chose is called Adam optimizer [25].

C. Performance Evaluation

1) Evaluation Metrics: We evaluated our CNN model based on accuracy, precision, recall, F1 and F-beta score metrics. Precision metrics measures the accuracy of the positive predictions, indicating how many of the predicted positive instances are actually relevant. Precision is calculated as: Precision = TP / (TP + FP), where TP is true positives and FP is false positives. Recall measures the model's ability to capture all the relevant instances, indicating how many of the actual positive instances were correctly predicted. Recall is calculated as: Recall = TP / (TP + FN), where FN is false negatives. F1 score is the harmonic mean of precision and recall. F1 Score is calculated as: F1 Score = 2 * (Precision * Recall) / (Precision + Recall). F-beta score is a generalized form of the F1 score, allowing more importance to either precision or recall based on the value of beta.

2) Training Evaluation: Figure 4 displays the training/validation loss alongside the training/validation accuracy metrics. Both the training and validation accuracies are reported as 99.98% and 99.49%, respectively. It took 10 epochs to converge the model. In this training, a dataset comprising 2,454,523 crash history records was utilized for training purposes, while a 306,812 records were used for validation.

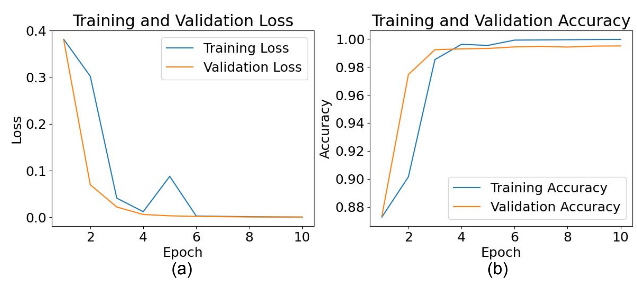

Fig. 4: Loss and Accuracy graph during Training and Validation

3) Model Evaluation: The structure of the trained model is illustrated in Table I. This convolutional neural network model features two convolutional (CONV) layers, each followed by a ReLU activation function and a Batch Normalization layer. Subsequently, three consecutive fully connected (FC) layers are positioned at the end of the model. In total, the model encompasses 38,751 parameters, which occupy a memory footprint of 155Kb when represented in 32-bit floating-point numbers. The input comprises a total of 62 features, and the weight and output shapes for each layer are detailed in Table I. As indicated in Table II, the final model attains a testing accuracy of 99.49%. Furthermore, the precision, recall, F1 score, and F-beta score are reported as 94.71%, 86.29%, 87.63%, and 87.63%, respectively. The model requires 7ms for prediction on the Raspberry Pi acting as the decision node and 5,000ms for sending the results with the display node, as detailed in Table II. In Figure 5, we present a confusion

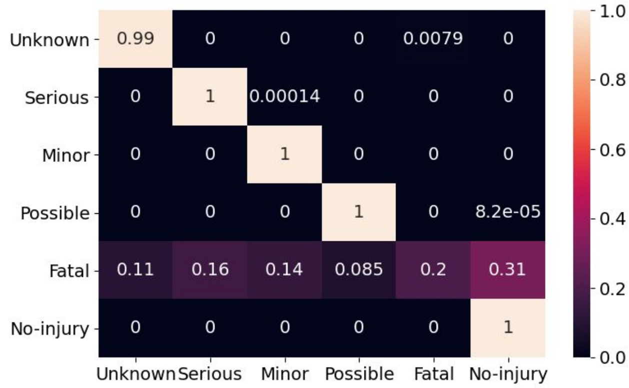

Fig. 5: Confusion Matrix

matrix, where the rows represent the actual classes and the columns represent the predicted classes. The diagonal elements of the matrix signify instances that were correctly classified, while the off-diagonal elements indicate instances that were misclassified. Our analysis reveals that the class predictions demonstrate a high level of accuracy on most classes.

V. Conclusion

In this paper, we introduce a real-time traffic crash risk prediction system aimed at improving public safety by providing early notifications to drivers. Our system is designed to address the critical need for proactive measures in reducing traffic accidents. Leveraging a combination of statistical analysis and deep learning techniques, our approach is grounded in comprehensive data analysis and predictive modeling. We begin by conducting in-depth statistical analysis utilizing five years of historical traffic crash data from the state of Texas. This analysis allows us to identify hotspot regions where a significant number of traffic crashes have occurred. Building upon this analysis, we deploy a multi-node Internet of Things system in a hotspot region. At the heart of this system lies the decision node, which is responsible for running a sophisticated deep learning-based prediction model. This model, trained offline using historical crash data, incorporates both dynamic and static features to accurately assess crash risk in real-time. Dynamic features, such as traffic flow and weather conditions, are collected through strategically placed sensors along the roadway, while static features encompass permanent characteristics of the road environment. Additionally, our system includes a display node positioned approximately 2,000 feet ahead of the decision node, ensuring early notification of predicted crash severity to approaching drivers. By seamlessly integrating statistical analysis, deep learning, and IoT technology, our system offers a proactive approach to traffic safety.

