Republic of Iraq
Ministry of Higher Education and Scientific Research
University of Babylon
College of Information Technology
Software Department

Accidents Area Detection using Deep Graph Learning Based on Naturalistic Driving Data

A Thesis

Submitted to the Council of the College of Information Technology University of Babylon, in Partial Fulfillment of the Requirements for the Degree of Master in Information Technology/Software

By

Meekat Hasan Ali Hussain

Supervised By

Dr. Wadhah R. Baiee

2021 A.D. **1443 A.H.**

بِسْمِ ٱللَّهِ ٱلرَّحْمَٰنِ ٱلرَّحِيمِ

﴿ وَلَقَدْ آتَيْنَا دَاوُودَ وَسُلَيْمَانَ عِلْمًا وَقَالَا الْحَمْدُ لِلَّهِ الَّذِي
فَضَّلَنَا عَلَى كَثِيرٍ مِنْ عِبَادِهِ الْمُؤْمِنِينَ ﴾

صدق الله العلي العظيم

سورة النمل الآية 15

Declaration

I hereby declare that this dissertation, submitted to the University of Babylon in partial fulfillment of requirements for the degree of Master in Information Technology \ Software, has not been submitted as an exercise for a similar degree at any other University. I also certify that this work described here is entirely my own except for experts and summaries whose sources are appropriately cited in the references.

Signature:

Name:

Date: /11/2021

Supervisor Certification

I certify that this thesis was prepared under my supervision at the Department of Software / Collage of Information Technology / Babylon University, by Meekat Hasan Ali as a partial fulfillment of the requirements for the degree of Master in Information Technology.

Signature:

Name: Dr. Wadhah R. Baiee

Date: / 11/ 2021

The Head of the Department Certification

In view of the available recommendation, we forward this thesis for debate by the examining committee.

Signature:

Name: Dr. Ahmed Saleem Al-Saffar

Date: /11 / 2021

Acknowledgements

All praise be to ALLAH Almighty who enabled me to complete this task successfully and my utmost respect to his last Prophet Mohammad PBUH and to the hero of Islam born in the heart of Kaaba Al-Imam Ali Ibn Abi Talib.

In recognition of those who are credited, I offer my thanks, appreciation, and gratitude to my supervisor Dr. Wadhah R. Baiee for his continuous support as his constructive directions had a great impact on the completion of this work.

I am deeply thankful to those who gave me unconditional support along with my life, my parents, my husband, I have no valuable words to express my gratitude to them.

Last but not least, but most importantly, I would like to thank all the kind, helpful and lovely people who helped me directly or indirectly to complete this work and apologize to them for not being able to mention them by name here, but they are in my heart. Also, a special thanks go to the college of Information Technology staff members for the great co-operation they have offered me.

Meekat Hasan Ali

Dedication

To the savior of mankind...

Al-Imam Al-Mehdi (peace be upon him) Who will fill the earth with justice and equity, after it has been filled with injustice and oppression. The survival of Allah in the earth.

To my father...

Who leads me through the valley of darkness with the light of hope and support.

To my mother...

The strong and gentle soul who taught me to trust Allah, believe in hard work and that so much could be done with little.

To my husband and children...

Who are the secret of my happiness in life.

To my sisters and brothers...

Who are encourages and support me.

Abstract

Road traffic accidents are common incidents that can result in severe injuries and have a direct effect on society's individuals. The focus of traffic safety research has changed from preventing injuries in a collision to countermeasures taken before an accident occurs, in order to avoid its effects altogether. These measures taken need to know the environmental and other factors causing the accident. So, to achieve these requirements and its related analysis; Naturalistic Driving Data (NDD) has emerged as a critical data source with high environmental validity.

In this study we attend to analyze and predict the severity of traffic accident injuries according to Queensland database which consists of 328247 traffic accident records that occurred over a 19-year period (from 2001 to 2019). Every crash has 52 special features that are gathered at the time of the incident. Different filter feature selection methods like (Information-Gain, Gain-Ratio, Relief-f, Chi-Squared) are used to reduce the number of features to be used later in the classification model, to classify five kinds of injury severity.

The proposed system is designed through two manners; the first one adopted Artificial Neural Network (ANN) classifier to achieve accidents severity prediction goal by using resulting features from the previous preprocessing step, the main goal is to raise the classification accuracy, and how we can use classifier results to identify the high accident areas.

The second proposed manner is carried out to improve the classification over ANN classification results. Traffic accident data is represented as a fully connected graph then learning this graph by using the Node2Vec

algorithm. This algorithm is used to extract latent features from graph data. It converts the graph into low dimensional vectors where each accident is represented as a vector of features, then the matrix of node embedding’s represented as an input to the ANN classifier again. The resulted features after using deep graph learning indicated a substantial improvement in the ANN classifier's accuracy; it achieves 95.88%. Compared with the previous accuracy by using resulting features from the feature selection methods, the results were 80.39%.

In order to verify the ANN model's performance, another model is used in this thesis. Convolution Neural Network (CNN) model is suggested here to classify accident severity. Both previous manners for selecting and/or extracting new features are used as the input data to the proposed CNN model. The results show that the features extracted from the deep graph learning with the ANN model have higher accuracy than those obtained from the CNN model, it gives 78.67% by using the same extracted features according to the evaluations that are applied over both classifiers. The reason for giving a lower accuracy is because there is no correlation between accidents, such as the correlation between image pixels.

Table of Contents

List of Algorithms

List of Tables

List of Figures

List of Abbreviations

Abbreviation	Meaning
ANNs	Artificial Neural Networks
BFS	Breadth First Search
CM	Confusion Matrix
CNNs	Convolution Neural networks
DGL	Deep Graph Learning
DL	Deep Learning
DFS	Depth First Search
FN	False Negative
FP	False Positive
FS	Feature Selection
FNNs	Feed-Forward Neural Networks
GR	Gain Ratio
GIS	Geographic Information System
GPS	Global Positioning System
IG	Information Gain
ML	Machine Learning
MAE	Mean Absolute Error
NDD	Naturalistic Driving Data
ReLU	Rectified linear unit
RMSE	Root Mean Square Error
SI	Split Information
Std	Standard deviation
TN	True Negative

Abbreviation	Meaning
TP	True Positive
TPR	True Positive Rate
UTM	Universal Transverse Mercator
WHO	World Health Organization

Thesis Related Publication

- **Name of conference:** International Conference on Communication & Information Technology (ICICT2021).
  - **Paper Title:** Choosing an Appropriate Feature Selection Method to Enhance Feed-Forward ANN
- **Authors:**
  - Msc. Student: Meekat Hasan Ali.
  - Dr.: Wadhah R. Baiee.

College of Information Technology, Babylon University, Iraq

Email: meekat.almusawi@itnet.uobabylon.edu.iq

Email: wadhah.baiee@uobabylon.edu.iq

Chapter one

General Introduction

1.1 Introduction:

Road accidents impose serious problems on society in terms of human, economic, medical and environmental costs. As the World Health Organization (WHO) announced, the total number of road fatalities was approximately 1.25 million in 2016 [1]. The increase in number of vehicles moving on roads accelerated the risk of accidents [2]. In the past several years, the focus of traffic safety research has shifted from injury prevention during a crash to measures taken before a crash, in order to mitigate its effects or avoid it completely. Advanced driver assistance systems, safety elements of autonomous driving and infrastructure design, behavior-based safety (driver training), and policy-making are some of the measures that are being taken. All of these measures need a knowledge of driving behavior [3].

Naturalistic observations, which is called naturalistic driving, is a way that can be applied. It is “a study undertaken to provide insight into driver behavior during every day trips by recording details of the driver, the vehicle and the surroundings through unobtrusive data gathering equipment and without experimental control”[4]. Typically, in study of naturalistic observation passenger cars are the cars that are subject to the test because they are equipped with devices that constantly monitor various aspects of driving behavior, including information about the driver - e.g. head, eye and hand movements -, about vehicle-movements - e.g. deceleration, acceleration, position on the road, driving speed -, and about the direct environment - e.g. time headway, road, traffic densities and weather conditions . Naturalistic-driving-data (NDD) appeared as a result of this need, as a crucial data source with high environmental validity. NDD allows for the assessment of not just driver behavior but also road infrastructure and pre-crash safety measures in the real world. However, NDD’s great potential is hindered by its complexity. Consequently, new methods to analyze NDD are greatly needed [3].

Recording and Analysis System can provide a comprehensive GIS-based solution for accident event control and management as a real-time monitoring system [6]. Naturalistic driving presents two main advantages with regard to the methods traditionally used in road safety: (1) the experiment is unconditioned due to the fact that the research staff does not intervene -at least, theoretically, and (2) it allows for recording a large number of parameters that may potentially influence the driving performance [7].

Deep Learning (DL) is a branch of machine learning and in the two last years, has become the main tool for many applications domains such as NLP, vision, audio, etc. lately, researchers are beginning to successfully apply deep learning methods to graphic datasets in areas such as social networks, road networks, and etc., where data is inherently structured in a graphical way [8]. Graphs are popular data structures to represent information with connections where information represent in nodes and connections in edges. For example, the Facebook network where users are nodes and connections between the users are edges [9].

If using deep learning techniques to generate predictions on graphs, we need a way to convert them into d-dimensional vectors of real values. As a result, node embedding is used, a low-dimensional representation that help in the generalization of the input data better. one of this way is a

Node2Vec algorithm [9]. The algorithm aim is to preserve a neighborhoods network of nodes, by finding a representation for the graph as real vectors with a small Euclidean distance between neighborhoods, simultaneously represented the node by the structure[9]. When graphical data is used in classification tasks, we usually want to describe each node in the graph in such a way that it can be feed into a machine learning algorithm. Without DL techniques, I will manually extract's the features, this is a waste of time like the neighbors' number of a node has [8].

The data used in this thesis is crash data from Queensland roads taken from the Queensland Government website [10].This databases are divided into six tables contains data on accidents reported to the police that occurred as a result of the movement of at least one road vehicle on a road or in a road-related region. From 1 January 2001 to 31 December 2019, fatal road traffic crashes to 31 December 2019, hospitalization, medical treatment, and minor injury crashes to 31 December 2019 and property damage only crashes to 31 December 2010, because the Queensland police department only stopped reporting/recording property damage after December 31, 2010. We convert this data into graph where the nodes are one or more features and the edges are Euclidean distance between the features.

1.2 Problem statement:

The key problems of this thesis relate to the techniques are used upon dataset, which are:

1- Machine learning applications that are deal with GPS large-scale data are restricted.

2- Researcher have to process multiple datasets to verify accident areas.

3- Some researchers have used such kind of deep learning to detect traffic accident data from social media and have shown problems like time and location bias, as well as impacts of influential hashes and users.

4- In NDD there are many latent features that are not covered in the previous researches for the purpose of prediction and classification systems, and increase the accuracy of the proposed system.

1.3 Challenges:

This thesis has faced many challenges, most importantly:

1- Curse of Dimensionality: the number of records in the dataset is very huge (328247 instances).

2- Apply classifier models to this dataset with the less possible error and less execution time.

3- Fully connected graph generation.

4- How to apply a CNN model to this dataset, which in most research is used with images classification.

1.4 Thesis motivations:

The motivations that encouraged to engage in this field are summarized as:-

1- The need to avoid the traffic accidents.

2- Analyzing drivers ’orientation towards discovering accident places on the roads is an important matter for drivers themselves, as well as for passengers.

3- Determine the factors that cause road accidents. This puts the power in the hands of the drivers and the people involved in road maintenance matters.
4- Reducing the risk of accidents that sometimes lead to death, encourage the establishment of a prediction model for the location of accidents.

1.5 Aims of the thesis:

This thesis has a number of main aims, including:

1. Determine the areas with frequent accidents, depending on the type of severity and the force of the accident.
2. Predict and classify the severity of accidents.
3. Extract new features by using deep graph learning.
4. Develop classification system such as ANN and CNN to raise the accuracy of detecting crashes area compared with other methods used in other researches.

1.6 Thesis Contribution:

1- Using Node2Vec algorithm for features extraction from this NDD dataset.
2- Increase the accuracy of accident detection and classification.
3- Use the CNN classifier on this type of dataset.

1.7 Related Works:

This section reviews current methods used to predict road accidents and what methods are used to detect patterns that lead to these accidents.

The data set used for this thesis is from the Queensland Official Site, and there are not many studies on it. Thus, through a review of previous works, it was found that many of them used weather changes as one of the leading causes of road accidents, as well as the age of the driver and diseases such as blindness, Alzheimer's, and others.

- ALkheder et al [11] The researchers used an Artificial Neural Network (ANN) to predict the seriousness of injuries in traffic accidents based on 5973 traffic accident reports from Abu Dhabi over a 6 year period (from 2008 to 2013). The ANN classifier was created using the data-mining program WEKA (Waikato Environment for Knowledge Analysis). The results of the experiments showed that the developed ANN classifiers can accurately predict accident severity. For the training and testing results, the overall model prediction output was 81.6 percent and 74.6 percent, respectively. Traffic accident data was split into three clusters using a k-means algorithm to improve the ANN classifier's prediction accuracy. An ordered probit model was used as a comparison model to validate the efficiency of the ANN model. the performance of the two models was the ordered probit model's accuracy of 59.5 percent was significantly lower than the ANN's accuracy of 74.6 percent.

- L. Yasaswini and et al [2] The researchers proposed work to determine the factors that contribute to fatal accidents. This is done by using Convolution-Neural-Networks(CNN) to analyzing road accidents by effective records clustering and consider appropriate features. For discover hidden patterns that are the main root for accidents, to this reason several combinations of attributes for large datasets are analyze. Furthermore, one of the probabilistic approaches used to predict independence between variable pairs is the use of the Naïve-Bayesian classification. The results for the classification techniques are dependent on performance metrics finding the Recall is 0.68 for naive and 1 for CNN. This mean CNN is more efficient than Naïve-Bayes classifier in identifying the risk factor.

- ❖ Jingqiu Guo et al [1] develop an unsupervised deep learning hybrid model to trace driving behavior and dangerous forms. This model brings together Maps that are Self-organized and Auto-encoder, to extract implicit features and classify driving behavior. Neural networks then are implemented to the data taken from 4032 cases gathered from a System that is Global Positioning as sensors in China, Shenzhen. The experiments have shown that back propagation through multi-layer autoencoders is effective for non-linear and multi-modal dimensionality reduction, giving low reconstruction errors from big GPS datasets..

- ❖ Frank Knoefel et al [12] The researchers provide a framework for evaluating driving and discuss how natural driving studies can aid in such evaluations depending on the sensors and computers in vehicles. The system includes driving characteristics such as (location, pace, amount of driving, road type), behavior and reactions such as (intersections, traffic, merging, lane changes, traffic lights, pedestrians, and other vehicles), and destinations (distance, sequence, diversity, and route planning) and conditions of driving (season and day of time). To clarify the use of naturalistic driving data, researchers used data from a subset of Ottawa drivers from the study of Can-drive.

