IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 4 February 2024, accepted 25 February 2024, date of publication 5 March 2024, date of current version 14 March 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3373885

RESEARCH ARTICLE

Road Crash Injury Severity Prediction Using a Graph Neural Network Framework

KARIM A. SATTAR, ISKANDAR ISHAK, LILLY SURIANI AFFENDEY, AND SITI NURULAIN BINTI MOHD RUM
Department of Computer Science, Faculty of Computer Science and Information Technology, Universiti Putra Malaysia (UPM), Serdang 43400, Malaysia

Corresponding author: Iskandar Ishak (iskandar_i@upm.edu.my)

This work was supported in part by Universiti Putra Malaysia through Putra Grant Scheme under Grant GP/2020/9692500, and in part by the Ministry of Higher Education Malaysia through the Fundamental Research Grant Scheme under Grant FRGS/1/2023/ICT06/UPM/02/1.

ABSTRACT Crash severity prediction is a challenging research area, where the objective is to accurately assess the extent of severity of an injury resulting from road traffic accidents. The main aim of existing studies is to precisely assess the potential severity of crashes under diverse circumstances, such as weather conditions, vehicle attributes, road characteristics and layout, and traffic control factors. This effort aids authorities in establishing effective emergency response systems. The novelty and objective of our work involve contributing to this research area by employing a graph architecture to capture relationships among various crash records to uncover any hidden patterns that traditional ML models might overlook. The current study extends existing knowledge by leveraging Graph Neural Networks (GNN) and comparing their performance to popular ensemble-based models, which include Extreme Gradient Boosting (XGBoost), Random Forest (RF), and Artificial Neural Networks (ANNs). Real data from the United Kingdom (UK) was employed to achieve our goal. The data was obtained from the Department for Transport open data portal. All models underwent training using the training dataset, followed by performance evaluation using diverse metrics such as the accuracy, precision, recall, f1-score, Matthews Correlation Coefficient (MCC), confusion matrix, and computational cost on the test dataset. Overall, our proposed GNN-based model demonstrated better performance when compared to other models. Specifically, the GNN model outperformed all other models across all metrics. For instance, the accuracy of the GNN model was 85.55% as compared to 83.36%, 83.18%, and 83.27% for the XGBoost, RF, and ANN models, respectively. The GNN model assisted in identifying hidden patterns by considering non-linear relationships among crash records. Thus, the model had the potential to improve its ability to predict severe accidents, which could in turn significantly improve emergency response efforts and reduce the likelihood of severe accidents resulting in fatalities.

INDEX TERMS Categorical embedding, graph neural network, GraphSAGE, kNN graph, road crash injury severity.

I. INTRODUCTION

Road safety research encompasses a wide array of topics and is a multidisciplinary field of interest. Prediction of accident injury severity is a popular area of research in road safety [1], [2]. With the rapid urbanization of cities and the concentration of private vehicles in urban areas, the topic has become of the utmost importance due to increased accident rates [3]. The availability of various data sources, and data-driven applications enables transport engineers to develop new prediction models; thereby contributing to mitigate the effects of crashes. The outcomes can be used in various sectors, including the development of new policies, the modification of the existing road infrastructure, the integration of prediction

The associate editor coordinating the review of this manuscript and approving it for publication was Li He.

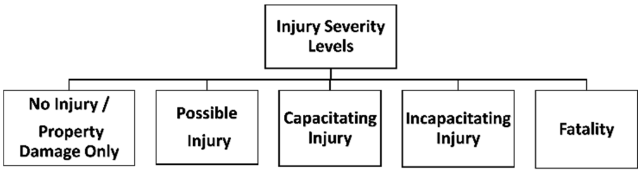


FIGURE 1. Crash injury severity levels.

models with external systems such as healthcare and emergency services, and the promotion of different awareness plans for the public.

In the literature, from the perspective of road transportation, a road crash or accident is described as an unforeseen incident that occurs on a road [4]. Typically, it involves one or more motor vehicles, such as cars, trucks, or motorcycles, and its consequences may lead to injuries, fatalities, or property damage. Various prediction techniques exist depending on the problem domain in road crashes. Some studies delved into identifying road crash predictions to assess crash likelihood and pinpoint high-risk areas where accidents are more likely to occur [5], [6]. Another field of study related to crash predictions is the road crash rate prediction, which deals with the estimation of crash frequency [7], [8], [9]. Lastly, crash severity injury prediction techniques aim to estimate severity levels depending on various injury severity categories [1], which is also the focus of this paper. Traffic crash severity prediction is an active research area where the research community is actively involved in risk assessment and mitigation of vehicle crashes. Different studies have been done to identify relationships between crash injury severity and different risk factors associated with human traits, environmental factors, and road geometry. Fig. 1 shows various categories of crash severity, ranging from no injury or property damage only (PDO) to fatalities at the higher end of the severity scale. Some authors have employed three crash severity levels [10], [11] which include PDO with no injuries, injuries, or fatalities.

Two types of techniques are popular in the literature for crash injury severity predictions, which include classical statistical modeling and machine learning (ML) techniques. Traditional statistical models have historically held a dominant position, where they played an essential role in helping researchers discover the key factors that influence the severity of injuries. Researchers use these models to investigate the elements influencing the seriousness of traffic crashes [12], [13], [14], [15], [16], [17]. However, such models rely on a priori hypothesis to establish the expected relationships between the variables of interest. In practice, these initial assumptions may not accurately represent the true nature of these variables, potentially leading to incorrect conclusions. Additionally, statistical models are better suited for examining relationships in datasets with limited sample sizes and features.

Recent developments in this field witnessed a surge in the application of machine learning (ML) algorithms, owing to their capacity to provide augmented precision. Adaptive and highly accurate models can be produced using flexible, non-parametric methods like ML and deep learning (DL). These methods prove to be particularly advantageous in cases where the connection between predictor attributes and the targeted levels of injury severity remains poorly understood or exhibits a highly nonlinear relationship [18], [19]. RF, a commonly employed tree-based ensemble ML technique, has been found to be popular in different crash injury severity studies [20], [21]. In addition to RF, other techniques that are prominently used to predict crash injury severity include decision trees [19], [21], support vector machines (SVMs) [22], and XGBoost [19], [23]. Various studies used DL techniques to predict road crash severity [22], [24], [25].

In the context of road crash data, categorical features represent distinct categories or labels, such as the driver's gender, vehicle type, or lighting conditions. The fundamental characteristic of these features is their lack of inherent order. Unlike numerical features, where values have a clear and meaningful order (such as a road speed limit), categorical values do not possess any inherent ranking or hierarchy. This absence of inherent order has significant implications for data handling in machine learning models. Therefore, it becomes necessary to use appropriate techniques for encoding or transforming them into a suitable format. Additionally, crash severity data is typically categorical in nature, with only a couple of features having numerical values, such as speed. Hence, there is a need to preprocess them into high-quality numerical data, using techniques such as embedding or in sparse formats like one-hot encoding. In one-hot encoding, each category value is transformed into a binary vector. Because each unique category is represented by a different binary vector in the dataset, this transformation can result in high-dimensional data when there are many categorical values. Additionally, the one-hot encoding technique may lead to high memory usage and longer modeling times, especially when dealing with many categorical features [26]. Careful consideration of these factors is essential when applying such preprocessing techniques to crash severity data. Another advanced method is the embeddings technique, which is used to represent categorical data as continuous-valued vectors in a lower-dimensional space. These continuous-valued vectors are called embeddings [76]. Additionally, this method enables models to comprehend the connections, parallels, and associations between various categories that the one-hot encoding method does not. These embeddings capture essential information about the categorical features while reducing the dimensionality compared to one-hot encoding. In other words, embeddings also help reduce the dimensionality of data, which is also known as dimensionality reduction.

The k nearest neighbors (kNN) graph method is based on the theory of kNN. kNN assists in predicting the label of a new sample by examining the labels of the k most similar samples within the dataset. The idea is that the label of the new sample may share similarities with those of its k-nearest neighbors [27]. The applications of the kNN graph in machine learning include data classification and graph-based classification. The construction of a kNN graph for high-dimensional data represents a crucial data structure with numerous applications spanning diverse domains such as data mining and machine learning. One of the properties of the kNN graphs is that they will always connect k neighbors for a given node [28]. Reference [29] mentioned that the kNN graph considered each data point in the dataset as an individual graph node. It then established directed edges connecting each graph node to its k nearest neighbors. The aforementioned process systematically repeated for each individual data point in the dataset.

This research presents the development and usage of a DL model that employs the Graph SAmple and aggreGatE (GraphSAGE) GNN model [30] to predict the severity of injuries resulting from road accidents. Based on our initial survey, it is found that no similar study on crash injury severity prediction has been taken using GNN. One potential explanation could be that the crash severity dataset lacks the inherent graph structure necessary for GNNs, as these models depend on input data with a graph-like structure. Previous works on crash severity prediction using ML and DL methods have utilized datasets to predict severe cases in accidents but have failed to model the relationships among the crash records in the form of a graph. Reference [31] suggested that generating graph structures based on tabular data could facilitate the understanding of samples relationships and feature interactions present in the dataset. GNNs have been gaining popularity in various fields and are considered one of the most prominent DL techniques, alongside other competitive technologies and ensemble machine learning techniques. GNNs utilize graph structures to process and extract information, making them particularly well-suited for tasks such as edge link classification, node classification, and graph classification [32]. In this study, our focus is on predicting crash injury labels, which fall under the category of node classification. GNN differs from traditional ML algorithms, such as RF or SVM, by naturally capturing feature correlations in graph-structured data through consideration of the local and global neighborhoods of each node. In the context of road crash data, the dataset is typically presented in a tabular format. Tables are commonly used to represent data in a generic manner. In contrast, graphs utilize a specific data structure. To transform the crash dataset into a format usable by GNN, a framework is required in which categorical features of the dataset are first converted into embeddings. These embeddings are then used to create a graph dataset. In the graph representation, each node symbolizes an element, which corresponds to a complete record in a table. In graph theory, an edge is a fundamental element that establishes a connection between two nodes, thereby denoting their relational association. In the context of this study, an edge will represent an association between two crash data records. In our case, each data record is characterized as a node, and relationships among neighbors are recorded as edges. This approach allows for the utilization of GNN's capabilities in capturing complex relationships and dependencies between different elements of the crash dataset, ultimately enhancing the prediction of injury severity in traffic accidents.

Taking everything into account, the proposed framework brings forth the following significant contributions:

- Introducing a GraphSAGE GNN model designed to do the prediction of crash injury severity using the data records from the UK.
- Presenting a kNN graph-based approach for constructing graph data from the UK accident records.
- Assessing the proficiency of the proposed framework by comparing it with various popular ensemble models and ANN model known for their effectiveness in predicting crash injury severity.