Acknowledgement

The authors would like to acknowledge the USDOT Transportation Consortium of the South-Central States (TRANSET) (# 21-034 to YFJ and MX, and #21-049 to YFJ and MX). The funding sources had no role in the design of the study; collection, analysis, and interpretation of data; or in writing the manuscript.

References

[1] V. WHO, “Global status report on road safety 2018,” *World Health Organization*, 2018.

[2] “2021 marks second deadliest year on texas roads.” [Online]. Available: https://www.txdot.gov/about/newsroom/statewide/2021-marks-second-deadliest-year-on-texas-roads.html

[3] H. Ren *et al.*, “A deep learning approach to the citywide traffic accident risk prediction,” in *ITSC*. IEEE, 2018, pp. 3346–3351.

[4] R. C. Park *et al.*, “Urban traffic accident risk prediction for knowledge-based mobile multimedia service,” *Personal and Ubiquitous Computing*, pp. 1–11, 2022.

[5] S. He *et al.*, “Inferring high-resolution traffic accident risk maps based on satellite imagery and gps trajectories,” in *CVPR*, 2021.

[6] S. Zhou, M. Xie, Y. Jin, F. Miao, and C. Ding, “An end-to-end multi-task object detection using embedded gpu in autonomous driving,” in *2021 22nd International Symposium on Quality Electronic Design (ISQED)*. IEEE, 2021, pp. 122–128.

[7] E. Cadete, R. Alva, A. Zhang, C. Ding, M. Xie, S. Ahmed, and Y. Jin, “Deep learning tackles temporal predictions on charging loads of electric vehicles,” in *2022 IEEE Energy Conversion Congress and Exposition (ECCE)*. IEEE, 2022, pp. 1–6.

[8] S. Islam, J. Deng, S. Zhou, C. Pan, C. Ding, and M. Xie, “Enabling fast deep learning on tiny energy-harvesting iot devices,” in *DATE*. IEEE, 2022, pp. 921–926.

[9] S. Islam, S. Zhou, R. Ran, Y.-F. Jin, W. Wen, C. Ding, and M. Xie, “Eve: Environmental adaptive neural network models for low-power energy harvesting system,” in *ICCAD*, 2022, pp. 1–9.

[10] Y. Lv, Y. Duan, W. Kang, Z. Li, and F.-Y. Wang, “Traffic flow prediction with big data: A deep learning approach,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 2, pp. 865–873, 2015.

[11] B. Yao *et al.*, “A deep learning framework about traffic flow forecasting for urban traffic emission monitoring system,” *Frontiers in public health*, vol. 9, p. 804298, 2022.

[12] X. Ma, H. Yu, Y. Wang, and Y. Wang, “Large-scale transportation network congestion evolution prediction using deep learning theory,” *PloS one*, vol. 10, no. 3, p. e0119044, 2015.

[13] L. Li, Y. Lv, and F.-Y. Wang, “Traffic signal timing via deep reinforcement learning,” *IEEE/CAA Journal of Automatica Sinica*, vol. 3, no. 3, pp. 247–254, 2016.

[14] B. Al-Mistarehi *et al.*, “Using machine learning models to forecast severity level of traffic crashes by r studio and arcgis,” *Frontiers in built environment*, vol. 8, p. 860805, 2022.

[15] M. A. Rahim and H. M. Hassan, “A deep learning based traffic crash severity prediction framework,” *Accident Analysis & Prevention*, vol. 154, p. 106090, 2021.

[16] M. Kamruzzaman, M. M. Haque, and S. Washington, “Analysis of traffic injury severity in dhaka, bangladesh,” *Transportation Research Record*, vol. 2451, no. 1, pp. 121–130, 2014.

[17] A. L. Achu *et al.*, “Spatio-temporal analysis of road accident incidents and delineation of hotspots using geospatial tools in thrissur district, kerala, india,” *KN-Journal of Cartography and Geographic Information*, vol. 69, no. 4, pp. 255–265, 2019.

[18] S.-h. Park *et al.*, “Highway traffic accident prediction using vds big data analysis,” *The Journal of Supercomputing*, vol. 72, pp. 2815–2831, 2016.

[19] Q. Chen *et al.*, “Learning deep representation from big and heterogeneous data for traffic accident inference,” in *AAAI*, vol. 30, no. 1, 2016.

[20] G. K. Sahoo *et al.*, “An iot-based intimation and path tracing of a vehicle involved in road traffic crashes,” in *2020 IEEE-HYDCON*, 2020, pp. 1–5.

[21] N. Desai *et al.*, “Iot based post crash assistance system,” in *ICOEI(48184)*, 2020, pp. 734–739.

[22] M. I. Mir, “Automatic accident detection and notification system using iot-based vehicular environment,” 2023.

[23] S. Chandran *et al.*, “Konnect: An internet of things(iot) based smart helmet for accident detection and notification,” in *INDICON*, 2016.

[24] “Txdot crash report purchase portal.” [Online]. Available: https://cris.dot.state.tx.us/public/Purchase/app/home

[25] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.