- Maher Ibrahim Sameen and Biswajeet Pradhan [13] Based on reports about 1130 accidents that happened on the Expressway of North-South Malaysia over a six-year span between 2009 and 2015, the two researchers created a deep learning model, it was used to predict the seriousness of injuries sustained in traffic accidents by using a Recurrent Neural Network (RNN). In comparison to conventional Networks, such an approach is more accurate for successive data, and it is supposed to reach correlations that are temporal between records relate to traffic accidents. Multi-Layer Perceptron (MLP) and Bayesian Logistic Regression (BLR) models were compared to the proposed RNN model which overlooked the BLR and MLP models, according to the comparative analyses results. The RNN model had a validation accuracy of 71.77 percent, while the MLP and BLR models had 65.48 percent and 58.30 percent, respectively.

- Md Nasim Khan and et al [14] The researchers focused in their study on understanding in general the behavior of a driver and in particular speed selection during foggy and clear weather conditions. Because the lack of visibility in fog is one of the main causes of car accidents. An initial comparative analysis development and a model of ordered logit were used to evaluate driver speed performance in clear and fog conditions of weather, the work data taken from SHRP2 database for the Natural-Driving-Study (NDS). Preliminary analysis findings showed a 10% and 3% decrease in velocity due to near and distant kinds of fog, in that order. Furthermore, the findings show that reducing speed amounts to 1.31 and 1.28 times which is higher for near and distant fog, accordingly, in relation to drivers driving in pure conditions of weather.

- M. I. Sameen et al. [13] Researchers have studied the effectiveness of deep learning in predicting the seriousness of casualties in traffic incidents on the highways in Malaysia. They proposed three architectures dependent upon networks, convolutional neural networks, and recurrent neural networks improve it by optimizing network search to set hyper- parameters for the models that can better anticipate output with computational costs that are lower. The results are dependent on an approach that is 10-fold cross-validation where the RNN model bested the CNN model (70.30 %) and the NN model (68.79 %) with an average accuracy of 73.76 % among the tested algorithms.

- Zhenhua Zhang and et al [15] The researchers used deep learning for the traffic accidents detection from social media data. They investigated over 3 million tweets relating to traffic incidents over the course of a year in two metropolitan areas: Northern Virginia and New York City. This verification is done by using two deep learning methods: Long Short Term Memory (LSTM) and Deep Belief Network (DBN), these ways are implemented in extracted tokens. The researchers found the results of DBN classification outperform on supervised Latent Dirichlet allocation (sLDA) and Support Vector Machines (SVMs), where each token has 44 unique features and 17 paired token features, DBN can achieve an overall accuracy of 85%. DBN also outperforms the ANN with a single hidden layer.

- Zhuoning Yuan and et al [16] The researchers was proposed the Hetero-Conv-LSTM framework to solve the problem of spatial heterogeneity challenge in the data , where a few new ideas are implemented on top of the basic Conv-LSTM model, such as spatial model ensemble and incorporating spatial graph features. The experiments of work implemented on the entire state of Iowa over the data 8-year, and the proposed framework shows the reasonable accurate predictions and improves the prediction accuracy significantly over approaches of baseline.

- Hongtao Shi and et al [17] They propose a new feature optimization approach based on Deep Learning (DL) and Feature Selection (FS) techniques to provide robust and optimal features for traffic classification. First, symmetric uncertainty is used to remove unrelated features in traffic data network sets, then a deep learning feature creation model is applied to these relevant features to reduce dimensions and feature creation, and finally, weighted asymmetric asymmetry is used to determine the optimal features by removing the redundant ones.

- Dongwei Xu and et al [18] The researchers have represented the traffic road network as a graph and proposed a new framework for predicting a traffic flow named the Graph Embedding Recurrent Neural Network (GERNN). It is difficult to get high accuracy because the traffic flow data have complex Spatio-temporal characteristics, especially under the Sydney coordinated adaptive traffic system. Therefore, this representation of the traffic road network as a graph with the proposed framework addresses the difficulty of predicting road state. The proposed model compared with baselines with metrics RMSE and MAE. the results are RMSEs of GERNN are by approximately 24.8, and MAEs are approximately 21.5% reduced.

In this thesis, we use deep graph learning to extract new features instead of working on the dataset features directly as in the previous works mentioned above, to raise the accuracy of classifiers.

Table 1.1: Summary of the related works.

No	Methodology	Evaluation Measures	Dataset	Result
1	ANN model with WEKA	Accuracy	Traffic accident reports from Abu Dhabi	74.6
	Ordered Probit model			59.5
2	CNN model	cRecall	FARS (Fatality Analysis Reporting System)	0.68
	Naïve-Bayesian			1
3	combines Autoencoder and Self-organized Maps (AESOM)	_______________	Shenzhen Urban Transport Planning Center, Shenzhen, China	extract latent features and classify driving behaviour
4	sensors and computers in vehicles	________________	a subset of Ottawa drivers	number of driving parameters extracted easily from these data
5	RNN model	Accuracy	North-South Malaysia	71.77
	Multi-Layer Perceptron model			65.48
	Bayesian Logistic Regression model			58.30
6	a comparative preliminary analysis	speed selection	SHRP2 Naturalistic Driving Study (NDS)	speed selection behavior where,

	and an ordered logit model			10% and 3% reduction in speed because of near fog and distant fog,respectively.
7	RNN model	10-fold cross-validation and accuracy	Malaysia highways	73.76
	CNN model			70.30
	ANN model			68.79
8	LSTM	Accuracy	two metropolitan areas: Northern Virginia and New York City	85
	DBN			
9	the Convolu- tional Long Short-Term Memory (ConvLSTM) neural network	Accuracy	state of Iowa	A number of detailed features such as weather, environment, road condition, and traffic volume are extracted from big datasets
10	Deep Learning (DL) and Feature Selection (FS) (Weighted Symmetric Uncertainty (WSU))	________________	network traffic	reduce the dimension of feature space
				remove the irrelevant features
11	Graph Embedding Recurrent Neural Network (GERNN)	RMSE	Sydney coordinated adaptive traffic system	24.8
		MAE		21.5

1.8 Thesis organization:

After Chapter one, which presents an introduction to the entire thesis, the rest of the thesis is structured as the following:

- Chapter two: shows the main concepts used in this thesis based on naturalistic driving data (NDD) and how to select important features, represent and classify data.

- Chapter three: shows the basic steps to identify the important features in the roads crash of Queensland dataset (feature selection). Then building the classification models.

- Chapter four: illustrate experimental results of the techniques implemented on the data set.

- Chapter five: reviews the conclusions have reached by this thesis and gives suggestions for future work.

Chapter Two
Theoretical Background

2.1 Introduction

The Naturalistic Driving Data (NDD) and the main factors that cause accidents are clarified in this chapter as well as it explains the concept of feature selection methods and filter methods such as Info Gain, Gain Ratio, Relief-F, and Chi-Squared that used with ranker search method for select the main features of the dataset. Furthermore, Deep Graph Learning (DGL) extracts more significant features in the data as a set of nodes with its relationships (edges) depending on the locations of accidents such as latitude and longitude coordinates. Then represent each node as a low-dimensional vector, the same length for each node's vector by using the Node2Vec algorithm, also we explain evaluation metrics to evaluate between the two models artificial neural network (ANN) and convolution neural network (CNN).

2.2 Naturalistic Driving Data (NDD)

Naturalistic driving (ND) is a method of research that has been identified as promising to get new insights into many different road safety issues. In most ND studies, cars are outfitted with a number of tiny cameras and sensors that continually record driver behavior and environmental variables. this permits observation and study of the interrelationships between the driver, vehicle, road, and other traffic in normal situations, conflict situations, and crashes [19]. In the last several years, traffic safety research, particularly that related to the automobile industry, has shifted its focus from preventing injuries during an accident to avoiding the crash or reducing its consequences [20]. Understand driver behavior is crucial in the evaluation and development of safety measures. Naturalistic driving data can facilitate this understanding via providing information about crash causation and contribute to the evaluate the measures of safety of pre-crash and the effects of driver behavior on safety[20].

2.3 Statistical Methods

Most of the time, preprocessing the dataset is needed in order to turn it into a format that the neural network can learn from. Since the input data must be a numerical value, when dealing with categorical values, for example, the data must be converted and normalized or standardized before the training can begin [21].

- **Normalization:-** It is the process of converting the data to a particular range, such as between 0 and 1. Normalization is required when there are big differences in the ranges of different features. To convert the data to the range 0, 1 using the following formula [22].

$$N = \frac{X-Min}{Max-Min} \tag{2.1}$$

- **Standardization:-** The data need to be scaled so that all the features or dimensions have the same scale. Usually, z-score standardization is used, yielding a mean x=0 and a standard deviation std=1 of: [23].

$$Z = \frac{X-\bar{X}}{Std} \tag{2.2}$$

Where can scale the values via calculating their means:

$$\bar{X}=\frac{sum(X)}{N} \tag{2.3}$$

and their standard deviations:

$$Std=\sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(X_i - \bar{X})^2} \tag{2.4}$$

2.4 Feature selection techniques

Feature selection (FS) methods can be used in data pre-processing to achieve efficient data reduction. This is helpful for finding careful data models [24]. The tricky task of choosing a feature is how to get the perfect subset of related and unnecessary features that will provide an ideal solution without further complicating the task [25]. It plays a major role in machine learning, data mining, and pattern classification applications. Therefore, In order to speed up testing, a successful feature selection system based on the number of features tested for sample grading is needed. reduce time complexity, evaluate predictive accuracy, and reduce computational complexity [26]. FS methods There are three types of approaches: filter methods, wrapper methods, and embedded methods.

2.4.1 The Filter Methods

The filter method applies some measuring techniques to assign a score for each feature. Then the features are ranked or selected based on their score value [27]. Filter techniques are less computationally expensive, avoids overfitting, but ignore dependencies between the features [28]. Filter methods are based on some measures such as (dependency, distance, and consistency measures). Examples of ranking algorithms that are used for this process are information gain, Pearson's correlation, Chi-square, mutual information, correlation coefficient, etc [27].

2.4.1.1 Information Gain (IG):

Information gain is attributed evaluator used in feature selection when information gain chooses then default the ranker search method gets selected [29]. Selects candidate features that contain more information, assigning a score to each feature depending on how much more information about the class is gathered when that feature is used. [25].

IG is measure based on the information theory of entropy. Entropy is a measure of disorderliness or noisiness. It is measures the reduction in entropy before and after including the features [30]. Disadvantage of IG is that it favors features with more values even once they might not be more informative [25]. IG is described as follows:

$$IG(X/Y) = H(X) - H(X/Y) \tag{2.5}$$

Where,

$$H(X) = -\sum_{i=1}^{k} P(x_i) \log_2 P(x_i) \tag{2.6}$$

is the entropy of the variable X, and

$$H(X/Y) = -\sum_i P(y_i) \sum_i P(x_i|y_i) \log_2(P(x_i|y_i)) \tag{2.7}$$

is the entropy of X after observing another variable Y.

2.4.1.2 Gain Ratio (GR):

The Gain Ratio is a non-symmetrical metric that was created to adjust for the IG's bias [25]. It takes into account the probability of each attribute value. The split information (SI) element considers the possibility that an attribute has several values [31]. It's summarized this way:

$$\mathrm{SI(A)} = -\sum_{\mathrm{veValues(A)}} \frac{|\mathrm{S_v}|}{|\mathrm{S}|} \log_2 \left(\frac{|\mathrm{S_v}|}{\mathrm{S}}\right) \tag{2.8}$$

and the GR is

$$\mathrm{GR(A)} = \frac{\mathrm{IG(A)}}{\mathrm{SI(A)}} \tag{2.9}$$

2.4.1.3 Relief-F:

Used to evaluate the relationship between the target class and the feature. The relief algorithm was inspired by instance-based learning. This algorithm calculates a proxy statistic for each feature that can be used to estimate the "relevance" or "quality" feature for the target. These statistics of features are referred to as weights that can range from −1 (worst) to +1 (best) [32]. The algorithm isn't restricted to two-class problems, it's more robust, and it can handle noisy and incomplete data. [33].

This algorithm is used for classification where is similar to Relief, chooses the instances r_i at random and then searches for k nearest neighbor in the same class is named as (nearest hit H_j) and in the dissimilar classes is named as (nearest miss M_j) [34].

The algorithm updates the quality estimation W[A] for all the attributes A that mainly depends on the values of r_i, H_j, and M_j. If the instances $H_j$and r_i have similar values of attribute A, then the attribute A is separated into two instances with similar classes, which is essential to minimize the quality estimation W[A].In contrast, if the instances $H_j$and r_i have dissimilar values of the attribute A, then the attribute A separated into two instances with the dissimilar classes, which is essential to maximize the quality estimation W[A]. The whole mechanism repeat for *m* times, where m is represented as a user-defined parameter[34].

The weight of each feature is updated via the following equation:

$$W[A] = W[A] + (\bar{H} + \bar{M})/m \quad (2.10)$$

Where:

$$\bar{H} = -\sum_{j=1}^{k} D(A, r_i, H_j)/\mathrm{k} \quad (2.11)$$

$$\bar{M} = \sum_{C \neq cl(r_i)} [\left(\frac{P(C)}{1-P(cl(r_i))}\right) \sum_{j=1}^{k} D(A,\ r_i,\ M_j(c))]/k \quad (2.12)$$

Where, W[A] is denoted as quality estimation, r_i is represented as instances, A is indicated attributes, H_j and M_j are denoted as nearest hit and nearest miss values, P(C) is represented as prior class, D is indicated as distance between the selected instances r_i, C is represented as total number of classes, and cl(r_i) is denoted as class of the i^{th} sample.

2.4.1.4 Chi-Squared:

The chi-squared filter method used to check the independence between two events [28]. It assesses the value of an attribute by computing the chi-square statistic value with respect to the class. The initial hypothesis H0 is the assumption that the two features are unrelated [35]. The Chi-square statistic is calculated in the following way.

$$\mathrm{X}^2 = \sum_{\mathrm{i}=1}^{\mathrm{r}} \sum_{\mathrm{j}=1}^{\mathrm{c}} \left(\frac{\mathrm{O_{ij}} - \mathrm{E_{ij}}}{\mathrm{E_{ij}}}\right)^2 \quad (2.13)$$

Where :

$\mathrm{O_{ij}}$ = observed frequency of class j, given attribute value i.

E_{ij} = expected frequency of class j, given attribute value i.
r = number of distinct values of attribute
c = number of classes.

Basically, the greater than calculated chi-squared value, the most important feature is.

2.4.2 Wrapper methods

Wrap techniques get their name from the fact that they raise a classifier in the feature selection algorithm [36]. Instead of arranging each individual feature, this technique creates a subset and uses this subset to train the learning model. This method is more expensively in computation than the previous method [30]. Filter methods may fail to improve classification accuracy due to specific features, but the wrapper method can always provide the best subset of features. The Warp model uses a search algorithm to search for a possible set of features. Examples of search algorithms include: Forward Selection, Backward Selection, Genetic algorithm [30].