II. RELATED WORK

A. EMBEDDING TECHNIQUES

Embedding techniques can be employed to represent categorical data as continuous-valued vectors in a lower-dimensional space, effectively mitigating memory overhead, especially when dealing with a large number of categorical features [76]. Traditional algorithms require numerical inputs, leading to the utilization of encoding methods to transform categorical values into numerical ones [33]. To generate categorical embeddings, various techniques have been proposed. These include using deep learning to create distributed representations for categories [33], capturing complex relationships between categorical and numerical features through representation learning [34], employing graph-based methods to learn categorical data representations [35], and utilizing domain knowledge and semantic similarity measures for embeddings [36], [37]. All these methods aimed to convert categorical features into numerical vectors, enhancing their usability in machine learning algorithms and improving classification performance.

Categorical features lack an inherent order, which addresses the challenge of clustering categorical data, a prevalent issue in machine learning. The problem arises due to the absence of inherent order in categorical features. Reference [35] introduced a graph-based framework for categorical data clustering that effectively acquires representations from similarity graphs. The framework under investigation exhibited its superiority over existing methodologies through comparisons conducted on benchmark datasets. This noteworthy achievement established its significance as a valuable addition to the realm of categorical data clustering. Reference [33] addressed the common challenge of the necessity to process categorical data for tasks such as classification and regression in machine learning and deep learning. The authors introduced a novel technique that involved assigning a unique vector to each category, and the characteristics of these vectors were obtained through the process of training a neural network. The process encompassed the creation of a data vocabulary, tokenization of categorical data, and mapping it to word vectors through feature learning. When comparing this deep-learned embedding technique to one-hot encoding, it was observed that the proposed technique outperformed the latter, achieving a higher F1 score (89%) compared to one-hot encoding (71%) when training data using the Long Short-Term Memory (LSTM) model.

Reference [38] addressed the challenge of processing mixed data comprising categorical and numerical attributes and proposed an innovative solution called Attribute-Weighted Isometric Embedding (AWIE) to enhance data transformation quality. In AWIE, the authors combined isometric embedding and attribute weighting, effectively mitigating dimensionality expansion and improving classification performance. Another novel approach to categorical representation learning was termed the 'categorifier' by [39]. The proposed solution addressed the challenge of improving representation learning beyond traditional set-theoretic methods. The article proposed a category-theoretic approach to representation learning.

B. GRAPH DATA GENERATION

Reference [28] introduced methods for building graphs from flat data and improved the performance of graph-based algorithms. The authors discussed several methods, such as kNN, for creating similarity-based graphs. Reference [40] generated a correlation graph using the Spearman correlation approach. In order to ensure that the model received only informative associations, the authors maintained the correlations bigger than 0.55 in the graph and substituted a bidirectional connection with a constant value of 1 to generate an adjacency matrix. The minimum value of 0.55 was selected to provide a decent number of connections without overwhelming the network. A secondary goal in selecting the 0.55 threshold was to achieve equilibrium among connections and reduce the quantity of singleton nodes, or nodes that were disconnected from any other node in the network.

Reference [41] used a tabular dataset and represented patients as nodes in a kNN graph. The connections between patients were represented by edges, which indicated the relationships between individuals based on the similarity of their features. To determine how similar the feature vectors in data records were to one another, the authors created a similarity metric. To determine the edge information, the authors employed the chosen measure to find the k-nearest neighbors for every data record. After the kNN graph was built, the produced graph dataset was utilized in several models, including GNNs.

GNNs are commonly utilized in graph prediction applications, including graph classification and node classification. GNNs have shown great accuracy in graph classification tasks. Studies have been proposed to enhance the accuracy and utilization of GNN models [41], [42]. Reference [43] proposed a novel architecture called Boosted GNN (BGNN), which was a combination of Gradient Boosted Decision Trees (GBDT) and GNN, to address the challenge of graphs with tabular node features. It combined the strengths of GBDT and GNN in a unique way. While GBDT exceled at handling heterogeneous tabular data, GNNs were proficient in capturing the graph structure. The work demonstrated significant performance improvements over existing GBDT and GNN models through extensive experimental comparisons on various graphs with tabular features. Reference [41] proposed a fusion model, comprised of the GNN and tabular data models, for predicting chronic kidney disease, where the former model helped in identifying complex connections between kidney patients and their medical illnesses, and the later model carefully handled patient-specific features.

C. CRASH INJURY PREDICTION

Crash injury severity studies utilized various statistical methodologies and traditional machine learning, tree-based ensemble, and deep learning models. For example, probit models were utilized by [2] to examine characteristics that influence the degree of injuries sustained in truck-related incidents. In recent years, an increasing amount of research used random parameter models to analyze the severity of injuries sustained in road accidents [44], [45], [46]. A DL model for predicting accident severity built on the TabNet framework, called TabVAE, was proposed by [24]. The suggested TabVAE model selected and extracted features through a multi-step decision-making process. To predict injury severity, the TabVAE model comprised multiple networks: the observation network gathered observed data; the heterogeneity network used contrastive learning and variational autoencoder technique extracted unobserved heterogeneity; and the aggregation network combined the observed and unobserved data. Reference [47] forecasted road accidents and employed various ANN models to predict the severity of road crashes. Reference [48] conducted a literature review of the neural network methodologies that were used in the analysis of traffic accidents. Their review encompassed a selection of articles, primarily focusing on the prediction of crash severity. The researchers also provided concise guidance regarding the thoughtful selection of appropriate neural network techniques tailored to the specific type of accident prediction.

Reference [49] used an LSTM neural network for modeling traffic accident-related factors and predicting accident severity classes. The authors used two fully connected layers on top of the LSTM layers to accelerate learning and to align the output dimensions of the LSTM layer with the number of crash severity classes. The final output layer, constituting a fully connected feed-forward layer, was designed to directly map the learned features to the crash severity classes. To avoid overfitting, dropout layers were also incorporated into the architecture. Table 1 lists a compilation of recent references on different ML and DL models that were used for predicting crash injury severity.

III. DATA DESCRIPTION

A. ABOUT THE DATASET

The dataset utilized in this research was the UK road accident dataset, accessible at [58]. The original dataset contained

TABLE 1. Compilation of recent references on ML models for predicting crash severity.

Authors' Name	Reference	Year	Region	Features Category	Severity Considered	Models Used
Ahmed et al., (2023)	[1]	2023	New Zealand	Crash Details, Road Conditions, Vehicle Details, Driver Details, and Environment Factors	Non-Injury, Minor, Serious, and Fatal	**RF**, Decision Jungle (DJ), Adaptive Boosting (AdaBoost), Categorical Boosting (CatBoost), XGBoost, and Light Gradient Boosting Machine (LGBM)
Dimitrijevic, Asadi, & Spasovic (2023)	[50]	2023	USA	Crash Details, Road Conditions, Vehicle Details, and Environment Factors	Property Damage Only (PDO), Injury and Fatality	**Optimized SVM based upon Genetic Algorithm (GA-SVM)**, Optimized SVM based upon greedy-search (GS-SVM), and traditional SVM
Mohammadpour, Khedmati, & Zada (2023)	[51]	2023	USA	Crash Details, Road Conditions, Vehicle Details, Driver Details, and Environment Factors	Fatality and Severe injuries, Less Severe Injuries, and PDO	kNN**,** SVM, **RF**, GBDT, and MLP
Pérez-Sala et al., (2023)	[52]	2023	Spain	Crash Details, Road Conditions, Vehicle Details, Driver Details, and Environment Factors	Slight Injury, Severe Injury, and Fatal	kNN, Naive Bayes, Support-Vector Classifier (SVC), **1-D CNN**, and **2-D CNN**
Raja et al., (2023)	[47]	2023	Ethiopia	Crash Details, Road Conditions, Location Details, and Environment Factors	Slight, Serious, and Fatal	**Recurrent Neural Networks (RNN)**, Multilayer Perceptron Neural Network (MLPNN), Radial Basis Function Neural Network (RBFNN), Backpropagation Neural Network (BPNN), Feed Forward Neural Network (FFNN), and LSTM
Panda et al., (2023)	[53]	2023	India	Vehicle type, Road Type, Condition Type, Traffic Violation Type and Fault Type	Killed and Injured	SVM, RDF, **GBM**, and XGB
Megnidio-Tchoukouegno & Adedeji (2023)	[54]	2023	UK	Crash Details, Road Conditions, Vehicle Details, Driver Details, and Environment Factors	Fatality, Serious Injury, and Slight Injury	**Decision Tree**, LightGBM, and XGBoost.
Niyogisubizo et al., (2023)	[55]	2023	New Zealand	Crash Details, Road Details, Lane Type, Road Conditions, and Environmental Factors.	Non-Serious and Serious	kNN, **The Wide and DL model**, XGBoost, AdaBoost, and Gradient Boosting (GB)
Niyogisubizo et al., (2023)	[56]	2023	UK	Crash Details, Road Type, Road Conditions, Vehicle Details, and Environment Factors	Non-Serious and Serious	**A hybrid of Balanced Bagging Classification (BBC) and LGBM**, BBC, LSTM, Gaussian Naïve Bayes (GNB), SVM, and RF
Azhar et al., (2022)	[57]	2022	Malaysia	Crash Details, Road Conditions, Vehicle Details, Driver Details, and Environment Factors	Fatal, Severe injury, Slight injury, and No-injury	**CART, RF**

approximately 200,000 records. The data was preprocessed to remove null, noisy, or incomplete entries. Subsequently, the data was filtered to retain only vehicle-to-vehicle accidents; thus, the number of records was reduced to about 100,000. The processed dataset exhibited an inherent class imbalance, necessitating the application of random under sampling to reduce the number of records from the majority class. The dataset encompassed various feature categories, including crash severity type, road configurations, spatial and temporal factors, environmental variables, and vehicle characteristics. Notably, all variables within the dataset were categorical. The final dataset included data from 2011 to 2016 for the purpose of developing and testing the models. The predictions were based on a dataset of 7611 crash instances. Table 2 offers a comprehensive overview of these features, presenting their respective frequencies and the distribution of severity labels. Additionally, certain attributes from the original dataset, such as age, gender of the driver, or the presence of police involvement, were omitted from this subset. This decision was made to streamline the identification of severity types with a minimal set of features, as many crash attributes remain unknown until an initial emergency or police response occurs. The primary objective here is to proactively classify severity types, enabling the consideration of appropriate actions in advance.

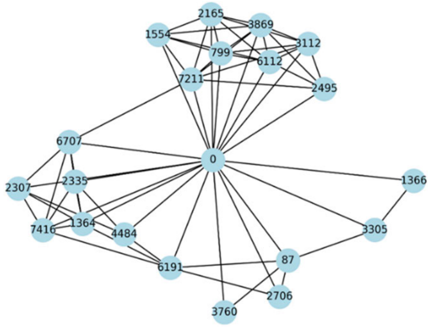


FIGURE 2. A subset of graph data for a sample crash record with ID '0'.

B. DATASET PREPARATION

Incorporating GNNs for classification necessitated a fundamental data transformation process wherein the data was structured into a graph representation. Within this graph structure, entities were represented as nodes, and their interconnections were denoted as edges. The first step involved in the conversion of the tabular dataset into a graph structure was to transform each data record into a node within the graph. The collection of 16 features in the dataset corresponding to each node (record) served as attributes for the respective node within this graph representation. Since the UK dataset was divided into five feature types, which included road configuration, spatial configuration, vehicle characteristics, environmental conditions, and temporal variables, features in each feature type were further encompassed by different sub-features, e.g., temporal variables included day of the week and time ranges. PCA based embedding technique was used to convert these categorical values into embeddings [80]. To establish connections between these nodes, relationships needed to be identified. One such method involved creating edges between nodes based on certain feature similarities that surpassed a predefined threshold. One such method is to use kNN graph algorithm, which can help establish interconnections between nodes and their k nearest counterparts, determined by shared feature similarities. Once the PCA based embeddings were generated, graph data was prepared in which each node has k neighbors showing similar relationships to the node under consideration. PCA embeddings were used to establish connections (edges) among graph nodes during the data processing technique for constructing graph data. Fig. 2 mainly illustrates a subset of a graph data pertaining to a crash record with ID '0' and showing its relationships or interconnections with 20 neighboring crash records which are resulted from kNN graph algorithm.

IV. METHODOLOGY

In this study, the performance of the GNN model was investigated using the factors mentioned in Table 2. To compare its performance with popular machine learning models, we selected RF [59], XGBoost [60] and Multi-Layer Perceptron (MLP) based ANNs [61]. These techniques were chosen due to their widespread use in classification problems as well as their ability to resist overfitting and their tendency for high predictive accuracy [62], [63], [64]. Ensemble methods, both RF and XGBoost models, incorporate the predictions of multiple individual models, thereby reducing bias and variance. Similarly, ANNs are known for their learning techniques using nonlinear functions, which help map input features to target variables [61]. The details of the GNN architecture are discussed in the next section. Afterwards, the evaluation metrics that were used in this study are presented. The GridSearchCV optimization technique, which is used to select and fine-tune different hyperparameters pertaining to the GNN model and the other models, is discussed in Section IV(C).

A. GNN MODEL

Graph Neural Network (GNN), a subset of DL neural networks, is specifically designed to process data structured as graphs. GNN models are specialized DL networks designed for processing graph data. Graph data is typically defined as G = (V, E), where V denotes nodes (entities) and E denotes edges (relationships) between nodes. Despite its voluminous and complex structure, graph data can be represented using adjacency matrices. In the case of an undirected graph, the adjacency matrix exhibits symmetry. In this representation, each node corresponds to a row and column in the matrix, and edges are represented as matrix entries. GNNs have various applications, including node classification, link prediction, and graph classification.

In the context of this study, node classification is a significant application in graph analysis. It aims to predict the label associated with each node. Labels can represent types, categories, or attributes, among other possibilities. Successful node classification techniques often rely on exploiting interconnections between nodes. As discussed in [65], a key concept in this domain is "homophily" [66], which refers to nodes sharing attributes with their neighbors in the graph. ML models can be constructed based on homophily to assign similar labels to connected nodes in a graph [67].

The main function of GNN is to learn representations for nodes and edges within a graph. To accomplish this, data is collected from nodes that are in the neighborhood of the target node. Typically, each node is represented as a low-dimensional vector, which encodes not only the node's characteristics, but also its interactions with other nodes (i.e., its edges) within the graph. Furthermore, a GNN network consists of one or more layers designed to efficiently capture increasingly complex features of the graph. It accomplishes this by aggregating important data from the surrounding neighborhood of each individual node at each layer. In essence, GNNs are structured to progressively understand and represent the intricate relationships and properties of the graph.

There are different types of GNN models that exist including Graph Convolutional Network (GCN) [68], Graph Attention Networks (GAT) [69], and Graph Sample and Aggregated (GraphSAGE) [30], [75]. The key function that differentiates GNNs from other neural networks is the

TABLE 2. Dataset and its features.

Feature Type	Categorical Features	Description	Frequency	Percent (%)	Non-Severe	Severe
Road Configuration	**Road Type**	Single carriageway	5998	78.8%	3168	2830
		Dual carriageway	1220	16.0%	475	745
		One way street	175	2.3%	135	40
		Round about	170	2.2%	105	65
		Slip road	42	0.6%	17	25
		Unknown	6	0.1%		6
	Road Class	Motorway	177	2.3%	5	172
		A(M)	20	0.3%	0	20
		A	4706	61.8%	2588	2118
		B	758	10.0%	327	431
		C	816	10.7%	494	322
		Unclassified	1134	14.9%	486	648
	Speed Limit	20	66	0.9%	29	37
		30	5333	70.1%	3687	1646
		40	436	5.7%	79	357
		50	327	4.3%	82	245
		60	1023	13.4%	7	1016
		70	426	5.6%	16	410
Spatial Configuration	**Area Type**	Urban	5565	73.1%	3821	1744
		Rural	2046	26.9%	79	1967
	Junction Location	Not at or within 20 meters of junction	2971	39.0%	866	2105
		Approaching junction or waiting/parked at junction approach	1373	18.0%	816	557
		Cleared junction or waiting/parked at junction exit	704	9.2%	407	297
		Exiting roundabout	36	0.5%	6	30
		Entering roundabout	25	0.3%	0	25
		Exiting main road	117	1.5%	46	71
		Entering main road	163	2.1%	42	121
		Joining from slip road	16	0.2%	5	11
		Mid Junction - on roundabout or on main road	2190	28.8%	1696	494
		Others	16	0.2%	16	0
	Junction Control	Not at junction or within 20 meters	2988	39.3%	882	2106
		Authorized person	9	0.1%	5	4
		Auto traffic signal	1264	16.6%	991	273
		Stop sign	33	0.4%	17	16
		Give way or uncontrolled	3317	43.6%	2005	1312
	Junction Detail	Not at junction or within 20 meters	2988	39.3%	882	2106
		Roundabout	239	3.1%	134	105
		Mini roundabout	66	0.9%	44	22
		T or staggered junction	2859	37.6%	1827	1032
		Slip road	77	1.0%	20	57
		Crossroads	1102	14.5%	853	249
		More than 4 arms (not roundabout)	80	1.1%	59	21
		Private drive or entrance	137	1.8%	64	73
		Other junction	63	0.8%	17	46
	Carriageway Hazards	None	7496	98.5%	3848	3648
		Vehicle load on road	8	0.1%	0	8
		Other object on road	39	0.5%	11	28
		Previous accident	11	0.1%	0	11
		Pedestrian in carriageway - not injured	39	0.5%	32	7
		Any animal in carriageway (except ridden horse)	18	0.2%	9	9
	Road Surface Conditions	Dry	5618	73.8%	3032	2586
		Wet or damp	1886	24.8%	829	1057
		Snow	17	0.2%	8	9
		Frost or ice	81	1.1%	30	51
		Flood over 3cm. deep	9	0.1%	1	8

TABLE 2. ***(Continued.)*** **Dataset and its features.**

Vehicle Characteristics	**Vehicle Type**	Car	4729	62.1%	2258	2471
		Motorcycle <= 50cc	85	1.1%	63	22
		Motorcycle <= 125cc	400	5.3%	286	114
		Motorcycle > 125cc & <= 500cc	118	1.6%	67	51
		Motorcycle > 500cc	589	7.7%	185	404
		Taxi cars	382	5.0%	302	80
		Bus or coach (17 or more pass seats)	608	8.0%	468	140
		Van / Goods 3.5 tons mgw or under	425	5.6%	222	203
		Goods 7.5 tons mgw and over	212	2.8%	27	185
		Goods over 3.5t. and under 7.5t	40	0.5%	21	19
		Minibus (8 - 16 passenger seats)	12	0.2%	0	12
		Agricultural vehicle	4	0.1%	0	4
		Other vehicle	7	0.1%	1	6
Environmental Conditions	**Light Conditions**	Daylight	4860	63.9%	2681	2179
		Darkness - lights lit	2067	27.2%	1191	876
		Darkness - lights unlit	28	0.4%	7	21
		Darkness - no lighting	582	7.6%	12	570
		Darkness - lighting unknown	74	1.0%	9	65
	Weather Conditions	Fine no high winds	6423	84.4%	3301	3122
		Raining no high winds	832	10.9%	470	362
		Snowing no high winds	20	0.3%	8	12
		Fine + high winds	82	1.1%	26	56
		Raining + high winds	72	0.9%	28	44
		Snowing + high winds	2	0.0%	1	1
		Fog or mist	33	0.4%	8	25
		Other	85	1.1%	39	46
		Unknown	62	0.8%	19	43
Temporal Variables	**Day of Week**	Sunday	952	12.5%	395	557
		Monday	1016	13.3%	519	497
		Tuesday	1106	14.5%	591	515
		Wednesday	1091	14.3%	636	455
		Thursday	1162	15.3%	604	558
		Friday	1154	15.2%	607	547
		Saturday	1130	14.8%	548	582
	Time-range	00:00-06:00	841	11.0%	294	547
		06:00-12:00	1844	24.2%	1003	841
		12:00-18:00	2844	37.4%	1550	1294
		18:00-00:00	2082	27.4%	1053	1029
		Total	**7611**	**100.0%**	**3900**	**3711**

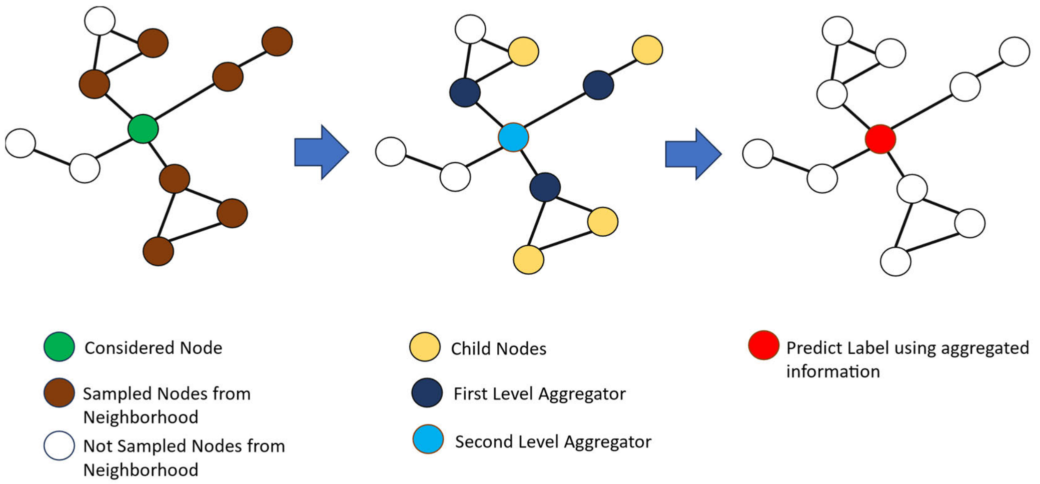

Considered Node
Sampled Nodes from Neighborhood
Not Sampled Nodes from Neighborhood
Child Nodes
First Level Aggregator
Second Level Aggregator
Predict Label using aggregated information

FIGURE 3. **GraphSAGE sample and aggregate approach structure.**

Message Passing Neural Network (MPNN). MPNN operates by iteratively updating the state for each node in a graph through a process involving message aggregation and state update. Depending on the specific type of GNN being used, such as GCN, GAT, or GraphSAGE, the details of how MPNN is applied may vary. These GNN variants incorporate different mechanisms for message aggregation and state updates.