2.4.3 Embedded method

A filter and a wrap model are combined to create an embedded structure. It is applied using its built-in feature selection techniques, and the learning model and feature selection are not separated [30]. It reduces the computation time it takes to reclassify the various subgroups that are done by wrapper methods [27]. This method chooses the features based on the training. Least Angle Regression (LARS), decision trees, and function cancellation[18], as well as KP-SVM [27], are examples of embedded approaches.

2.5 Graphical Deep Learning

Graphical are an effective way to represent data generated by a diverse variety of natural and artificial processes. It has both a compositional and relational nature, as it is made up of nodes represent the information pieces, and the relationships between the connected nodes are represented by the links that determine its structure [37]. Graph-structured data are widespread in many areas such as social networks. Many analytics tasks in these domains such as graph classification, clustering, and regression require representing graphs as fixed-length feature vectors to make Machine Learning (ML) algorithms easier to apply [38].

Regardless of the training objective one cares about, almost all deep learning models working on graphs ultimately produce node representations, and this models automatically extract the relevant features from a graph [37].

There are three types of deep graphical networks. The first are Deep Neural Graphical Networks (DNGNs), which are systems that are inspired by neural architectures. The second category consists of Deep Bayesian Grid Networks (DBGNs), which are probabilistic graphical structures. Lastly, the family of Deep Generative Graph Networks (DGGNs) leverages both neural and probabilistic models to generate graphs [37].

2.5.1 Node2Vec algorithm

Node2vec is a graph representational learning framework that can provide continuous vector representations for nodes based on network structure [39]. It creates embedding's of nodes in low-dimensional spaces using bias random walks, which can then be used for tasks such as link prediction and multi-label classification [40].

For each node, Node2Vec design a flexible neighborhood sampling strategy which allows us to smoothly interpolate between BFS and DFS. We achieve this by designing a flexible biased random walk procedure that can examine neighborhoods in a Breadth-First Search (BFS) and Depth-First Search (DFS) way [41]. Node2vec has mechanisms to balance the random walk between BFS and DFS sampling, allowing more local or global structures to be preserved [42].

If we define a second order random walk with two parameters p and q which guide the walk: Consider a random walk that just traversed edge (t; v) and now resides at node v, to make a decision see the figure (2.1), and other details are shown in the algorithm1 Node2Vec.

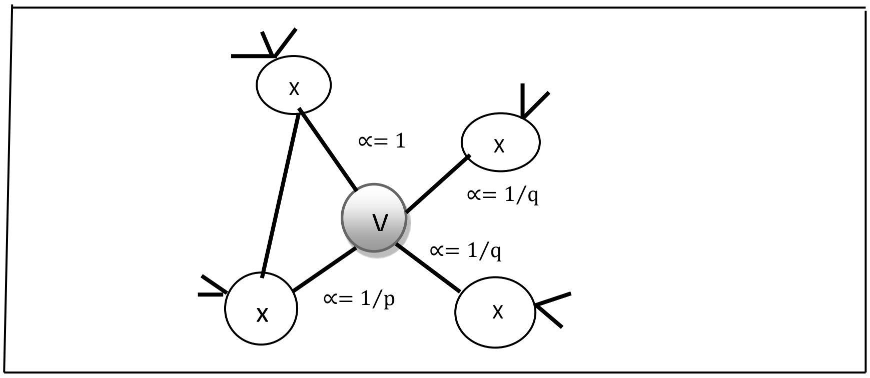


Figure (2.1) Random walk procedure in Node2Vec.

node2vec alg0rithm

Algorithm 1 Node2Vec algorithm.

```
                    Learn Features
Input: (Graph G = (V,E,W), Dimensions d, Walks per
  node r, Walk length l, Context size k, Return p, In-out q)
  $\pi$ = PreprocessModifiedWeights(G, p, q)
  $\acute{G}$= (V,E, $\pi$)
  Initialize walks to Empty
  for iter = 1 to r do
     for all nodes u ∈ V do
        walk = node2vecWalk( $\acute{G}$, u, l)
        Append walk to walks
  f = StochasticGradientDescent(k, d, walks)
Output: f
```

```
                    Node2VecWalk
Input: (Graph $\acute{G}$ = (V,E, $\pi$ ), Start node u, Length l)
  Inititalize walk to [u]
  for walk_iter = 1 to l do
     curr = walk[-1]
     $V_{curr}$ = GetNeighbors(curr, $\acute{G}$)
     s = AliasSample( $V_{curr}$, $\pi$)
     Append s to walk
Output walk
```

2.6 Artificial Neural Network (ANN)

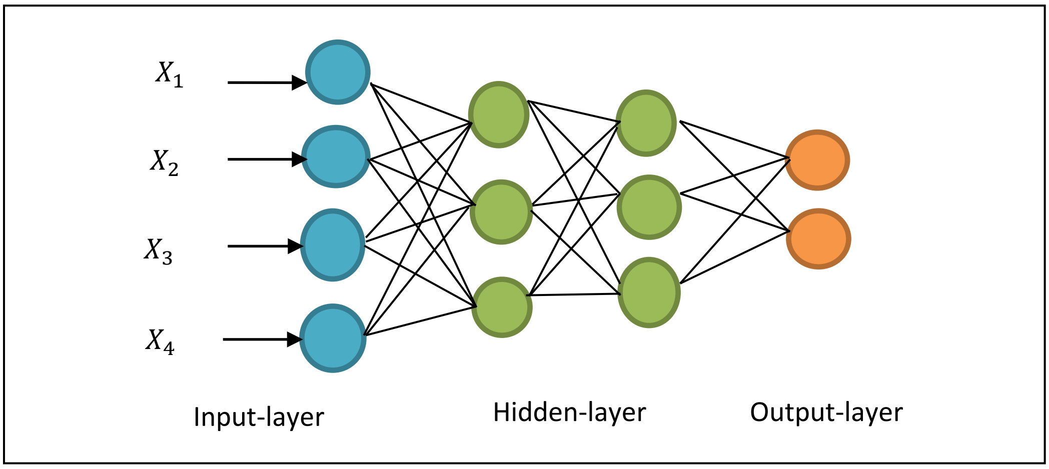
An Artificial Neural Network (ANN) is a mathematical model based on the functional properties of biological neural networks. A neural network is made up of a group of artificial neurons that work together to process information using a connection form for computation [43]. Furthermore, various weights are assigned to the interactions between different neurons, each representing the amount of influence of one neuron on another neuron [44].

ANN learning can be either supervised or unsupervised. Supervised training is accomplished by giving the neural network a set of sample data along with the expected outputs from each of these samples. Unsupervised training is similar to supervised training except that no expected outputs are provided [43]. ANN was found to be more efficient and more accurate than other classification techniques [45].

The classification process with a neural network is divided into two phases. To identify the input-output mapping, the network is first trained on a set of data. The network is then used to determine the classifications of a new set of data once the weights of the connections between neurons are fixed [46]. Though several different types of ANNs have been suggested, feed-forward neural networks (FNNs) are the most common and widely used in many applications [45].

Artificial neural networks, in general, have three categories of layers like the following [47]:

- Input layer: Get the raw data that the network has received.
- Hidden layers: the function of these layers is determined by inputs, weight, the relationship between them, and the hidden layers. Weights between input and hidden units determine when a hidden unit needs to be activated.
- Output layer: The output unit's function is determined by the hidden unit's activity and weight, as well as the link between hidden units and output



Figure(2.2) Architecture of feed-forward-neural-network

2.6.1 Mathematical Model of FNNs [48]:

A neuron N_i accepts a set of n inputs, S ={ $x_j / j \ = 1, 2, \ldots, n$ }, each input is weighted before reaching the main body of a neuron Ni by connection strength or weight factor w_{ij} for j = 1, 2, …, n. In addition, it has a bias term w_0 , a threshold value θ_k, which has to be reached or exceeded for the neuron to produce an output signal. A function *f(s)* acts on the produced weighted signal. This function is called the activation function. Mathematically, the output of the *i-th* neuron N_i is calculated mathematically.

$$O_i = f\left[w_0 + \sum_{j=1}^{n} w_{ij}\, x_j\right] \qquad (2.14)$$

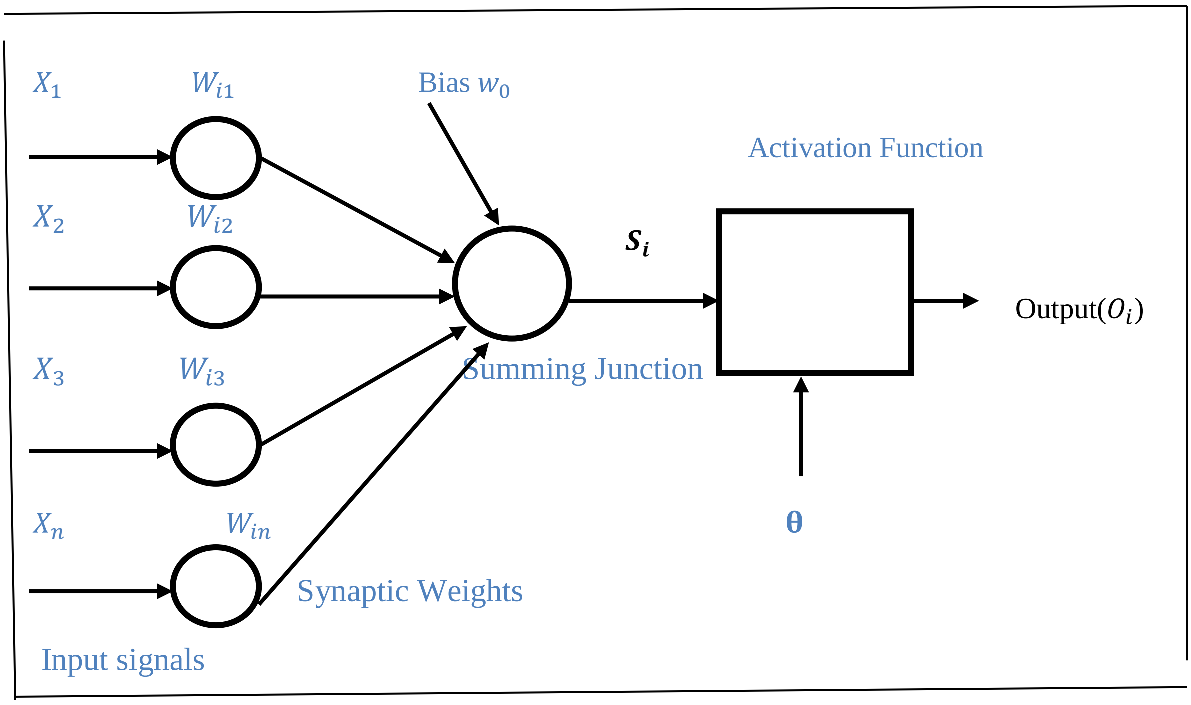


Figure (2.3)Mathematical model of artificial-neural-network

And the neuron's firing condition is,

$$w_0 + \sum_{j=1}^{n} w_{ij}\, x_j \geq \theta \qquad (2.15)$$

Figure 2.3 shows the detailed computational steps of an artificial neuron's working concept in a neural network. Now the input signal for the *i-th* neuron $N_i$is

$$s_i = \mathrm{w}_0 + \sum_{\mathrm{j}=1}^{\mathrm{n}} \mathrm{w}_{\mathrm{ij}}\, \mathrm{x}_{\mathrm{j}} \qquad (2.16)$$

This is achieved using the adder function, and the activation function yields the following output signal:

$$O_i = f(s_i - \theta_i) \qquad (2.17)$$

2.6.2 Activation Functions

As we discussed in the previous section, the output signal is a function of the various inputs x_j and the weights $= w_{ij}$ that are applied to the neuron. The neuron output function was originally proposed as a threshold function, but linear, sign, sigmoid, and step functions are now commonly used. In general, inputs, weights, thresholds, and neuron output may be binary or bipolar or real-valued . The net input to the neuron called *(net)* is generated by multiplying all inputs by their weights and adding them together. In terms of mathematics, can be formulate:

$$net = w_{i1}\, x_1 + w_{i2}\, x_2 + \cdots + w_{ij}\, x_j + \theta \qquad (2.18)$$

Where θ is a value that given as a threshold for the neurons. The neuron functions as an activation or mapping function f(net) to generate an output y that can be written as:

$$y = f(net) = f\left(\sum_{j=1}^{n} w_{ij}\, x_j + \theta\right) \qquad (2.19)$$

The neuron activation function, also known as the neuron transition function, is described by the letter f. Some examples of neuron activation functions are as follows:

- Sigmoid:

Sigmoid activation function takes the input value and makes to map it into values inside the 0 and 1 range. Accordingly, the sigmoid activation function can be used with a binary classification which results in either 0 or 1, either yes or no classification applications. The sigmoid activation function is calculated in equation 2.20.

$$f(x) = \frac{1}{1 - e^{-x}} \qquad (2.20)$$

- Rectified Linear Unit (ReLU):

ReLU activation function takes the input value and maps it into zero, if it is smaller than zero, else it returns it as it is without mapping. In another concept, ReLU makes to compare the input value with zero and take the maximum one as a winner. ReLU is faster than the sigmoid and tanh activation function due to simplicity in its computation. ReLU activation function is clarified in equation 2.21.

$$f(x) = \max(x, 0) \qquad (2.21)$$

- Softmax:

It is often used in the output layer of a neural network for classification. The softmax function is a more generalized logistic activation function which is used for multiclass classification. It is mathematically represented as

$$f(x_j) = \frac{\mathrm{e}^{\mathrm{x_j}}}{\sum_{\mathrm{k=1}}^{\mathrm{d}} \mathrm{e}^{\mathrm{x_k}}} \qquad (2.22)$$

2.7 Deep Learning

Deep learning is a form of machine learning that learns by using a number of algorithms. Artificial neural networks are used in some of the most popular deep learning techniques, such as Deep Neural Networks (DNN), Deep Belief Networks (DBN), and Convolution Neural Networks (CNN) [49]. Deep learning techniques are outperforming current machine learning techniques. It enables computational models to learn features from data at multiple levels in a gradual manner [50].

Many deep learning techniques depend on neural network architectures to classify data sets, which are referred to as deep neural networks. The word "deep" refers to the number of hidden layers in deep neural networks; unlike traditional neural networks, which can only contain two or three hidden layers, deep neural networks can hold over 150 hidden layers [51].

deep-learning models' biggest flaw is that they learn by observation. They can just see what's in the data they used to train with. Other drawbacks and challenges include the following: [52].

- Deep learning generally requires a lot of data. Furthermore, more robust and accurate models will require the use of more parameters, which will necessitate the use of more data.

- Once trained, deep learning models become inflexible and cannot handle multiple tasks. They are capable of providing effective and precise solutions, but only for a single problem. Also if want to solve a similar problem must be retrain the system.

2.7.1 Convolution Neural Network

The convolution neural network (CNN) is a supervised learning neural network of many layers. For the feature extraction function of a convolution neural network, the convolution layer and the pool sampling layer are the main modules. By using the gradient descent technique to minimize the loss function and change the weight parameters in the network, the network model increases the network's accuracy by repeated iterative training. Figure 2.3 show the core architecture of the convolution neural network [53].

Convolution Neural Networks are made up of three primary layers. Convolution Layer, Pooling Layer, and Fully-Connected Layer are the three layers [2].

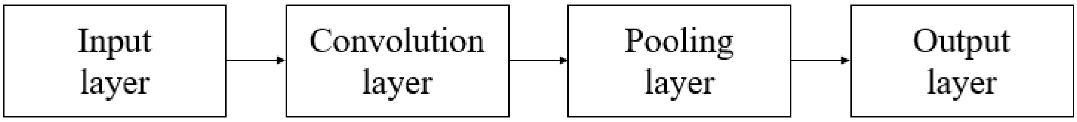


Figure (2.4). Structure of CNN [53].

2.7.1.1 Convolution Layers

Convolution layers are a collection of parallel feature maps generated via sliding separate kernels (feature detectors) over the input and projecting element-by-element dot maps as feature maps. if the input has the dimensions HxWxC (stacked together). each neuron in the input layer must be fully connected to the next layer neurons. This concept cannot be found in deep learning using Convolution Neural Network (CNN). The first layer of the network (convolution layer) is partially connected with the next layer (pooling layer), where a small window from the input neurons, e.g. 3x3, will be connected to the pooling layer, this window starting from the upper left corner to the bottom down the corner of the input matrix, it still go ahead in the matrix by moving its location every time according to the value of stride, usually, the stride is 1 where the kernel moved one cell to the right direction until reach end of columns, and one-cell below until reach end of rows and the entire input matrix complete.

Let the input source called I, convolution kernel or weight called W, bias called b. Let us suppose k filters, W, of size LXL are convolved with the input matrix to produce k feature maps called C^k each with i, j indices. The convolution operation is given mathematically in equation 2.23 as the following [54] :

$$C^{k}{}_{i,j} = \sum_{m=0}^{L-1} \sum_{n=0}^{L-1} W^{k}{}_{m,n} * I_{i+m,j+n} + b^k \qquad (2.23)$$

2.7.1.2 Pooling Layers

The main purpose of this layer is to reduce the dimensionality of its input to generate reduced dimension output by keeping only the most important information. There are two ways followed by this layer to reduce the dimensionality which are max and average pooling. In any way, the pooling layer to divides the input feature map into non-overlapping blocks, and keep the maximum number in max pooling and average of block numbers in average pooling and finally return only single value corresponding each block .

The pooling layer (max or average pooling) does the same process with each of the previously achieved feature maps from the previous convolution layer.

The mathematical operation of max-pooling operation, P, on the previously activated feature maps from convolution layer is clarifying in equation 2.7 as follows [54]:

$$P^{k}{}_{i,j} = \max\begin{pmatrix} C^{k}{}_{(i,j)} & , & C^{k}{}_{(i+1,j)} \\ C^{k}{}_{(i,j+1)} & , & C^{k}{}_{(i+1,j+1)} \end{pmatrix} \tag{2.24}$$

2.7.1.3 Fully Connected Dense Layers

The final convolution or pooling layer's output function maps are usually flattened, or converted into a one-dimensi0nal (1D) series of numbers (or vector), and linked to one or more fully c0nnected layers, often known as dense layers, in which each input and output are connected by a learnable weight. A subset of fully linked layers maps the features extracted through the convolution layers and down-sampled through the pooling layers to the network's final outputs, such as the probabilities for each class in classification tasks, if they are produced. The number of output nodes is usually equal to the number of groups in the final totally connected layer. A nonlinear function, such as ReLU, is placed after each fully connected sheet [55].

2.7.1.4 Activation functions

There are many activation functions used for many purposes. With CNN, the ReLU activation function used between layers to achieve non-linearity. The output feature maps from the convolution layers are fed into activation function to convert the linear output into nonlinear. ReLU, previously mentioned in section 2.6.2 with a traditional fully connected neural network, makes to keep any zero and positive numbers while replacing any negative number with zero, i.e. it compares between the current number and zero and keeps the maximum one. The mathematical operation of the activation function is clarifying the following equation 2.25 that makes to substitute equation 2.24, which is the linear output resulted from the convolution layer with the ReLU activation function equation 2.21 that is mentioned in the previous sections [54]

$$C^{k}{}_{i,j} = \max(0, \textstyle\sum_{m=0}^{L-1} \sum_{n=0}^{L-1} W^{k}{}_{m,n} * I_{i+m,j+n} + b^{k}) \qquad (2.25)$$

They are some examples of commonly used activation functions:

- Sigmoid: $f(x) = \frac{1}{1+e^{-x}}$ (2.26)

- ReLU: $f(x) = \begin{cases} 0 & x < 0 \\ x & x \geq 0 \end{cases}$, $\acute{f}(x) = \begin{cases} 0 & x < 0 \\ 1 & x \geq 0 \end{cases}$ (2.27)

- Softmax: $f(x_j) = \frac{e^{x_j}}{\sum_{k=1}^{d} e^{x_k}}$ (2.28)

2.8 Evaluation Measures and Performance Metrics

Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) measures are probability metrics measured by the deviation between the predicted and true values. These metrics are used in regression problems, especially to evaluate the reliability of classifiers [56].

2.8.1 Root Mean Square Error (RMSE):

This is a principal and frequently used metric, which measures the difference between the value predicted by a classifier and its true value. It is defined as follows:

$$RMSE = \sqrt{\frac{1}{M}\sum_{i=1}^{M}(Pred_c(i) - True_c(i))^2} \qquad (2.29)$$

Where $Pred_c(i)$denotes the prediction probability of instance i , which belongs to class c , and $True_c(i)$ represents the actual probability.

2.8.2 Mean Absolute Error (MAE):

It is commonly used as a substitute for root measure square error. Without their signs involved, it only averages the magnitude of the individual errors. It is defined as follows:

$$MAE = \frac{1}{M} \sum_{i=1}^{M}|Pred_c(i) - True_c(i)| \qquad (2.30)$$

The classification predictor models are evaluated using another performance measures such as Precision, Recall, F1-measure, Specificity, and Accuracy, often known as threshold metrics. Because all of the aforementioned measures rely on Confusion Matrix (CM), it is one of the most significant methods.

Table(2.1) Confusion Matrix

		Predicted Class	
		Positive	Negative
Actual Class	positive	TP	FN
	negative	FP	TN

CM is a conception of the performance of a supervised learning method. A CM can be seen in table (2.1), where TP(True Positive) refers to the number of predictions where the classifier correctly predicts the positive class as positive. TN(True Negative) refers to the number of predictions where the classifier correctly predicts the negative class as negative. FP (False Positive) refers to the number of predictions where the classifier incorrectly predicts the negative class as positive. FN (False Negative) refers to the number of predictions where the classifier incorrectly predicts the positive class as negative. The formula for these metrics is shown in Table 2.2.

Table(2.2) Briefing Formula of Metrics

Measure name	Formula
Precision	$\frac{TP}{TP+FP}$ (2.31)
Recall	$\frac{TP}{TP+FN}$ (2.32)
F1-measure	$\frac{2*TP}{2*TP+FN+FP}$ (2.33)
Accuracy	$\frac{TP+TN}{TP+TN+FP+FN}$ (2.34)

The equations of these measures that mentioned above could be described as follows [57].

- Precision: Is the ratio of true predicted samples to all predicted samples for each class. To calculate precision, use the formula (2.31) described in table (2.2).

- Recall: Is the ratio of true predicted sample to actual samples for each class. It is also known as True Positive Rate (TPR), Sensitivity, Probability of Detection. To calculate Recall, use the formula (2.32) described in table (2.2).

- F1-score: Is the average of Precision and Recall. It can be calculated, use the formula (2.33) described in table (2.2).

- Accuracy: Is the ratio of all true predicted samples to all predicted samples. To calculate accuracy, use the formula (2.34) described in table (2.2).

Accuracy is about how close measurements are to a standard true value while Precision is about how measurements are close to each other. Accuracy tells how often a model performs correctly while precision tells how the model predicts true correctly.

2.9 Universal Transverse Mercator (UTM)

Universal Transverse Mercator (UTM) coordinate systems are rarely used compared to latitude-longitude coordinate systems [58]. However, in order to determine the distance between two points using the Euclid distance method, it will be more precise to UTM coordinate system. UTM is important for two reasons, it is applicable to any location on Earth between 80° south latitude and 84° north latitude and although UTM is globally applicable apart from the poles, it maintains distance error distortions to be, at worst, 1 portion in 1,000 by dividing it up the Earth into 60 zones starting from the International Date Line, each with a width of 6 ° longitude [59]. The process of converting from spherical coordinates using a function built in the Python language by using the pyproj library.

2.10 Dataset

The dataset of crash data from Queensland [10] roads are available at Queensland government, where contains information about the location and accident characteristics of Queensland for all road traffic accidents reported since January 1 in year 2001 to 31 December 2018. This dataset is very large so it divided into six categories as databases.

Chapter Three

The Proposed Methodology

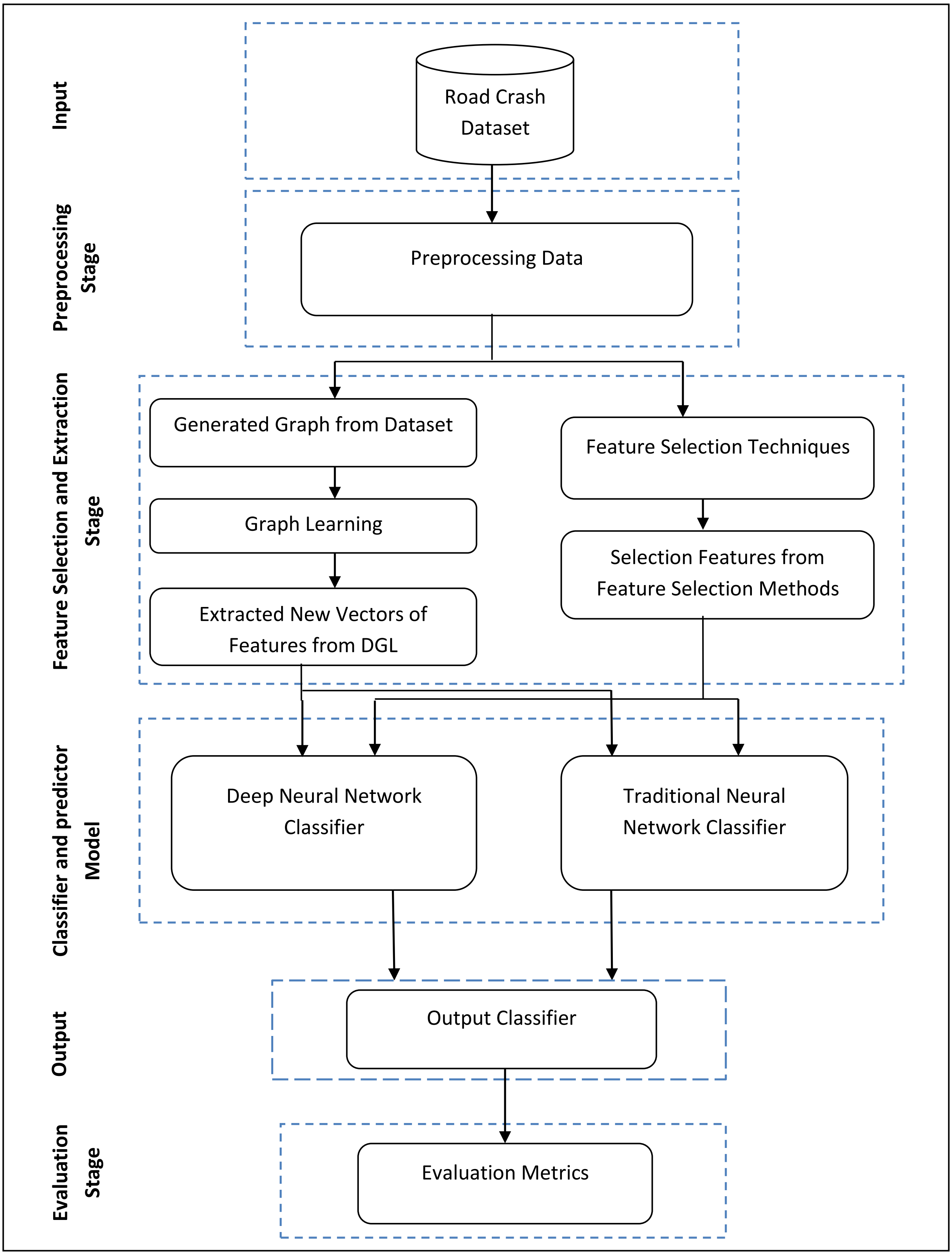
3.1 Introduction

Naturalistic driving can be generated in a huge datasets with great potential for researches. When a road accident occurs, there are many attributes affect on intensity accident such as crash street, weather conditions, road conditions etc. The analysis of these accidents helps in identifying the factors that cause accidents. This thesis presents a system to classify and predict high accident areas from the crash road of Queensland government. In this chapter, the proposed algorithms and methods will be reviewed and discussed.

3.2 Proposed System Architecture:

The architecture of the proposed system involves four main stages (data-preprocessing, filter methods for feature selection, classification - prediction, and evaluation) to achieve the thesis' aim, as shown in figure (3.1).

The first stage (data-preprocessing) includes data cleaning; transform data from nominal to numerical form, standardization. The second stage (applying filtering methods for feature selection) involves a comparison of four filtering methods: chi-square, relief, gain ratio, and information gain and select best one for classification results, also, involved extract new features from the deep graph by using the node2vec algorithm . The third stage is the construction of system classification models (ANN, CNN) and the fourth stage evaluate the proposed system models.


Input
Preprocessing Stage
Feature Selection and Extraction Stage
Classifier and predictor Model
Output
Evaluation Stage
Road Crash Dataset
Preprocessing Data
Generated Graph from Dataset
Graph Learning
Extracted New Vectors of Features from DGL
Feature Selection Techniques
Selection Features from Feature Selection Methods
Deep Neural Network Classifier
Traditional Neural Network Classifier
Output Classifier
Evaluation Metrics

Figure(3.1) Proposed System Block Diagram

3.3 Dataset Description:

The dataset of crash data from Queensland roads are available at Queensland government data website , it offers details on the location and features of Queensland crashes for all reported road traffic crashes that occurred from 1 January 2001 to 31 December 2018 [60]. This dataset is divided into six categories of database tables.