Our study used GraphSAGE network which is a model that generates feature representations of graph nodes, which are useful for predictions and graph analysis [30]. Both GraphSAGE [30] and GAT [69] are enhancements to GCN. GraphSAGE takes an inductive approach, learning rules from training data and applying them to test data. On the other hand, GCN uses a transductive approach, which means it has limitations when it comes to generalizing to unseen data [30]. GCN also becomes computationally expensive as the number of convolutional layers increases. GraphSAGE and GAT, similar to GCN, are designed to capture neighboring features based on graph structures. During model training, for each node, GAT uses the attention technique to compute attention scores for its neighbors. This process becomes computationally expensive, particularly for large and dense graphs [81]. In contrast, GraphSAGE addresses this issue by employing sampling techniques to handle the computational challenges.

In GraphSAGE, the target node's neighbors are randomly selected at first. Subsequently, the target node and its neighboring nodes are collectively aggregated to generate a novel feature representation for the target node. The newly created vector comprises the feature information of the target node as well as its neighboring nodes. Also, GraphSAGE enhances model optimization by using small batch training and different aggregator functions for neighbors' features. The aggregation function, which takes the representations of a node's neighbors as input and produces a new representation for the node, is the most important component of a GNN layer. There are three types of aggregator functions: mean aggregator, LSTM aggregator, and pooling aggregator. The mean aggregator computes the mean of adjacent vectors. The LSTM aggregator uses LSTM networks to properly manage and evaluate data received from nearby nodes, whereas the Pooling aggregator employs max-pooling after neural network-based neighbor processing.

Fig. 3 illustrates the flow of the sample and aggregate approach for GraphSAGE. Initially, nodes will be sampled in the network from the local neighborhood of the considered node. Then, child node information will be aggregated at the first level using a selected aggregate mechanism, e.g., the mean aggregator. In our study, we used mean aggregator function. In the second pass, the final aggregation will happen at the node level.

B. EVALUATION METRIC

To gauge the efficacy of the embedding-based kNN graph-constructed proposed model, well-established metrics for binary classification were used in this study. These metrics included accuracy, precision, recall, and f1-score. These metrics helped in identifying valuable insights into the model's ability to generalize as well as make accurate predictions, ultimately guiding its optimization. In addition, a confusion matrix was also computed to identify true positive (TP), false positive (FP), false negative (FN), and true negative (TN) metric values [18], [33]. To understand further, TP denoted the accurate identification of severely injured records by the classifier model. FP represented instances where non-severe records were erroneously classified as severe, TN signified the correct classification of non-severe records, and FN indicated cases where severe records are incorrectly classified as non-severe.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TN} + \text{FP} + \text{FN} + \text{TP}} \tag{1}$$

In this study, accuracy was quantified by calculating the ratio of true predictions (TP +TN) to the total number of predictions made (TP + FP + TN + FN), where TP and TN denoted cases in which the model correctly predicted actual severe and non-severe cases, respectively. On the other hand, FN and FP represented cases where the model incorrectly predicted actual severe and non-severe cases, respectively, by marking them as non-severe and severe. The TP, FP, TN, and FN values were also helpful in computing precision and recall.

$$\text{Precision} = \frac{\text{TP}}{\text{FP} + \text{TP}} \tag{2}$$

The precision metric was computed using the ratio of actual predicted severe cases (TP) over all samples that were predicted as severe cases (TP + FP), where FP denoted cases where non-severe cases were predicted as severe cases and TP indicated actual predicted severe cases. It demonstrated the model's ability to make accurate positive predictions, indicating how many of the predicted positive cases were positive.

$$\text{Recall or Sensitivity or TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{3}$$

On the other hand, the recall metric, also referred as the true positive rate (TPR) or sensitivity, was calculated using the ratio of actual severe cases (TP) over all samples that were truly severe cases (TP + FN). In other words, recall evaluated the ability of the model to accurately detect all instances of positive records within the set of actual positive records. To summarize, recall in a model referred to its ability to correctly recognize all positive instances, while precision pertained to its ability to accurately identify only those instances that are relevant.

$$\text{F1} - \text{score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4}$$

We also employed the F1-score to evaluate model performance, which is a harmonic mean of recall and precision. This metric helps in identifying a value that would balance both precision and recall values.

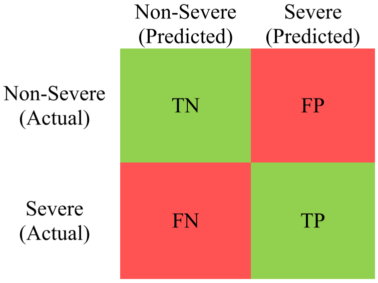
The other two metrics that are seldom used are TPR and false positive rate (FPR) [18]. TPR was calculated using the same method as recall. However, FPR quantified the percentage of negative records (non-severe accidents) that the model mistook for positive records (severe cases).

$$\text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}} \tag{5}$$

$$\text{MCC} = \frac{\text{TP} \times \text{TN} - \text{FP} \times \text{FN}}{\sqrt{(\text{TP} + \text{FP}) \times (\text{TP} + \text{FN}) \times (\text{TN} + \text{FP}) \times (\text{TN} + \text{FN})}} \tag{6}$$

Another metric commonly used in classification problems is the Matthews Correlation Coefficient (MCC). Similar to accuracy, this metric also utilized TN, FP, FN, and TP, and its value ranged from −1 to 1, where −1 indicated the worst performance and +1 denoted the best performance of the model [77]. It differed from accuracy in a way that it produced a high value only in cases where it could correctly predict the majority of positive records as well as negative records [78]. McNemar's test [82], which was a statistical significance test [83], was also used. It was used to test null hypothesis to compare predictions of selected machine learning models.

Lastly, a confusion matrix was also used to assess the efficacy of models by utilizing all key metrics (such as TN, FP, FN, and TP) [18], [33]. For the binary classification problem, the matrix is represented by a 2 × 2 table, as shown in Fig. 4.



	Non-Severe (Predicted)	Severe (Predicted)
Non-Severe (Actual)	TN	FP
Severe (Actual)	FN	TP

FIGURE 4. Confusion matrix structure.

C. HYPERPARAMETER OPTIMIZATION

In order to optimize the performance of the model, various hyperparameters were adjusted using a GridSearchCV technique, with the objective of maximizing the accuracy metric employed for evaluating the model. Using a systematic grid search, multiple network architectures and configurations were evaluated to determine the optimal network for predicting the severity of traffic accident injuries. In Table 3, configurations for hyperparameter optimization using GridSearchCV for various ML models are outlined. For the 'GNN' model, embedding dimensions for categorical data were optimized, in addition to the number of hidden layers, hidden neurons, and drop rate. Also, while constructing the graph, the number of neighbors, mode value, and metric values are also optimized. The 'num_neighbors' attribute represented the value of 'k' used to determine the number of nearest neighbor nodes sampled for each node during the aggregation step.

For the RF model, hyperparameter options such 'n_estimators,' 'max_depth,' 'min_samples_split,' 'max_features,' 'min_samples_leaf,' and 'criterion' were optimized. The criterion option was helpful in determining on how the impurity of nodes was measured when building the trees. On the other hand, max_features hyper parameter helped in better generalization of the model while controlling the number of features that shall be considered at each split in the tree. Similarly, for the 'XGBoost' model, hyperparameter configurations for 'n_estimators,' 'learning_rate,' 'max_depth,' 'min_child_weight,' 'gamma,' 'subsample,' and 'colsample_bytree' were shown. In XGBoost, the n_estimators parameter helped increase the number of boosting rounds, which can improve the model's performance. Similarly, a higher value of the max_depth parameter allowed individual trees in the ensemble to capture more complex patterns. Lastly, for MLP based ANNs, various architectures ranging from single layer to multiple hidden layers were considered during hyper parameter trainings. To introduce non-linearity to the ANN model, different activation functions were also explored. "Adam" or "sgd" were used as solver functions. More details on the hyperparameters of XGBoost, RF, and ANN can be found at [70]. These configurations served as a roadmap for the hyperparameter tuning process, systematically exploring different parameter combinations to optimize the performance of selected ML and DL models.

Fig. 5 illustrates the experimental setup of optimized prediction models. The UK crash dataset served as the basis for the modeling of the four models namely: GraphSAGE, RF, XGBoost, and ANN. For this study, we proposed an embedding-based GraphSAGE GNN model. For the GNN model, a data preparation step was undertaken, and to achieve our goal, we used the Principal Component Analysis (PCA) embedding technique [71] to convert all categorical variables into embeddings. Then, we used the popular K-neighbor graph-based model to construct a graph that considered data records, relationships among each other.

Afterwards, we used the GraphSAGE model to predict node level classification to infer whether node data records related to severe or non-severe class. The performance of the model was assessed using a set of metrics. On the other hand, for the RF, XGBoost, and ANN models, preprocessing involved converting categorical features into one-hot encoding. These prepared datasets were then utilized to train these models. Similar to the GNN model, their performance is also evaluated against a set of predefined metrics.