1. Road crash location contains 52 attributes, 328247 instances.
2. A Road casualty contains 7 attributes, 22372 instances.
3. A vehicle type contains 12 attributes, 2195 instances.
4. Driver demographics contain 16 attributes, 14198 instances.
5. Seatbelt restraints and helmet use contains 8 attributes, 25689 instances.
6. A factor in road crashes contains 13 attributes, 3277 instances.

To clarify each of these categories, appendix A show a brief description of each features for the location table, and dependence of the thesis work on that only because dependent on the location accident not on behavior of driver in this thesis.

3.4 Preprocessing Stage:

Preprocessing is intended to prepare datasets for machine learning techniques in an appropriate manner. This stage has been used in the thesis because road crash dataset contains many unimportant features values in classifier models and some of them affect the accuracy of the classifier model. Data pre-processing stage consists of three steps:

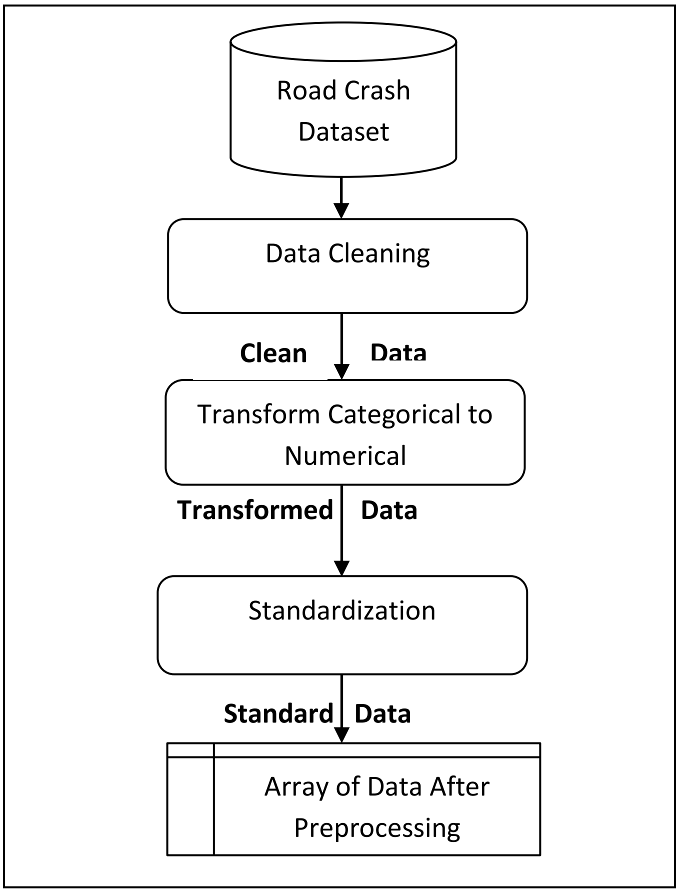


Figure (3.2) Preprocessing block diagram

3.4.1 Data Cleaning:

At this step, remove specific features according to the following condition.

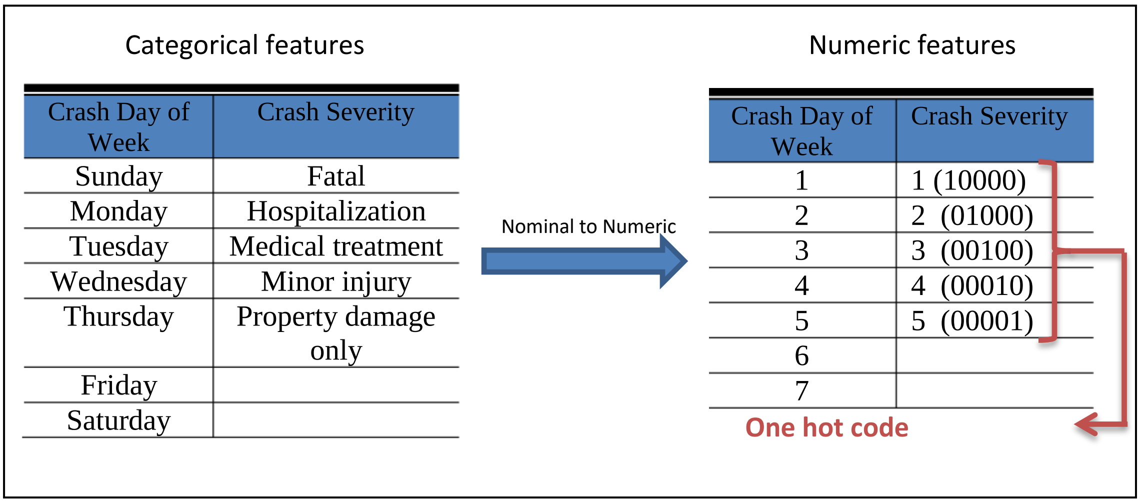
- Eliminate the features that suffer from sparcity, most of its cells contain a zero value.

```
Algorithm (3.1): preprocessing for data cleaning
 Input: Array of Road Crash Dataset ($DS_{ij}$), index [41,42,43,44,45]

 Output: Array of dataset after cleaning ($DP_{ij}$)
    n=road crash dataset.attribute_no.
 Begin:
    for i = 1 to n                // where n: number of factors (attributes) in dataset
       while  (the index of attribute found in index array)
           Deleted this feature from  $DS_{ij}$
    End for i
 End
```

3.4.2 Transform Categorical Features to Numerical:

Many machine learning algorithms cannot work directly on categorical data. These require convert all input (data) and output (label) variables to numeric. Here we use two approaches for transforming: ordinal encoding and one-hot encoding.

In ordinal encoding, for each unique value of categorical column is assigned an integer value (e.g. 'Fatal': 1, 'hospitalization': 2, 'medical treatment': 3, 'minor injury': 4, 'property damage only': 5), while in one-hot encoding used for represent each class of severity target in binary form, for example if the class is fatal will be represented as [1, 0, 0, 0, 0] with a 1 in for the first binary variable, so on with change the bit according the class. Figure 3.3 show an example on this transformation process.



Categorical features

Crash Day of Week	Crash Severity
Sunday	Fatal
Monday	Hospitalization
Tuesday	Medical treatment
Wednesday	Minor injury
Thursday	Property damage only
Friday	
Saturday	

Numeric features

Crash Day of Week	Crash Severity
1	1 (10000)
2	2 (01000)
3	3 (00100)
4	4 (00010)
5	5 (00001)
6	
7	

Figure (3.3) Categorical features to numerical

Algorithm (3.2): categorical data to numerical data

Input: Array of Road Crash Dataset after cleaning step **(DS_{ij})** where i: number of records and j: number of attributes

Output: Array of dataset after transformation **(DP_{ij})**

```
n=road crash dataset.attribute_no.
m=road crash dataset.record_no.
Begin:
    for i = 1 to n                  // where n: number of factors (attributes) in dataset
    for j = 1 to m                  // where m: number of instances (records) in dataset
        if  (the type of feature i is object)
            Call label encoder method
                give for each new item in feature label
    End for j
    End for i
End
```

3.4.3 Standardization:

At this step, all data must be scaled, in order to have the same scale with all the features or dimensions. To apply this step the equation (2.2) is used, where from this formula we obtain a mean $\acute{x}$=0 and a standard deviation std=1.

Algorithm (3.3): Standardization

Input: Array of Road Crash Dataset after above two steps **(DS_{ij})** where i: number of records and j: number of attributes

Output: Array of dataset after standardization **(DP_{ij})**

```
n=road crash dataset.attribute_no.
m=road crash dataset.record_no.
Begin:
    Let Mean and std as two arrays contain (mean and std value) for each feature.
    // find the mean and std
    Set the mean and std to zero
```

continue…..

```
for i = 1 to n                // where n: number of factors (attributes) in dataset
    for j = 1 to m            // where m: number of instances (records) in dataset
        Computing the mean for each feature by using Equation (2.3) $X^{\prime} = \frac{sum(X)}{N}$
        Computing the std for each feature by using Equation (2.4)
        $Std = \sqrt{\frac{1}{N-1} \sum_{i=1}^{N} (X_i - \bar{X})^2}$
    End for j
    Update the mean and std arrays
    scale any value x in feature i according to the Equation (2.2)  $\boldsymbol{Z} = \frac{\boldsymbol{X}-\bar{\boldsymbol{X}}}{\boldsymbol{Std}}$
End for i
End
```

3.5 Feature Selection Stage

Feature selection (FS) techniques are important to reduce the dimensionality of features space. These techniques are necessary for selecting a subset of more important features from original features because the features divided into strongly relevant, weakly relevant, and irrelevant. In this thesis need to take the features that are relevant with target. FS techniques divided into three types: filter, wrapper and embedded methods.

3.5.1 Filtering Methods

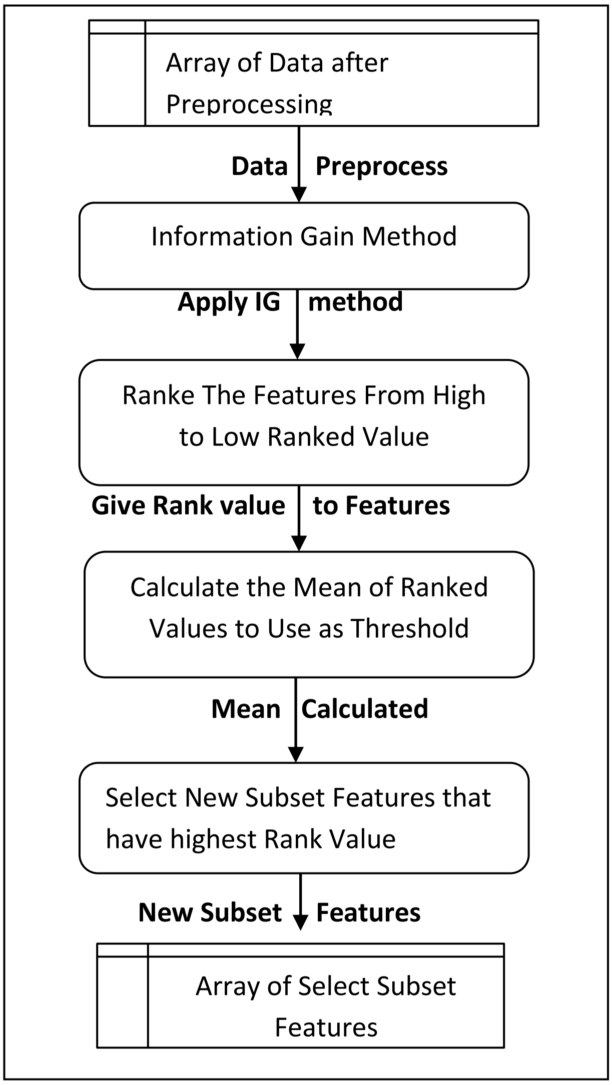
In this thesis there are many filtering methods have been used such as Information Gain, Gain Ratio, Relief-F, Chi-Squared, where in each method the features are ranked by using the ranker search method, a weight is given for each feature then the mean of these ranked features is calculated to be used as a threshold for selecting a subset of features, this thresholds value are shown in table (3.1). We use the filter methods because it is faster than other, independent on classifiers.

Table (3.1) Number of selected features by threshold value

Filtering Method	Threshold Value	No. of Selected Features
Info Gain	0.1492	10
Gain Ratio	0.0362	15
Relief-F	0.0084	21
Chi-squared	249.0483	10

3.5.1.1 Information Gain Filtering Method

It is a measure to calculate the difference between two probabilities of variables X and Y and then the features with high information gain value should be selected over other. It depends on entropy as mentioned in the chapter 2 section 2.4.1.1. The selected features are (crash street, loc suburb, loc ABS statistical area 2, loc post code, loc police division, crash street intersecting, crash DCA description, loc state electorate, loc ABS statistical area 3, state road name). Other details are shown in the algorithm 3.4.


Array of Data after Preprocessing
Data Preprocess
Information Gain Method
Apply IG method
Ranke The Features From High to Low Ranked Value
Give Rank value to Features
Calculate the Mean of Ranked Values to Use as Threshold
Mean Calculated
Select New Subset Features that have highest Rank Value
New Subset Features
Array of Select Subset Features

Figure(3.4) IG feature selection block diagram

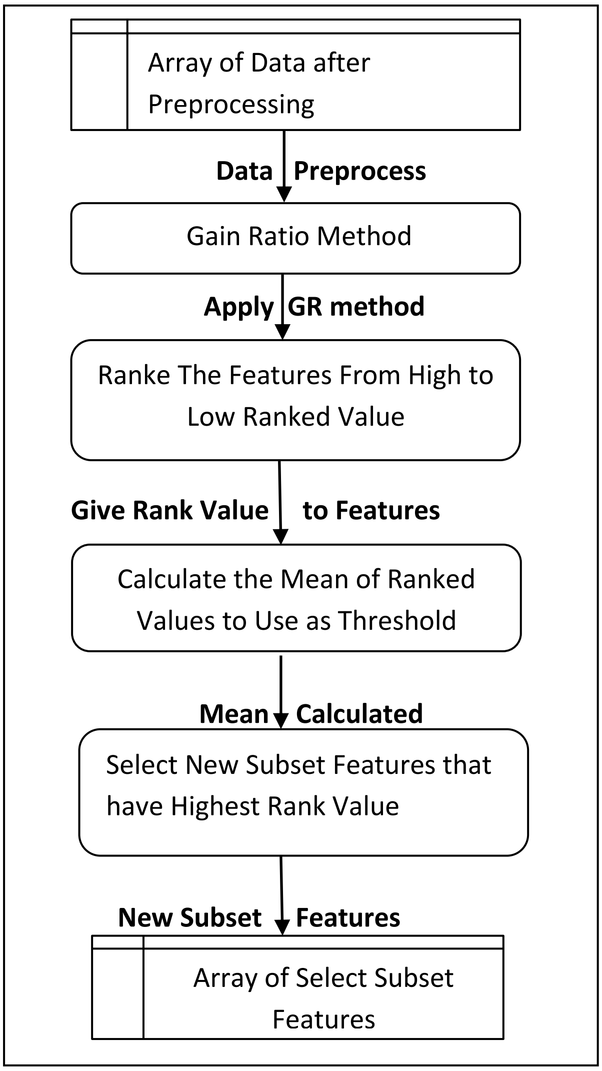
```
Algorithm (3.4): Information Gain  FS
Input: Array of Road Crash Dataset from ($DP_{ij}$) where i: number of records and j: number of features
Output: Array of ranking features
    n=road crash dataset.attribute_no.
Begin:

    1.  for i = 1 to n              // where n: number of factors (attributes) in dataset

    2.      Compute the entropy of attribute X according to equation(2.6)
              $H(X) = -\sum_{i=1}^{k} P(x_i)\log_2 P(x_i)$
```

continue….

```
3.  End for
4.   for j = 1 to n              // where n: number of factors (attributes) in dataset
5.       Compute the entropy of attribute X to the target Y according to the
     equation (2.7)   $H(X/Y) = -\sum_i P(y_i) \sum_i P(x_i|y_i) \log_2(P(x_i|y_i))$
6.    End for
7.    for i = 1 to n             // where n: number of factors (attributes) in dataset
8.      Compute the information gain is the result step2 minus the result from step5
     as equation(2.5)    $IG(X/Y) = H(X) - H(X/Y)$
9.          R_F[i] = the result of step 8
10. End for
11.  Return R_F matrix
End
```

3.5.1.2 *Gain Ratio Filtering Method*

It is used to measure the ratio of information in the feature and then select the features that have highest ratio value. This Features are (crash street, count unit pedestrian, loc suburb, loc ABS statistical area 2, count unit motorcycle moped, loc post code, loc police division, crash DCA description, crash DCA code, crash type, crash nature, loc state electorate, loc ABS statistical area3, crash street intersecting). Other details shown in algorithm 3.5.


Array of Data after Preprocessing
Data Preprocess
Gain Ratio Method
Apply GR method
Ranke The Features From High to Low Ranked Value
Give Rank Value to Features
Calculate the Mean of Ranked Values to Use as Threshold
Mean Calculated
Select New Subset Features that have Highest Rank Value
New Subset Features
Array of Select Subset Features

Figure(3.5) GR feature selection block diagram

Algorithm (3.5): Gain Ratio FS

Input: Array of Road Crash Dataset from (DP_{ij}) where i: number of records and j: number of features

Output: Array of ranking features

Begin:

```
                    //compute information gain or gain split
  1.  Compute the information gain according to equation(2.5)
          $IG(X/Y) = H(X) - H(X/Y)$
                    //compute gain ratio for each attribute
  2.  for i = 1 to n        // where n: number of factors (attributes) in dataset
```

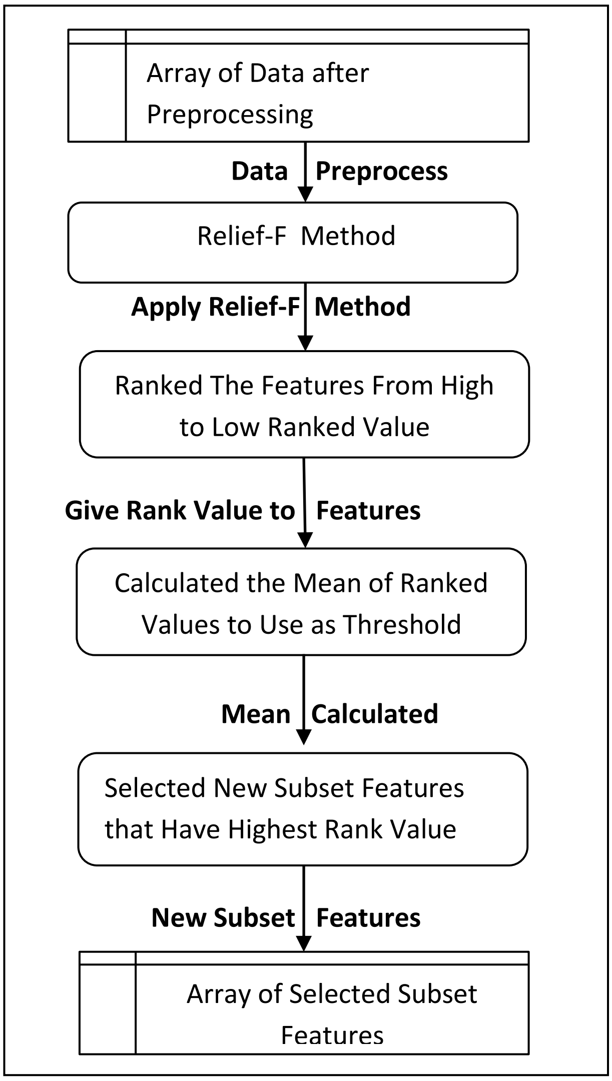
continue……

```
3.    Compute Split Information (SI) according to equation(2.8)
          $SI(A) = -\sum_{veValues(A)} \frac{|S_v|}{|S|} \log_2\left(\frac{|S_v|}{S}\right)$

4.    Compute Gain Ratio according to equation(2.9)   $GR(A) = \frac{IG(A)}{SI(A)}$

5.    R_F[i] = The result of step 4

6. End for

7. Return R_F matrix
End
```

3.5.1.3 Relief-F Filtering Method

In this algorithm the number of neighbors limited by 10 and ranked the features according to given threshold, and the given ranked value for each feature is between 1 and -1, where the low attributes are eliminated.

The selected features are (crash nature, crash type, crash DCA description, crash DCA group description, loc local government area, loc police district, loc ABS statistical area 4, crash speed limit, crash month, loc main roads region, loc federal electorate, crash lighting condition, loc police region, count unit motorcycle moped, crash road vert align, loc Queensland transport region, crash traffic control, loc ABS remoteness, crash DCA code, loc police division). For other details shown in algorithm 3.6.


Array of Data after Preprocessing
Data Preprocess
Relief-F Method
Apply Relief-F Method
Ranked The Features From High to Low Ranked Value
Give Rank Value to Features
Calculated the Mean of Ranked Values to Use as Threshold
Mean Calculated
Selected New Subset Features that Have Highest Rank Value
New Subset Features
Array of Selected Subset Features

Figure(3.6) Relief-F feature selection block diagram

Algorithm (3.6): Relief-F FS

Input: Array of Road Crash Dataset from (DP_{ij}) where i: number of records and j: number of features

Output: Array of ranking features

Begin:

```
1.  Assign all weights of feature A, W[A] to zero
2.  For  i=1 to m   //m number of random instances
3.    Randomly select a target instance $r_i$
```

continue…

```
4.      Find a  nearest hit H by using the equation  (2.11) and nearest miss
        M by using the equation (2.12)
5.   End for
6.   For A=1 to j    // j number of attributes
7.      Update the weight of feature W[A] according to equation(2.10)
8.       If  the result < zero    then
9.            ignore this feature
10.      Else
11.           R_F[A] = the result from step 7
12.   End for
13.  Return R_F matrix
End
```

3.5.1.4 Chi-Squared Filtering Method

It is used to check the independence relationship between two events the features with the target. As feature selection the aim to select the features are more dependent on the target that has high chi squared value. To calculate the chi square value we need find the observed values by observing the values in features with any class, then we calculate the expected value based on the two events are independent.

The selected features are (crash street, loc-suburb, loc ABS statistical area 2, loc post code, loc police division, crash street intersecting, crash DCA description, loc state electorate, loc ABS statistical area 3, state road name). For other details shown in algorithm 3.7.

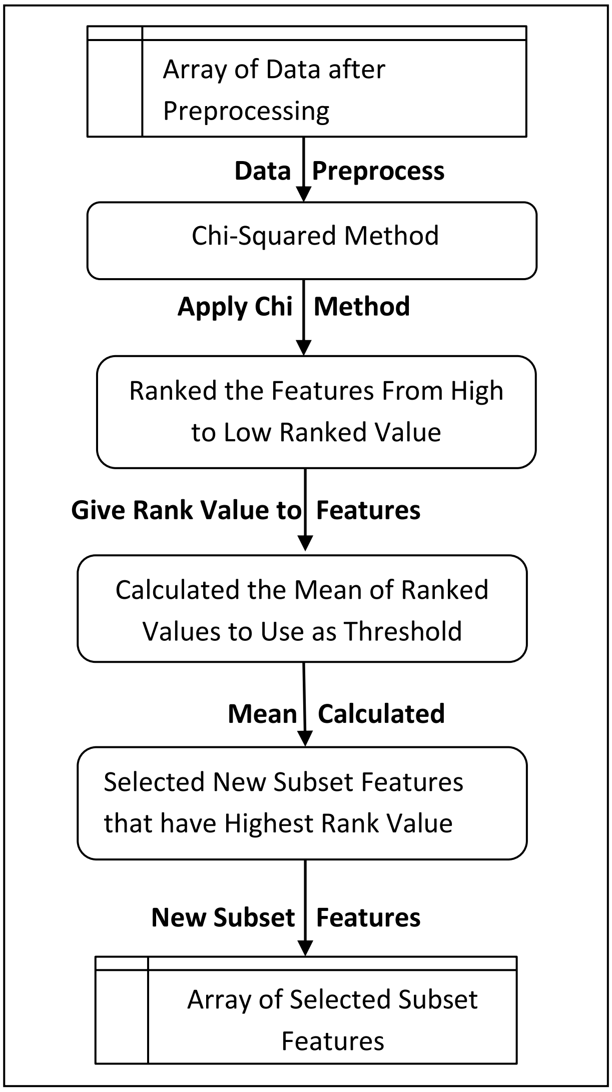

Array of Data after Preprocessing
Data Preprocess
Chi-Squared Method
Apply Chi Method
Ranked the Features From High to Low Ranked Value
Give Rank Value to Features
Calculated the Mean of Ranked Values to Use as Threshold
Mean Calculated
Selected New Subset Features that have Highest Rank Value
New Subset Features
Array of Selected Subset Features

Figure (3.7) Chi-square feature selection block diagram

Algorithm (3.7): Chi-Squared FS

Input: Array of Road Crash Dataset from (DP_{ij}) where i: number of records and j: number of features

Output: Array of ranking features

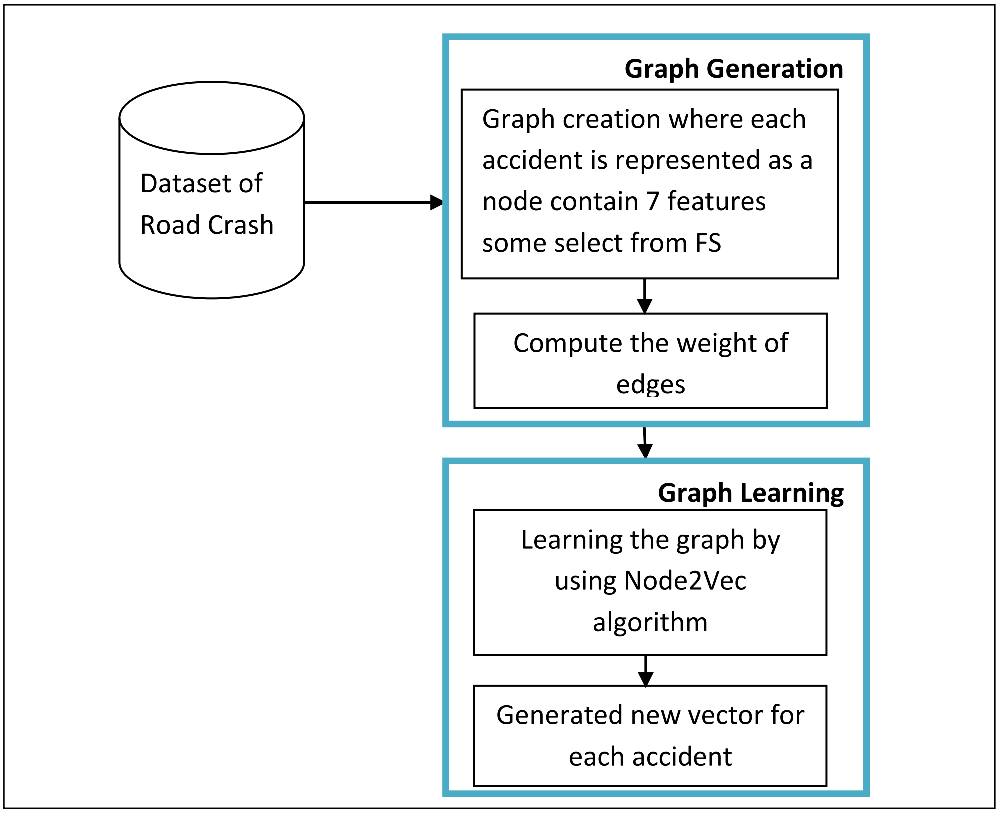
```
Begin:
    1. Define hypothesis      \\ H0 null hypothesis if features independent
                       \\ compute the observed count for each class
    2.   For i 1 to x      \\ where x feature values
    3.     For j 1 to y     \\ where y class labels
```

continue….

```
4.      Build Contingency Tables   \\ by the sum of the values of the ith feature
        across all samples in the class j
            \\  calculate the expected value for each cell in contingency table
5.      $E_i$ = sum of row observed * sum of column observed/n    \\ where n the
        total number of observed values
6.      Calculate the chi-squared value according to the equation 2.13
            $X^2 = \sum_{i=1}^{r} \sum_{j=1}^{c} \left(\frac{O_{ij} - E_{ij}}{E_{ij}}\right)^2$
7.    End for j
8.  End for i
End
```

3.6 Feature Extraction using Deep Graph Learning

The process of extracting features from learning graphs is the other side of the work for the proposed system as shown in Figure (3.8). The purpose of this process to extract new vectors of features for classification by using the Node2Vec algorithm and deep graph learning.

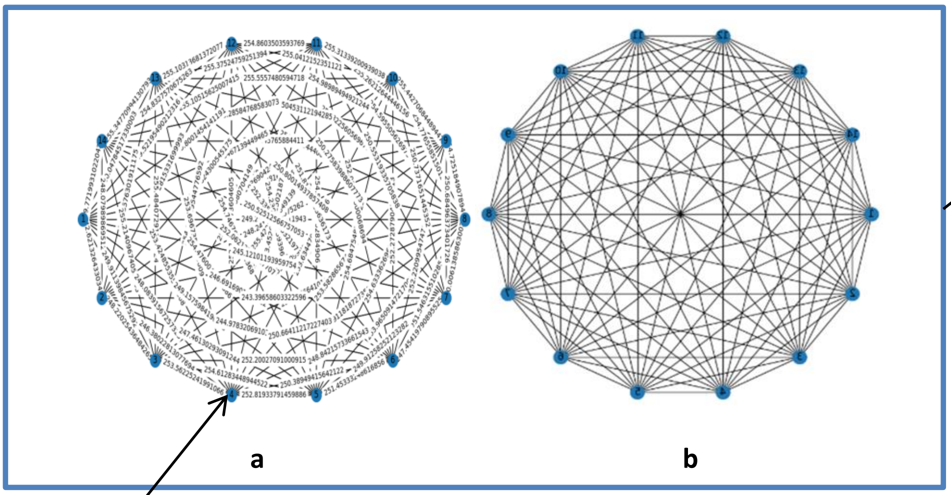


Dataset of Road Crash
Graph Generation
Graph creation where each accident is represented as a node contain 7 features some select from FS
Compute the weight of edges
Graph Learning
Learning the graph by using Node2Vec algorithm
Generated new vector for each accident

Figure (3.8) Graph learning block diagram

3.6.1 Graph Generation

Graph is a data structure that has two components the nodes and the relation between nodes called the edges. Each node is a structure and contain on the information, in this thesis, information is such as latitude and longitude, In addition to five other features that were selected manually based on the highest features selected from the feature selection methods, these features like (crash street, loc post code, crash street intersection, loc police division, loc suburb). The edge that may contain on value called weight is calculated by taking the Euclidian distance between this information of nodes but before take the distance must be convert longitude and latitude to the UTM system. It is a horizontal position representation, which means it ignores altitude and treats the earth as a perfect ellipsoid as preprocessing step. Figure 3.9 shows the graph generated from different sample of dataset contain on 4, 7, 14 accidents.