V. FINDING AND ANALYSIS

Fig. 6 illustrates the top-level structure of the GraphSAGE model, leveraging the capabilities of GNNs. The figure also outlines the components and flow of the model for predicting accidents based on various factors. PyTorch Geometric (PyG), a DL library based on PyTorch [72], was employed to implement a GraphSAGE-based GNN model. This library

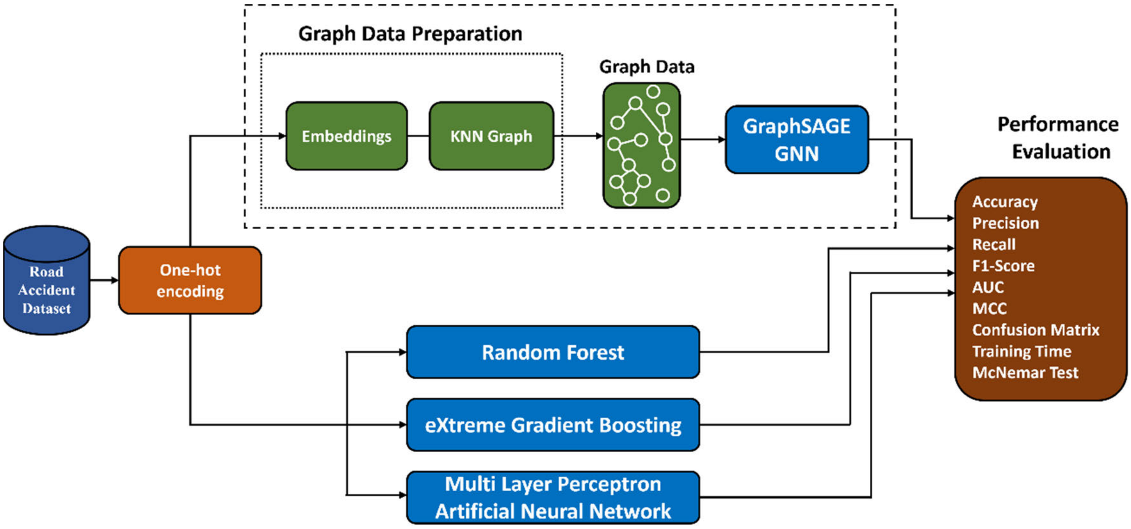


FIGURE 5. The experimental setup of optimized prediction models.

TABLE 3. Configuration for hyperparameters optimization using GridSearchCV.

Models	Parameters for hyperparameters
GNN	'embedding_dim': [**40**,60,80], 'hidden_layer': [**32**, 48, 64], 'number_of_layers': [3,4,**5**], 'drop_rate': [0, **0.1**, 0.2], 'num_neighbors': [40,**50**,60], 'mode_value': ['**distance**', 'connectivity'], 'metric_value': ['**manhattan**', 'cityblock', 'minkowski']
Random Forest	'n_estimators': [100, 300, **500**, 700], 'max_depth': [None, 10, **30**], 'min_samples_split': [2, 7, **10**], 'max_features': [**'sqrt'**, 'log2'], 'min_samples_leaf': [**4**, 10, 17, 20], 'criterion': ['gini', **'entropy'**]
XGBoost	'n_estimators': [100, 300, 500, **700**], 'learning_rate': [**0.01**, 0.1, 0.2], 'max_depth': [3, 5, **7**], 'min_child_weight': [**1**, 2, 3], 'gamma': [0, 0.1, **0.2**], 'subsample': [0.8, 0.9, **1.0**], 'colsample_bytree': [**0.5**, 0.7, 1.0]
ANN	'hidden_layer_sizes': [(100,), (50,),(25,),(100, 100),(100, 50), (100, 25),(50, 50),(50, 25),(25, 25),(100, 100, 100), (100, 100, 50),(100, 100, 25),(100, 50, 50), (100, 50, 25),(100, 25, 25),(50, 50, 50),(50, 50, 25),**(50, 25, 25)**,(25, 25, 25)], 'activation': ['relu', **'tanh'**], 'solver': [**'sgd'**, 'adam'], 'alpha': [0.0001, 0.001, **0.01**], 'learning_rate': [**'constant'**, 'invscaling', 'adaptive'],

incorporates various graph-related functions, optimization techniques, loss functions, and activation functions. These include the SAGEConv layer, dropout layer, and the ReLU activation function. SAGEConv is an improved and advanced version of GraphSAGE [73]. SAGEConv enables the simultaneous learning of both the topological structure of each node's neighborhood and the distribution of node features within that neighborhood [74]. It achieves this enhancement by using a more powerful convolutional operator, allowing it to capture intricate features. SAGEConv also differs from GraphSAGE in terms of its aggregation method, where SAGEConv considers node degrees and computes the aggregate representation as the normalized average of neighbor representations. This approach enables SAGEConv to capture finer graph structure details, leading to a more comprehensive representation.

The Google Colab platform served as the platform for configuring the coding environment, while the Python programming language was employed to create custom code, leveraging libraries like NumPy, Pandas, PyTorch, and PyG, among others. Embeddings were generated using the PCA functionality of the scikit-learn library. Graph data was then constructed using the kneighbors_graph of the sklearn neighbors class and the torch_geometric data method. The dataset was randomly divided into 70% training set, with 30% evenly split between validation and test sets. GridSearchCV was employed to optimize several parameters, as discussed previously.

In this section, we demonstrated the results of research that compared how well all machine learning models, GraphSAGE, RF, XGBoost, and ANN, predicted the severity of crash injuries. These models were evaluated based on metrics, including accuracy, precision, recall, and F1-score, and the results were presented in Table 4. As indicated, the GraphSAGE model achieved the highest accuracy of 85.55%, followed by XGBoost with an accuracy of 83.36%, and ANN and RF with 83.27% and 83.18%, respectively. While comparing the results related to other performance metrics, it was observed that the GraphSAGE achieved a better recall of 0.777 as compared to other models, indicating its ability to identify severe accidents better than the other two models. It also indicated that the GraphSAGE model helped in minimizing false negative (FN) cases where severe cases could have incorrectly been classified as non-severe. On the other hand, it also exhibited a precision of 0.915, which was better than other models, including the RF precision value of 0.886, the XGBoost precision value of 0.900, and the MLP value of 0.875, suggesting that the GNN model also helped in

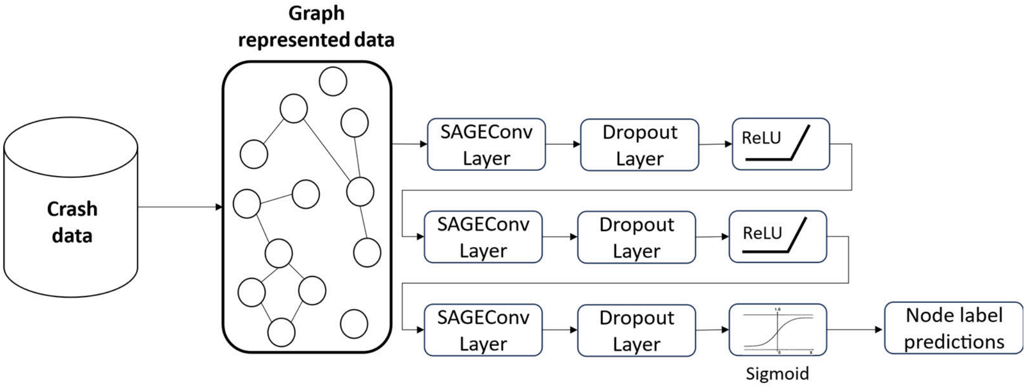


FIGURE 6. Graph neural networks model architecture.

TABLE 4. Performance comparison of GNN (GraphSAGE), RF, XGBoost, and ANN models.

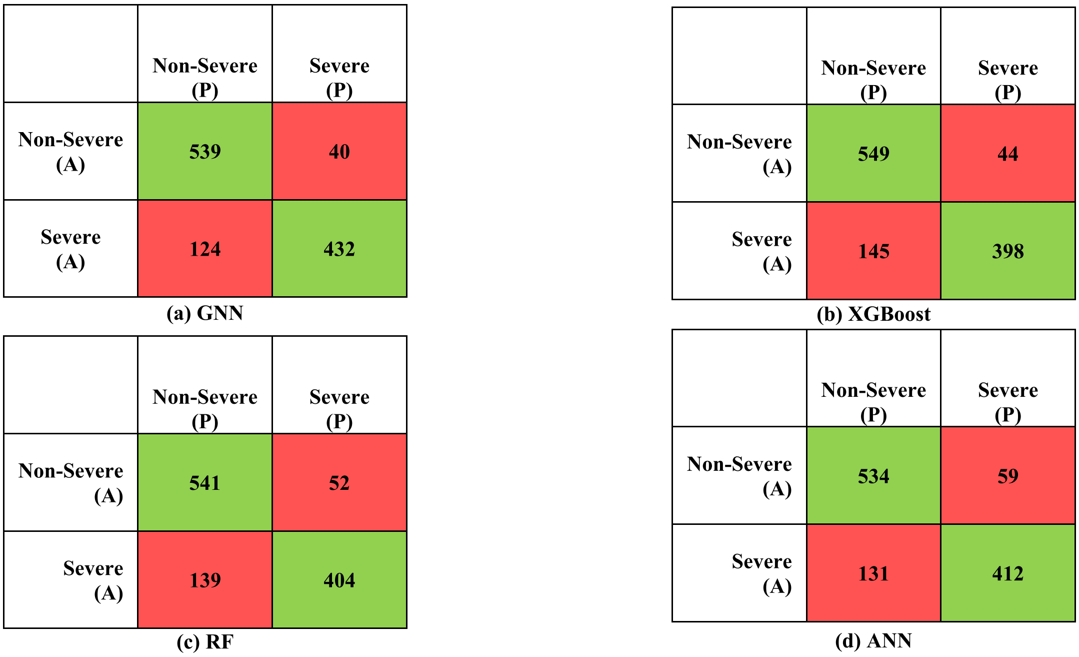
Models	Accuracy	Precision	TPR or Recall	FPR	F1-score	AUC	MCC
GNN	85.55%	0.915	0.777	0.069	0.841	0.902	0.717
XGBoost	83.36%	0.900	0.733	0.074	0.808	0.907	0.674
RF	83.18%	0.886	0.744	0.088	0.809	0.890	0.669
MLP(ANN)	83.27%	0.875	0.759	0.099	0.813	0.899	0.668

minimizing the non-severe cases as severe (i.e., FP) compared to other models. In other words, the GraphSAGE model improved both precision and recall values, thus identifying severe as well as non-severe cases more effectively. It was also found that XGBoost results were comparable to those of RF in identifying severe injury crashes.