Every node contain on:

longitude , latitude ,crash street, loc post code, crash street intersection, loc police division, loc suburb

The edges are the distance between these information

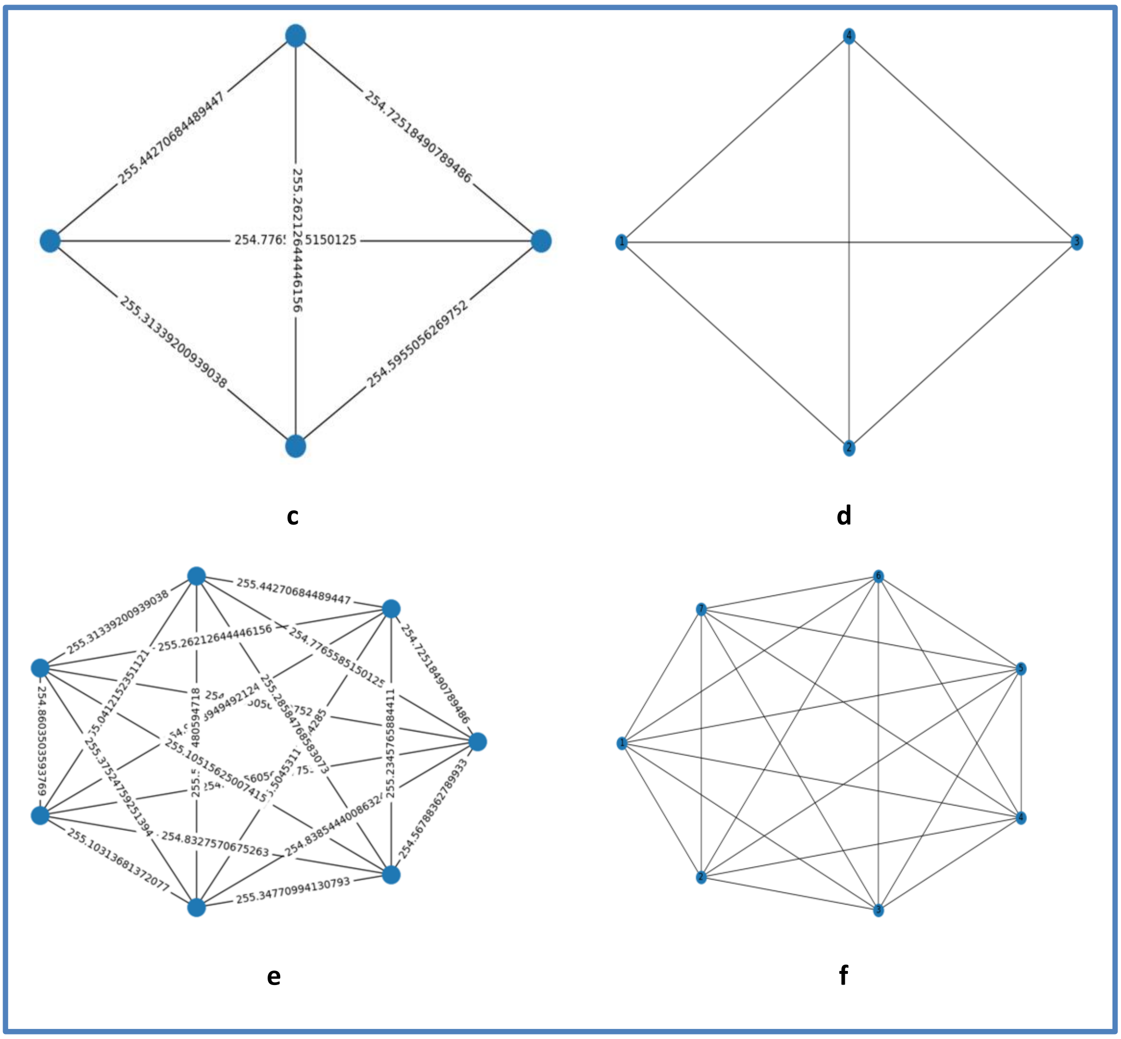


Figure (3.9) generated graph with and without label a,b with sample of 14 nodes, c,d with sample of 4 nodes and e,f with sample of 7 nodes.

Algorithm (3.8): Graph Generation

Input: Array of Road Crash Dataset where i: number of records and j: number of features

Output: Fully Connected Graph(G)

Begin:

```
for i=1 to n        // where n is number of records represent the number of accidents
    for j = 1 to m     // where m: number of attributes in dataset
```

continue…

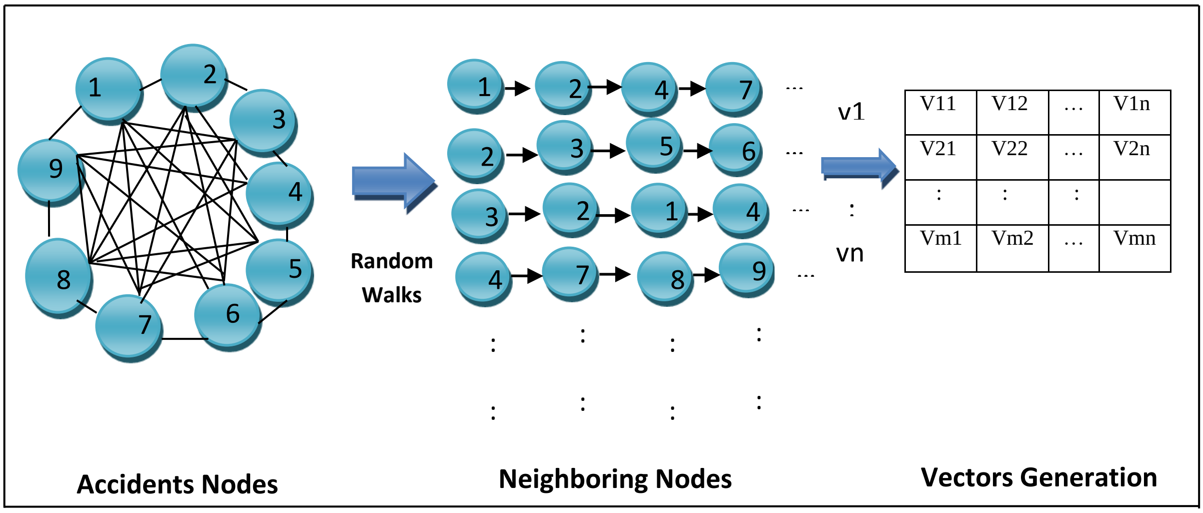
```
3.      G ← add node with some features (j) to node
4.      for each one node with other nodes
5.          Add edges between n1 and n2
6.          Compute the Euclidean distance between selected attributes of one
            node with other as edge label (weights)
7.   Return G(v, d)
End
```

3.6.2 Deep Graph Learning By Using Node2Vec Algorithm

To represent the nodes of the graph in a low-dimensional vector, we used the Node2Vec algorithm, which generates fixed length vectors of features based on the given length to simplify the implementation of Machine Learning (ML) algorithms, and exploring the varied neighborhoods using a biased random walk.

Random walks are computationally efficient in terms of both space and time requirements. It is important to control the information to be generated from the graph by using Node2Vec algorithm, this algorithm based on hyper-parameters first one is the output dimension have to decide the output dimension of the nodes embedding, the second parameter is the number of walks is mean how many time I'm going to start a random walk from each node in the graph, the third is the walk length means the number of nodes are in each random walk.

Biased random walk that can tradeoff between local and global view of the graph in two classic strategies to generate a neighborhood of a given node where local is (BFS) and the global is (DFS), this neighborhood generated randomly and the length of vectors representations for each node dependent on the walk length parameter. This representation the neighboring nodes and the vectors generations shown in figure 3.10 where vnm represent neighboring nodes for each node in graph as vectors representation. The algorithm mentioned in chapter2 section 2.5.1 Algorithm 1 Node2Vec algorithm.



V11	V12	…	V1n
V21	V22	…	V2n
:	:	:	
Vm1	Vm2	…	Vmn

Figure (3.10) Random walks generate neighboring nodes and represent this as vectors.

Algorithm (3.9): Node2Vec algorithm

Input: The Graph that Generated from Road Crash Dataset (G), window size, embedding dimension, walks number, walk length

Output: vertex representation matrix

```
Begin:
    Initialize all the elements in graph G
    For i=0 to  walk length
        For each vertex in G
            P= Random Walk(G, vertex,  walk length)
            Skip-Gram (vi, P, window)
        End for
    End for
End
```

3.7 ANN Classifier and Predictor Stage

Artificial Neural Network (ANN) model can be either for supervised or unsupervised classification. In this thesis used supervised because the target exist and it is classified for five class of severity.

3.7.1 ANN Model with Filtering Methods

ANN have three types of layers (input, output, and one or more hidden layers). The input layer used features selected by feature selection techniques, hidden layer is the core of ANN model contain on many neurons, where the node of the hidden layer is calculated at first from summation the weighted of input and the weights are generated randomly then this activation it is commonly used in hidden the ReLU. For any layer after the first hidden layer, the input is output from the previous layer. The main mathematical calculations occur in neurons to process the input and provide an appropriate output, in the output layer using the softmax to obtain the vector Y represents the pattern classification. ReLU and softmax functions are shown in chapter2 section 2.6.2 of activation functions. Algorithm 3.10 for ANN show the details. Figures (3.11, 3.12, and 3.13) show the ANN model with filtering methods. The chi-squared same IG.

3.7.2 ANN Model with Extracted Features from DGL

As above section, ANN has three types of layers (input, output, and one or more hidden layers). The input layer used features that extracted from deep graph learning by using Node2Vec algorithm, where take many forms to the input of vectors length 8,64,784 and using the input given the high accuracy, hidden layer is the core of ANN model contain on many neurons, where the node of the hidden layer is calculated at first from summation the weighted of input and the weights are generated randomly then this activation it is commonly used in hidden the ReLU.

For any layer after the first hidden layer, the input is output from the previous layer. The main mathematical calculations occur in neurons to process the input and provide an appropriate output, in the output layer using the softmax to obtain the vector Y represents the pattern classification. ReLU and softmax functions are shown in chapter2 section 2.6.2 of activation functions, building another algorithm same 3.10 ANN algorithm but with different input (extracted features from DGL) . Figure (3.14) show the ANN model with DGL features.