To further explore the performance of the proposed GraphSAGE model, its confusion matrix was critically analyzed and compared with the confusion matrices of the XGBoost, RF, and ANN models, as shown in Fig. 7. The confusion matrix helped in identifying model performance in terms of its capabilities in predicting TP, TN, FP, and FN predictions. The GraphSAGE model surpassed other models in terms of classification performance. In the testing dataset, out of 1135 crash records, 432 records were accurately classified as ''severe'' accidents. While analyzing confusion matrices, it was also observed that the GNN model showed improvement in TP as compared to other models, indicating that the GNN model was a good predictor of severe cases. It was also found that its TN values were comparable to other models, although XGBoost performed a little better in this case. It was found that all models had similar capabilities for identifying non-severe cases. Similarly, it was noted that TP or TN values got the gain by lowering FP and FN values. To address this concern and to verify which model performed well overall, we also computed MCC scores for all models. The results indicated that the GNN model had better overall performance as compared to other models. This suggested that the GNN model showed the capability to make correct predictions for both the majority of severe and non-severe crashes.

The GraphSAGE model also had a lower FPR despite misclassifying some crash records compared to the other models. The FPR measures how often a model incorrectly labels a negative sample as positive. With the lowest FPR value, GNN model outperformed other models, indicating its ability to minimize false positives. This suggested that the model can assist the emergency response team in focusing resources on severe cases and prioritizing their responses effectively. The GNN model also surpassed the other models in terms of TPR, which is equivalent to recall, with a value of around 0.777. Similarly, AUC values were also presented in the table, identifying the overall effectiveness of the GNN model in identifying severe cases. The GNN model showed an AUC value of 0.902, which was slightly lower than the XGBoost value of 0.907, indicating that both models had similar performance in discriminating between severe and non-severe crashes.

Since both the GNN and MLP (ANN) models belonged to the class of neural networks, we also compared their performances in terms of performance losses, as shown in Fig. 8. It was found that the GNN model took less time to converge, indicating the efficiency of the GNN architecture in capturing complex patterns in a lesser duration.



	Non-Severe (P)	Severe (P)
Non-Severe (A)	539	40
Severe (A)	124	432

(a) GNN

	Non-Severe (P)	Severe (P)
Non-Severe (A)	549	44
Severe (A)	145	398

(b) XGBoost

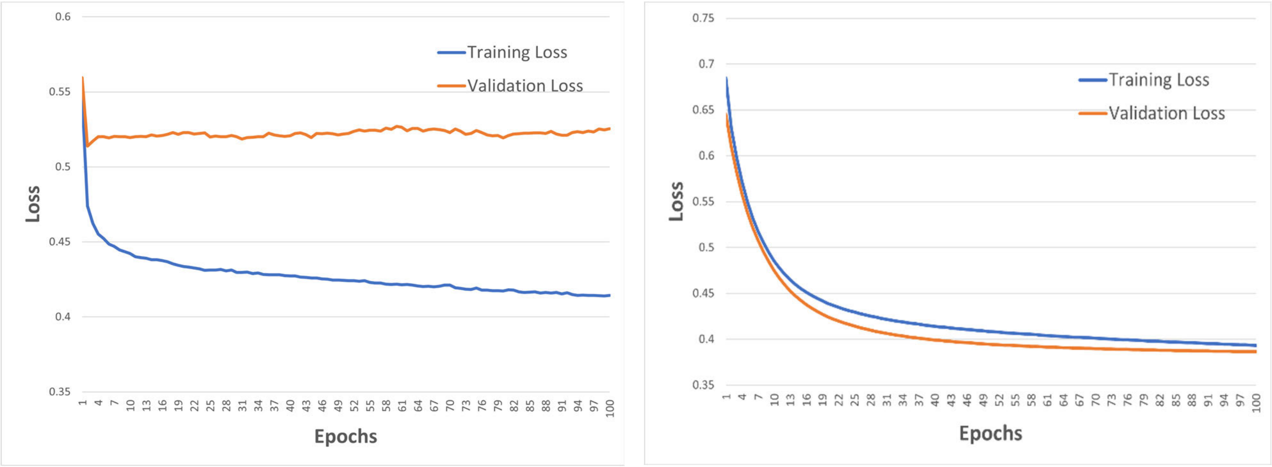
	Non-Severe (P)	Severe (P)
Non-Severe (A)	541	52
Severe (A)	139	404

(c) RF

	Non-Severe (P)	Severe (P)
Non-Severe (A)	534	59
Severe (A)	131	412

(d) ANN

FIGURE 7. Confusion matrix for crash injury severity for different models.



(a) GNN Model

(b) MLP (ANN)

FIGURE 8. Performance losses for the GNN and MLP (ANN) models.

Cross-validation with 10-fold was also used to evaluate the overall performance of all models. The average overall accuracy for each model was 83.40%, 82.80%, 81.52%, and 82.18% for the GNN, XGBoost, RF and MLP (ANN) models, respectively. In addition, precision, recall, and F1 score were also presented. The results indicated that the GNN model performed well overall in terms of overall effectiveness in predicting severe cases. Its F1-score, being the highest among the compared models, suggested that the GNN model achieved a good balance between precision and recall, implying that the GNN model exhibited better performance in correctly classifying severe cases while minimizing both false positives and false negatives.

We also utilized the McNemar test to evaluate the statistical significance of the performances of our models. The purpose of the test was to evaluate the null hypothesis and determine if the two comparison models exhibited a similar level of disagreement with the test predictions. We used the Mcnemar (Statsmodels) python library to analyze and compare the performance of the GNN models with other models. The GNN, XGBoost, and MLP (ANN) all had similar error rates, indicating that there was no significant difference between them ($p > 0.05$). However, the GNN and RF models did show different error rates, suggesting that there was a significant difference between them ($p <= 0.05$).

TABLE 5. Cross validation performance measures of all models.

Models	Accuracy	Precision	Recall	F1-Score
GNN	83.40%	0.871	0.772	0.816
XGBoost	82.80%	0.874	0.754	0.809
RF	81.52%	0.899	0.697	0.786
MLP (ANN)	82.18%	0.868	0.746	0.802

TABLE 6. Memory usage and computational overhead comparison for models.

Models	Model Size*	Data	Training Time (seconds)
GNN with embedding	156 KB	7.3 MB	**24.54
XGBoost	284 KB	2.3 MB	2.07
RF	17 MB	2.3 MB	6.12
MLP(ANN)	218 KB	2.3 MB	16.58

* Pickled model fitted on training set
** Early stopping mechanism was implemented.

In terms of computation cost, both the GNN and MLP (ANN) models took more time during training due to the nature of their architecture and complexity as compared to RF and XGBoost, which are tree-based ensemble models. In addition, in terms of memory footprint, the graph data required more memory space, but the trained GNN model itself took less memory footprint as compared to all other models.

VI. CONCLUSION

In this study, we used the GraphSAGE model to predict the severity of road crash injuries. Our goal was to determine the severity of these injuries and categorize them as either severe or non-severe. To the best of our knowledge, this was the first attempt to investigate the viability of using GNN to predict the severity of road crash injuries. Specifically, we took advantage of GraphSAGE's ability to analyze graph-based data quickly and effectively in the context of binary classification. The rationale behind the research was to exploit relationships among various crash records to uncover any hidden patterns that traditional ML models might overlook. The ability to predict an accident's severity would be greatly aided if these underlying patterns could be identified. This would aid in both emergency preparation and accident prevention. Furthermore, we presented an approach in this study for creating graphs that used a kNN graph to create edges between records. The graph was created using a balanced crash severity dataset from the UK, which included spatial and temporal factors, road configuration, vehicle characteristics, and environmental conditions.

We divided the dataset into training, validation, and testing subsets, with splits of 70%, 15%, and 15%, respectively. We evaluated GraphSAGE based GNN and other machine and deep learning models using metrics such as accuracy, precision, recall, F1-score, MCC, AUC, and the confusion matrix. The testing accuracy for the GraphSAGE model reached 83.55%, surpassing the performance of other models, and similar results were observed for other performance metrics. These results showed that our proposed embedding-based GraphSAGE model was better than other commonly used ensemble models for predicting the severity of crash injuries.

Compared to traditional models, the proposed GNN model can leverage graph data to capture relationships and dependencies within accident data (which is structured in graph format). By representing crashes as nodes and their relationships as edges of a graph, the proposed GNN model can effectively learn both nodes' features as well as their relationships. Traditional models usually identify similar data points based on feature importance towards the target variable [79], whereas the proposed GNN model, due to its architectural design, recognizes relationships among data points, thus considering both the edge connectivity within the graph and the information embedded in the nodes. The captured information can then be used to identify similar data observations and to design and implement solutions, such as changes to road infrastructure, improved traffic signage, and traffic awareness, to enhance road safety.

VII. LIMITATION AND FUTURE WORK

Initially, the dataset exhibited an imbalance in its nature. In this study, the random undersampling method was utilized to achieve a balanced class distribution and to evaluate prediction models. However, the random undersampling method may lead to the removal of important data observations that can cause underfitting of the model. Current imbalance handling techniques are not well-suited for GNNs in their present form. While techniques like SMOTE exist, their applicability to graph data necessitates thorough evaluation in future research. To mitigate this limitation, we plan to explore different sampling techniques, such as cost-sensitive methods, to address the challenges of class imbalance. In future work, we intend to investigate the effect of different sampling techniques on the performance of GNN in crash severity modeling. Looking ahead, our future endeavors will also focus on exploring the utilization of imbalanced multiclass datasets for predicting crash severity using GNNs.

The GNN model is considered a black box model and therefore lacks the explanatory capability necessary to understand its decision-making process. Our future work also aims to address this limitation by incorporating interpretability features into our model architecture to gain deeper insights into both the graph architecture and the contributing factors that influence the performance of our model.

ACKNOWLEDGMENT

The authors would like to acknowledge the support of Universiti Putra Malaysia.

DECLARATIONS

Conflict of interest: The authors declare that they have no conflict of interest.

REFERENCES

[1] S. Ahmed, M. A. Hossain, S. K. Ray, M. M. I. Bhuiyan, and S. R. Sabuj, "A study on road accident prediction and contributing factors using explainable machine learning models: Analysis and performance," *Transp. Res. Interdiscipl. Perspect.*, vol. 19, May 2023, Art. no. 100814.

[2] S. Chen, S. Zhang, Y. Xing, and J. Lu, "Identifying the factors contributing to the severity of truck-involved crashes in Shanghai river-crossing tunnel," *Int. J. Environ. Res. Public Health*, vol. 17, no. 9, p. 3155, May 2020.

[3] J. Li, Q. Liu, and Y. Sang, "Several issues about urbanization and urban safety," *Proc. Eng.*, vol. 43, pp. 615–621, Jan. 2012.

[4] S. Naznin, P. H. Sumayya, L. S. Panackel, S. Zaviar, and S. Babu, "Accident prediction modelling and crash scene investigation," in *Proc. Int. Conf. Struct. Eng. Construct. Manage.*, 2022, pp. 1121–1138.

[5] P. Li and M. Abdel-Aty, "A hybrid machine learning model for predicting real-time secondary crash likelihood," *Accident Anal. Prevention*, vol. 165, Feb. 2022, Art. no. 106504.

[6] L. Zheng and T. Sayed, "A novel approach for real time crash prediction at signalized intersections," *Transp. Res. C, Emerg. Technol.*, vol. 117, Aug. 2020, Art. no. 102683.

[7] T. Bhowmik, S. Yasmin, and N. Eluru, "A new econometric approach for modeling several count variables: A case study of crash frequency analysis by crash type and severity," *Transp. Res. B, Methodol.*, vol. 153, pp. 172–203, Nov. 2021.

[8] J. Ma and K. Kockelman, "Crash frequency and severity modeling using clustered data from Washington state," in *Proc. IEEE Intell. Transp. Syst. Conf.*, Jun. 2006, pp. 1621–1626.

[9] F. Tang, X. Fu, M. Cai, Y. Lu, and S. Zhong, "Investigation of the factors influencing the crash frequency in expressway tunnels: Considering excess zero observations and unobserved heterogeneity," *IEEE Access*, vol. 9, pp. 58549–58565, 2021.

[10] C. Ma, W. Hao, W. Xiang, and W. Yan, "The impact of aggressive driving behavior on driver-injury severity at highway-rail grade crossings accidents," *J. Adv. Transp.*, vol. 2018, pp. 1–10, Oct. 2018.

[11] R. Mesa-Arango, V. G. Valencia-Alaix, R. A. Pineda-Mendez, and T. Eissa, "Influence of socioeconomic conditions on crash injury severity for an urban area in a developing country," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 2672, no. 31, pp. 41–53, Dec. 2018.

[12] C. Chen, G. Zhang, H. Huang, J. Wang, and R. A. Tarefder, "Examining driver injury severity outcomes in rural non-interstate roadway crashes using a hierarchical ordered logit model," *Accident Anal. Prevention*, vol. 96, pp. 79–87, Nov. 2016.

[13] S. M. Rifaat and H. C. Chin, "Accident severity analysis using ordered probit model," *J. Adv. Transp.*, vol. 41, no. 1, pp. 91–114, Dec. 2007.

[14] P. Liu and W. Fan, "Modeling head-on crash severity on NCDOT freeways: A mixed logit model approach," *Can. J. Civil Eng.*, vol. 46, no. 4, pp. 322–328, Apr. 2019.

[15] G. Azimi, A. Rahimi, H. Asgari, and X. Jin, "Severity analysis for large truck rollover crashes using a random parameter ordered logit model," *Accident Anal. Prevention*, vol. 135, Feb. 2020, Art. no. 105355.

[16] X. Shao, X. Ma, F. Chen, M. Song, X. Pan, and K. You, "A random parameters ordered probit analysis of injury severity in truck involved rear-end collisions," *Int. J. Environ. Res. Public Health*, vol. 17, no. 2, p. 395, Jan. 2020.

[17] S. Xie, X. Ji, W. Yang, R. Fang, and J. Hao, "Exploring risk factors with crash severity on China two-lane rural roads using a random-parameter ordered probit model," *J. Adv. Transp.*, vol. 2020, pp. 1–14, Dec. 2020.

[18] B. Kumeda, F. Zhang, F. Zhou, S. Hussain, A. Almasri, and M. Assefa, "Classification of road traffic accident data using machine learning algorithms," in *Proc. IEEE 11th Int. Conf. Commun. Softw. Netw. (ICCSN)*, Jun. 2019, pp. 682–687.

[19] J. Lee, T. Yoon, S. Kwon, and J. Lee, "Model evaluation for forecasting traffic accident severity in rainy seasons using machine learning algorithms: Seoul city study," *Appl. Sci.*, vol. 10, no. 1, p. 129, Dec. 2019.

[20] I. Aldhari, M. Almoshaogeh, A. Jamal, F. Alharbi, M. Alinizzi, and H. Haider, "Severity prediction of highway crashes in Saudi Arabia using machine learning techniques," *Appl. Sci.*, vol. 13, no. 1, p. 233, Dec. 2022.

[21] H. Bhuiyan, J. Ara, K. M. Hasib, M. I. H. Sourav, F. B. Karim, C. Sik-Lanyi, G. Governatori, A. Rakotonirainy, and S. Yasmin, "Crash severity analysis and risk factors identification based on an alternate data source: A case study of developing country," *Sci. Rep.*, vol. 12, no. 1, p. 21243, Dec. 2022.

[22] B. Pradhan and M. I. Sameen, "Modeling traffic accident severity using neural networks and support vector machines," in *Laser Scanning Systems in Highway and Safety Assessment* (Advances in Science, Technology & Innovation). Springer, 2020. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-030-10374-3_9

[23] H. Jeong, I. Kim, K. Han, and J. Kim, "Comprehensive analysis of traffic accidents in seoul: Major factors and types affecting injury severity," *Appl. Sci.*, vol. 12, no. 4, p. 1790, Feb. 2022.

[24] T. Jiang, W. Hu, Y. Liu, and L. Xiao, "Capturing and exploring unobserved heterogeneity in traffic crash injuries in China: A deep learning approach," in *Proc. IEEE 7th Int. Conf. Intell. Transp. Eng. (ICITE)*, Nov. 2022, pp. 331–336.

[25] G. Singh, M. Pal, Y. Yadav, and T. Singla, "Deep neural network-based predictive modeling of road accidents," *Neural Comput. Appl.*, vol. 32, no. 16, pp. 12417–12426, Aug. 2020.

[26] C. Seger, "An investigation of categorical variable encoding techniques in machine learning: Binary versus one-hot and feature hashing," Ph.D. dissertation, KTH Roy. Inst. Technol., Stockholm, Sweden, 2018.

[27] S. Kang, "K-nearest neighbor learning with graph neural networks," *Mathematics*, vol. 9, no. 8, p. 830, Apr. 2021.

[28] L. Berton and A. De Andrade Lopes, "Graph construction for semi-supervised learning," in *Proc. 24th Int. Conf. Artif. Intell.*, Buenos Aires, Argentina, 2015, pp. 4343–4344.

[29] L. Yingfan, C. Hong, and C. Jiangtao, "Revisiting k-nearest neighbor graph construction on high-dimensional data : Experiments and analyses," 2021, *arXiv:2112.02234*.

[30] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1025–1035.

[31] J. Chiehen Liao and C.-T. Li, "TabGSL: Graph structure learning for tabular data prediction," 2023, *arXiv:2305.15843*.

[32] S. Roy, D. Sarkar, S. Malakar, and R. Sarkar, "Offline signature verification system: A graph neural network based approach," *J. Ambient Intell. Humanized Comput.*, vol. 14, no. 7, pp. 8219–8229, Jul. 2023.

[33] M. K. Dahouda and I. Joe, "A deep-learned embedding technique for categorical features encoding," *IEEE Access*, vol. 9, pp. 114381–114391, 2021.

[34] L. Tran, L. Fan, and C. Shahabi, "Clustering mixed-type data with correlation-preserving embedding," in *Proc. Int. Conf. Database Syst. Adv. Appl.*, vol. 26. Taipei, Taiwan, 2021, pp. 342–358.

[35] L. Bai and J. Liang, "A categorical data clustering framework on graph representation," *Pattern Recognit.*, vol. 128, Aug. 2022, Art. no. 108694.

[36] S. Mumtaz and M. Giese, "Hierarchy-based semantic embeddings for single-valued & multi-valued categorical variables," *J. Intell. Inf. Syst.*, vol. 58, pp. 1–28, Apr. 2022.

[37] W. Wang, Y. Han, S. Bromuri, and M. Dumontier, "Semantic correlation graph embedding," in *Proc. IEEE Int. Conf. Fuzzy Syst. (FUZZ-IEEE)*, Jul. 2022, pp. 1–10.

[38] Z. Liang, S. Ji, Q. Li, S. Hu, and Y. Yu, "An attribute-weighted isometric embedding method for categorical encoding on mixed data," *Appl. Intell.*, vol. 53, no. 22, pp. 26472–26496, Nov. 2023.

[39] A. Sheshmani and Y.-Z. You, "Categorical representation learning: Morphism is all you need," *Mach. Learn., Sci. Technol.*, vol. 3, no. 1, Mar. 2022, Art. no. 015016.

[40] R. Ramirez, Y.-C. Chiu, S. Zhang, J. Ramirez, Y. Chen, Y. Huang, and Y.-F. Jin, "Prediction and interpretation of cancer survival using graph convolution neural networks," *Methods*, vol. 192, pp. 120–130, Aug. 2021.

[41] P. K. Rao, S. Chatterjee, K. Nagaraju, S. B. Khan, A. Almusharraf, and A. I. Alharbi, "Fusion of graph and tabular deep learning models for predicting chronic kidney disease," *Diagnostics*, vol. 13, no. 12, p. 1981, Jun. 2023.

[42] M. T. Do, N. Park, and K. Shin, "Two-stage training of graph neural networks for graph classification," *Neural Process. Lett.*, vol. 55, no. 3, pp. 2799–2823, Jun. 2023.

[43] S. Ivanov and L. Prokhorenkova, "Boost then convolve: Gradient boosting meets graph neural networks," 2021, *arXiv:2101.08543*.

[44] R. Rusli, M. M. Haque, M. Saifuzzaman, and M. King, "Crash severity along rural mountainous highways in malaysia: An application of a combined decision tree and logistic regression model," *Traffic Injury Prevention*, vol. 19, no. 7, pp. 741–748, Oct. 2018.

[45] F. Chang, P. Xu, H. Zhou, A. H. S. Chan, and H. Huang, "Investigating injury severities of motorcycle riders: A two-step method integrating latent class cluster analysis and random parameters logit model," *Accident Anal. Prevention*, vol. 131, pp. 316–326, Oct. 2019.

[46] J. Wang, H. Huang, P. Xu, S. Xie, and S. C. Wong, "Random parameter probit models to analyze pedestrian red-light violations and injury severity in pedestrian–motor vehicle crashes at signalized crossings," *J. Transp. Saf. Secur.*, vol. 12, no. 6, pp. 818–837, Jul. 2020.

[47] K. Raja, K. Kaliyaperumal, L. Velmurugan, and S. Thanappan, "Forecasting road traffic accident using deep artificial neural network approach in case of Oromia special zone," *Soft Comput.*, vol. 27, no. 21, pp. 16179–16199, Nov. 2023.

[48] M. E. Shaik, M. M. Islam, and Q. S. Hossain, "A review on neural network techniques for the prediction of road traffic accident severity," *Asian Transp. Stud.*, vol. 7, 2021, Art. no. 100040.

[49] M. Sameen and B. Pradhan, "Severity prediction of traffic accidents with recurrent neural networks," *Appl. Sci.*, vol. 7, no. 6, p. 476, Jun. 2017.

[50] B. Dimitrijevic, R. Asadi, and L. Spasovic, "Application of hybrid support vector machine models in analysis of work zone crash injury severity," *Transp. Res. Interdiscipl. Perspect.*, vol. 19, May 2023, Art. no. 100801.

[51] S. I. Mohammadpour, M. Khedmati, and M. J. H. Zada, "Classification of truck-involved crash severity: Dealing with missing, imbalanced, and high dimensional safety data," *PLoS ONE*, vol. 18, no. 3, Mar. 2023, Art. no. e0281901.

[52] L. Pérez-Sala, M. Curado, L. Tortosa, and J. F. Vicent, "Deep learning model of convolutional neural networks powered by a genetic algorithm for prevention of traffic accidents severity," *Chaos, Solitons Fractals*, vol. 169, Apr. 2023, Art. no. 113245.

[53] C. Panda, A. K. Mishra, A. K. Dash, and H. Nawab, "Predicting and explaining severity of road accident using artificial intelligence techniques, SHAP and feature analysis," *Int. J. Crashworthiness*, vol. 28, no. 2, pp. 186–201, Mar. 2023.

[54] M. Megnidio-Tchoukouegno and J. A. Adedeji, "Machine learning for road traffic accident improvement and environmental resource management in the transportation sector," *Sustainability*, vol. 15, no. 3, p. 2014, Jan. 2023.

[55] J. Niyogisubizo, L. Liao, Q. Sun, E. Nziyumva, Y. Wang, L. Luo, S. Lai, and E. Murwanashyaka, "Predicting crash injury severity in smart cities: A novel computational approach with wide and deep learning model," *Int. J. Intell. Transp. Syst. Res.*, vol. 21, no. 1, pp. 240–258, Apr. 2023.

[56] J. Niyogisubizo, L. Liao, F. Zou, G. Han, E. Nziyumva, B. Li, and Y. Lin, "Predicting traffic crash severity using hybrid of balanced bagging classification and light gradient boosting machine," *Intell. Data Anal.*, vol. 27, no. 1, pp. 79–101, Jan. 2023.

[57] A. Azhar, N. M. Ariff, M. A. A. Bakar, and A. Roslan, "Classification of driver injury severity for accidents involving heavy vehicles with decision tree and random forest," *Sustainability*, vol. 14, no. 7, p. 4101, Mar. 2022.

[58] Department for Transport, U.K. (2023). *Road Safety Data*. U.K. Open Data. Accessed: Jul. 16, 2023. [Online]. Available: https://www.data.gov.uk/dataset/cb7ae6f0-4be6-4935-9277-47e5ce24a11f/road-safety-data

[59] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, pp. 5–32, Oct. 2001.

[60] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2016, pp. 785–794.

[61] X. Feng, G. Ma, S.-F. Su, C. Huang, M. K. Boswell, and P. Xue, "A multi-layer perceptron approach for accelerated wave forecasting in Lake Michigan," *Ocean Eng.*, vol. 211, Sep. 2020, Art. no. 107526.

[62] D. Zhang, L. Qian, B. Mao, C. Huang, B. Huang, and Y. Si, "A data-driven design for fault detection of wind turbines using random forests and XGboost," *IEEE Access*, vol. 6, pp. 21020–21031, 2018.

[63] Y. Javed and N. Rajabi, "Multi-layer perceptron artificial neural network based IoT botnet traffic classification," in *Proc. Future Technol. Conf. (FTC)*, vol. 1, 2019, pp. 973–984.

[64] E. K. Sahin, "Assessing the predictive capability of ensemble tree methods for landslide susceptibility mapping using XGBoost, gradient boosting machine, and random forest," *Social Netw. Appl. Sci.*, vol. 2, no. 7, p. 1308, Jul. 2020.

[65] W. L. Hamilton, *Graph Representation Learning*. San Rafael, CA, USA: Morgan & Claypool, 2020.

[66] M. McPherson, L. Smith-Lovin, and J. M. Cook, "Birds of a feather: Homophily in social networks," *Annu. Rev. Sociol.*, vol. 27, no. 1, pp. 415–444, Aug. 2001.

[67] D. Zhou, O. Bousquet, T. Lal, J. Weston, and B. Schölkopf, "Learning with local and global consistency," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 16, 2003, pp. 321–328.

[68] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," 2016, *arXiv:1609.02907*.

[69] P. Velič ković, G. Cucurull, A. Casanova, A. Romero, P. Lió, and Y. Bengio, "Graph attention networks," 2017, *arXiv:1710.10903*.

[70] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, and M. Blondel, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Nov. 2011.

[71] S. Liu, P.-T. Bremer, J. J. Thiagarajan, V. Srikumar, B. Wang, Y. Livnat, and V. Pascucci, "Visual exploration of semantic relationships in neural word embeddings," *IEEE Trans. Vis. Comput. Graphics*, vol. 24, no. 1, pp. 553–562, Jan. 2018.

[72] M. Fey and J. Eric Lenssen, "Fast graph representation learning with PyTorch geometric," 2019, *arXiv:1903.02428*.

[73] S. Sheikh. (Jan. 2023). *Exploring SageConv: A Powerful Graph Neural Network Architecture*. Medium. Accessed: Jun. 28, 2023. [Online]. Available: https://medium.com/@sheikh.sahil12299/exploring-sageconv-a-powerful-graph-neural-network-architecture-44b7974b1fe0

[74] U. Veleiro, J. de la Fuente, G. Serrano, M. Pizurica, M. Casals, A. Pineda-Lucena, S. Vicent, I. Ochoa, O. Gevaert, and M. Hernaez, "GENNIUS: An ultrafast drug-target interaction inference method based on graph neural networks," *Bioinformatics*, vol. 40, no. 1, 2024, Art. no. btad774.

[75] K. Syama, J. A. A. Jothi, and N. Khanna, "Automatic disease prediction from human gut metagenomic data using boosting GraphSAGE," *BMC Bioinf.*, vol. 24, no. 1, p. 126, Mar. 2023.

[76] I. Arkoudi, R. Krueger, C. L. Azevedo, and F. C. Pereira, "Combining discrete choice models and neural networks through embeddings: Formulation, interpretability and performance," *Transp. Res. B, Methodol.*, vol. 175, Sep. 2023, Art. no. 102783.

[77] D. Chicco, M. J. Warrens, and G. Jurman, "The Matthews correlation coefficient (MCC) is more informative than Cohen's Kappa and brier score in binary classification assessment," *IEEE Access*, vol. 9, pp. 78368–78381, 2021.

[78] D. Chicco and G. Jurman, "The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation," *BMC Genomics*, vol. 21, no. 1, pp. 1–13, Dec. 2020.

[79] A. Jamal, M. Zahid, M. Tauhidur Rahman, H. M. Al-Ahmadi, M. Almoshaogeh, D. Farooq, and M. Ahmad, "Injury severity prediction of traffic crashes with ensemble machine learning techniques: A comparative study," *Int. J. Injury Control Saf. Promotion*, vol. 28, no. 4, pp. 408–427, Oct. 2021.

[80] J. M. Johnson and T. M. Khoshgoftaar, "Semantic embeddings for medical providers and fraud detection," in *Proc. IEEE 21st Int. Conf. Inf. Reuse Integr. Data Sci. (IRI)*, Aug. 2020, pp. 224–230.

[81] K. Xu, C. Li, Y. Tian, T. Sonobe, K.-I. Kawarabayashi, and S. Jegelka, "Representation learning on graphs with jumping knowledge networks," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 5453–5462.

[82] T. G. Dietterich, "Approximate statistical tests for comparing supervised classification learning algorithms," *Neural Comput.*, vol. 10, no. 7, pp. 1895–1923, Oct. 1998.

[83] G. Sambasivam, J. Amudhavel, and G. Sathya, "A predictive performance analysis of vitamin D deficiency severity using machine learning methods," *IEEE Access*, vol. 8, pp. 109492–109507, 2020.

[84] I. H. Witten, E. Frank, M. A. Hall, and C. J. Pal, *Data Mining: Practical Machine Learning Tools and Techniques*. San Mateo, CA, USA: Morgan Kaufmann, 2016.

KARIM A. SATTAR received the B.E. degree in computer and information systems engineering from NED University, Karachi, Pakistan, and the M.Sc. degree in information and computer science from the King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia. He is currently pursuing the Ph.D. degree with Universiti Putra Malaysia (UPM). His research interests include smart traffic management, accident severity prediction, explainable artificial intelligence (AI), ethical AI practices, business intelligence, and project management.

ISKANDAR ISHAK received the bachelor's degree in information technology from Universiti Tenaga Nasional, Malaysia, the M.Tech. degree in information technology from the Royal Melbourne Institute of Technology, Australia, and the Ph.D. degree in computer science from Universiti Teknologi Malaysia. He is currently a Senior Lecturer with Universiti Putra Malaysia. His research interests include database systems, big data, and data analytics.

LILLY SURIANI AFFENDEY received the bachelor's degree in computer science from Universiti Putra Malaysia (UPM), in 1991, the M.Sc. degree in computing from the University of Bradford, U.K., in 1994, and the Ph.D. degree from UPM, in 2007. She is currently an Associate Professor with the Department of Computer Science, Faculty of Computer Science and Information Technology, UPM. Her current research interests include multimedia databases, video content-based retrieval, data science, and big data analytics.

SITI NURULAIN BINTI MOHD RUM received the bachelor's degree from Universiti Teknologi Malaysia (UTM), Malaysia, and the master's and Ph.D. degrees in computer science from the University of Malaya (UM), in 2012 and 2017, respectively. She was an IT Practitioner for 15 years. She is currently a Senior Lecturer with the Department of Computer Science, Faculty of Computer Science, and Information Technology, Universiti Putra Malaysia (UPM). She has published a number of journals and presented research works at international conferences. Her research interests include artificial intelligence, database processing, data science, semantic web, and social media analytics.