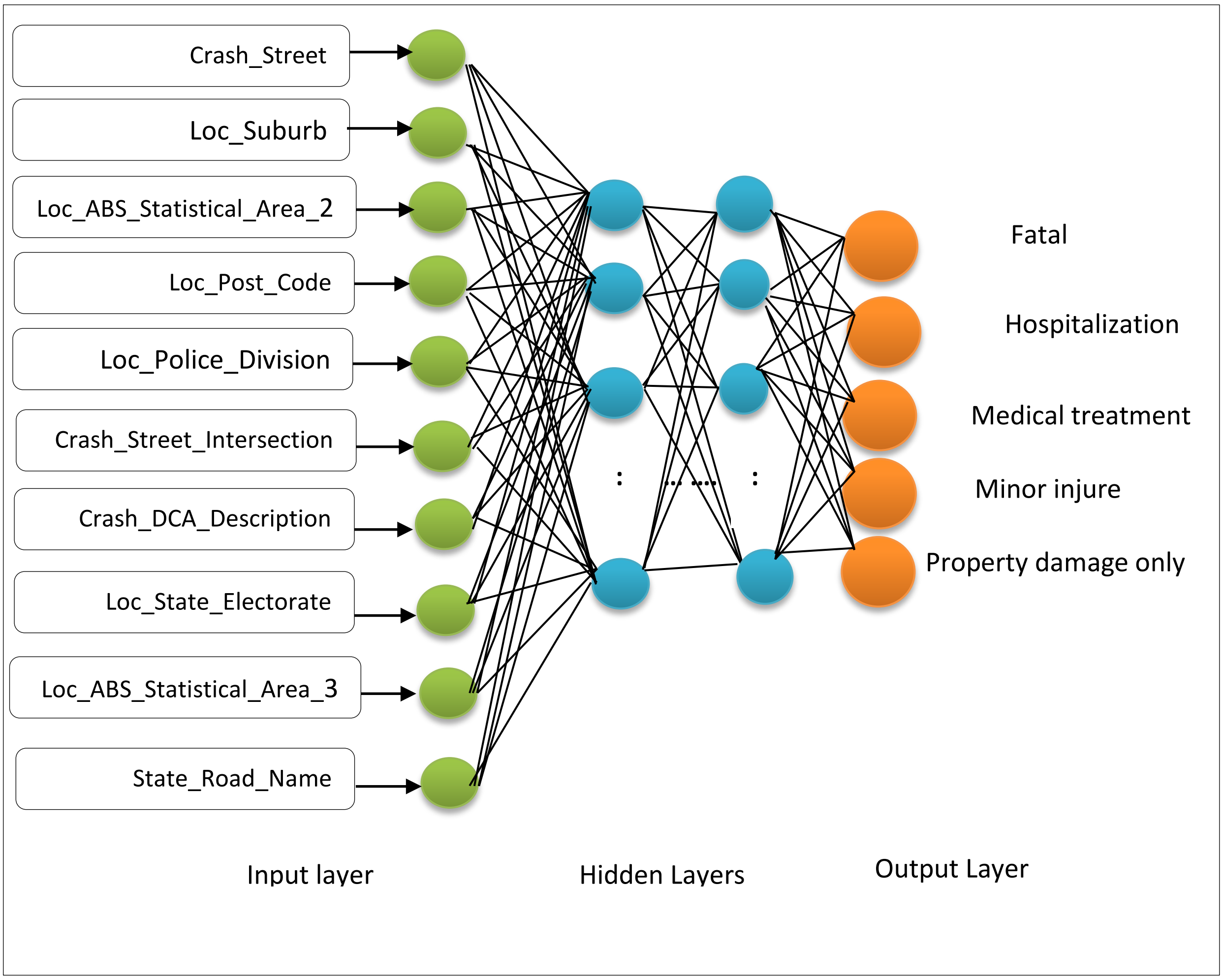


Figure (3.11) ANN architecture with IG selected features

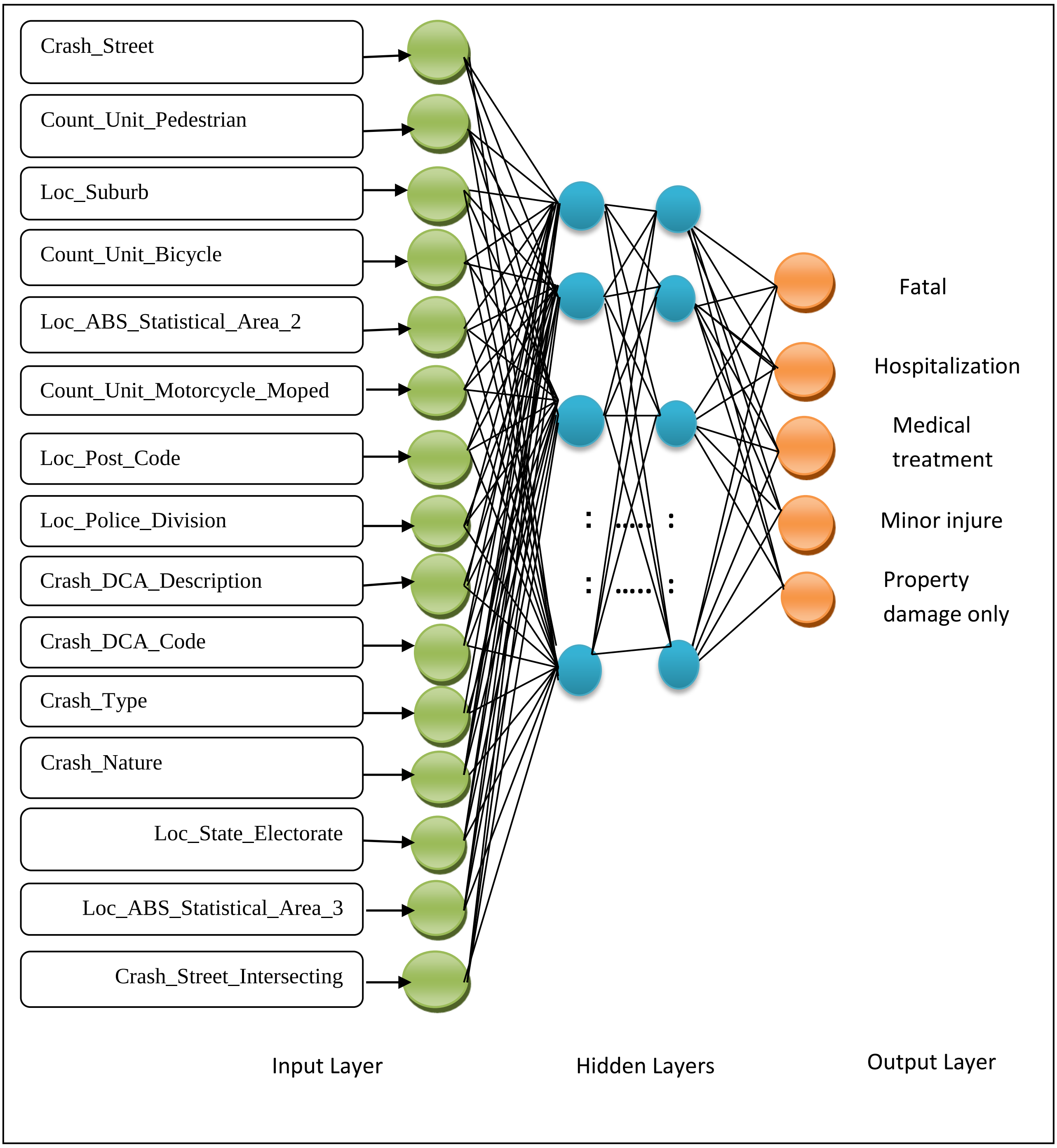


Figure (3.12) ANN architecture with GR

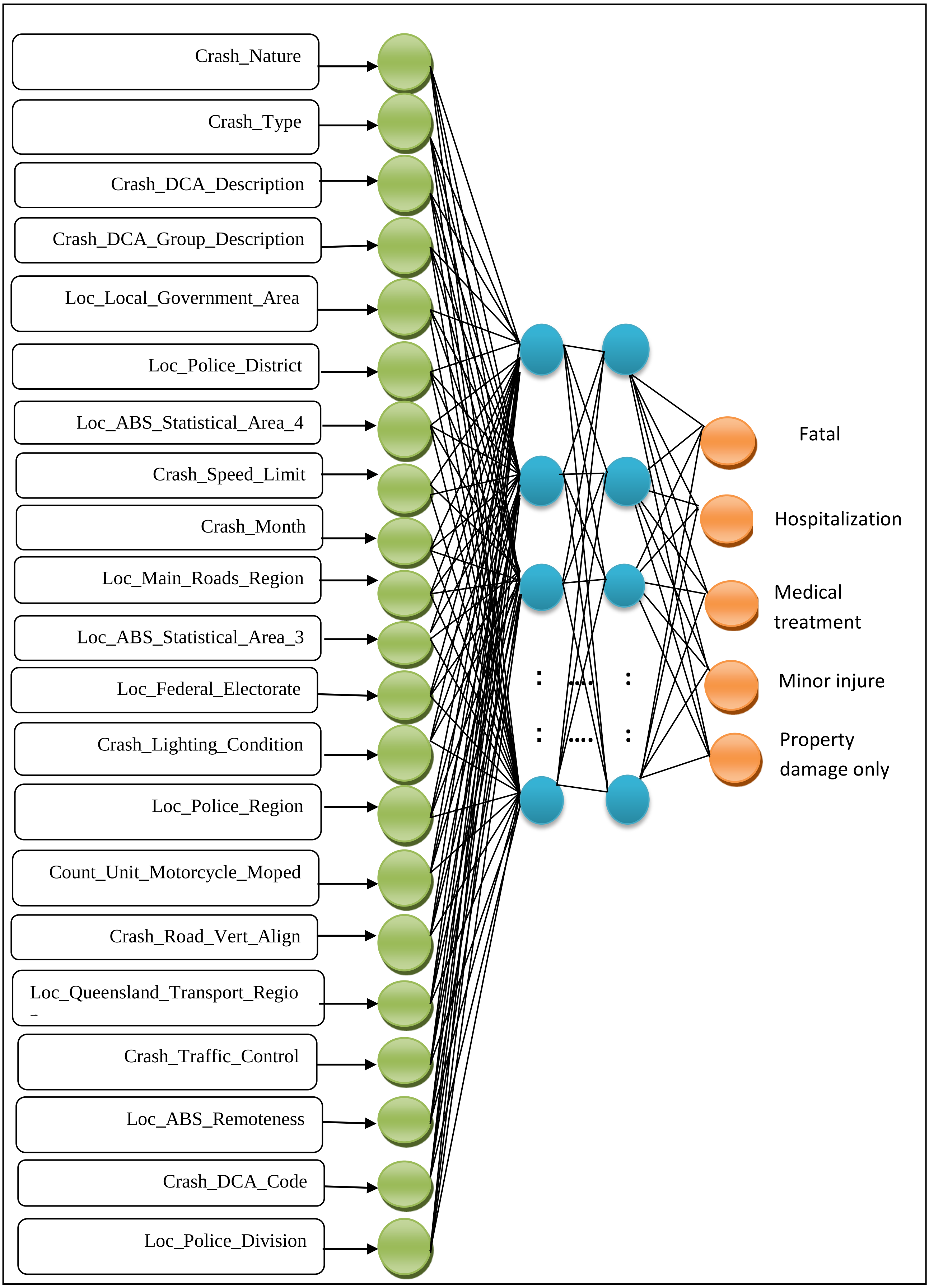
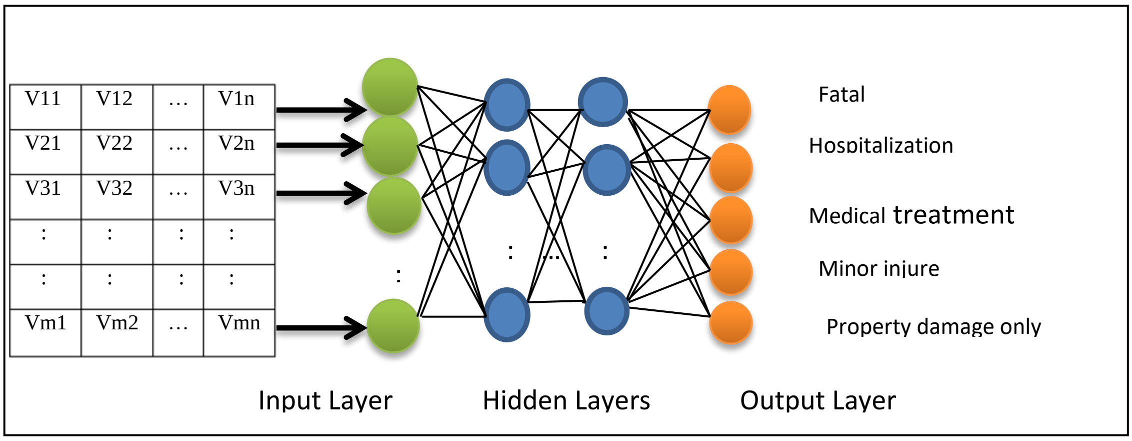


Figure (3.13) ANN architecture with Relief-F selected features



V11	V12	…	V1n
V21	V22	…	V2n
V31	V32	…	V3n
:	:	:	:
:	:	:	:
Vm1	Vm2	…	Vmn

Figure (3.14) ANN architecture with extracted features from DGL

Algorithm (3.10): ANN algorithm

Input: Array of features selected from one of the filter feature selection methods

Output: predicted class

Begin:

```
for i = 1 to n          // where n: number of factors (attributes) in dataset
    for j = 1 to m      // where m: number of instances (records) in dataset
        (calculation of output value of the hidden layer nodes). The input value of nodes in hidden layer according to the equation($s_i = w_0 + \sum_{j=1}^{n} w_{ij} x_j$)
        $h_j = f(s_i)$      // where f is ReLU activation function according to the equation (2.21)
    End for j
End for i
for i = 1 to k          // where k: number of nodes in output layer
    for j = 1 to p      // where p: number of nodes in hidden layer
        Calculation of the output value of the nodes of output layer). The input values of nodes of output layer k are according to the equation($s_{2ij} = w_0 + \sum_{j=1}^{p} w_{ij} h_j$)
        $m_j = f(s_{2ij})$      // where f is softmax activation function according to the equation (2.27)
    End for j
End for i
```

End

3.8 CNN Classifier and Predictor System

CNN can be used with numerous computer vision applications due to their observed effectiveness to offer satisfying results. The proposed methodology of CNN of multi-layer neural consists network that each input will pass through series of Convolution Layer, Pooling, Fully Connected layers (FC). All CNN models are of the same structure which is the best one according to trial and error, where the proposed structure is the best one between the numerous structure experiment.

3.8.1 CNN model with Filtering Methods

Since the features chosen from the filtering methods are a fixed-length vector that cannot be reshaped into a 2D matrix, we used a 1D convolutional neural network, but it did not have satisfactory results, so it was not mentioned as a working algorithm in this thesis.

3.8.2 CNN Model with Extracted Features from DGL

Let's begin by convoluting a matrix with just one convolution kernel, where the kernel is smaller than the inputs and used for the whole input matrix. The input is the embedding vectors that generated from deep graph learning by using node2vec algorithm, where the length according to the dimension specified by the algorithm is either 64, 128, and 784, then reshape these vectors to 2D array because using the 2D Conv to become ((8, 8, 1), (16, 8, 1), (28, 28, 1)). We passing the kernel on the top of the input matrix, for computing the product between the numbers at the same location in the kernel and the input, then summing these products together for get a single number for each kernel move. We continue to move the kernel from top to bottom and from left to right until complete the convolution process.

After getting the features map from this layer, then made a down-sampling for these features. The pooling layer is responsible for decreasing the dimensionality of the convolved feature. By using the max-pooling to return the maximum value from the portion of the input matrix covered by the kerne1.

The convolution layer and the pooling layer, form together the i-th layer of a CNN model. The number of these layers can be increased to obtain more information depending on the complexity of the data. We were able to get the model to understand the features. For classification, we'll flatten the final output and feed it into a regular neural network, then apply Softmax function to classify accident severity with probabilistic values between 0 and 1. the CNN structure used is shown in figure 3.15.

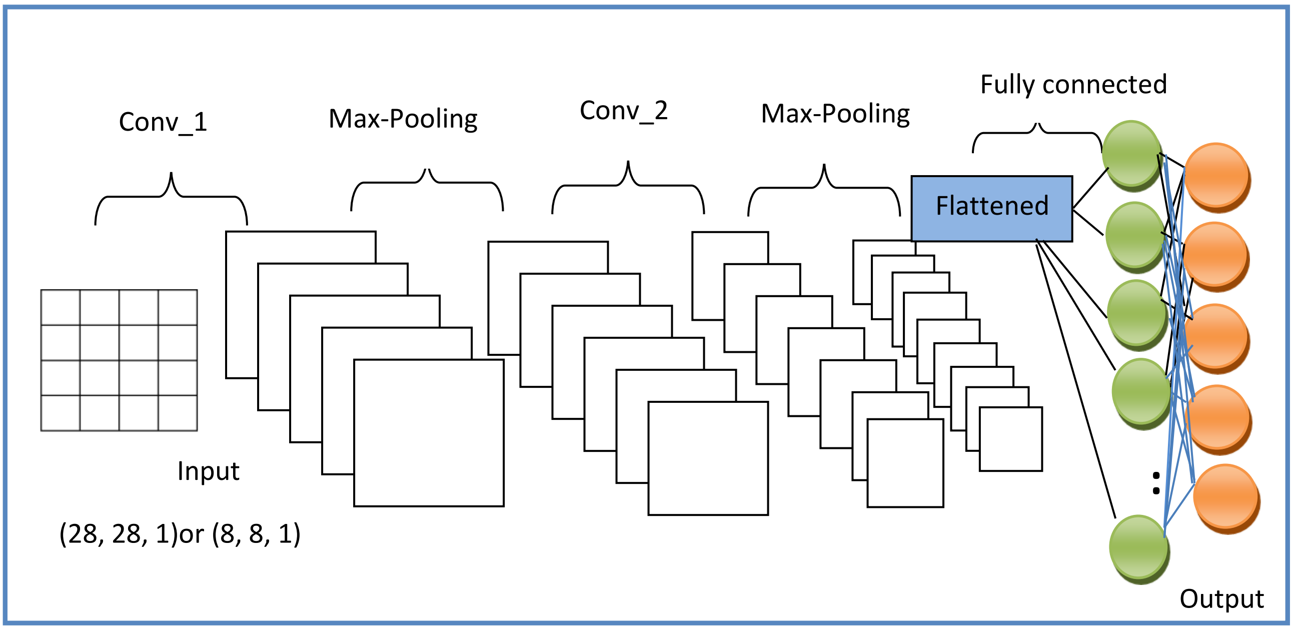


Figure (3.15) CNN Architecture

The convolution process is shown as following in the Algorithm3.11.

Algorithm (3.11): convolution operation algorithm

Input: the features extracting from graph learning

Output: important features map

Begin:

```
For k from 1 to no. of filters    // same filter on all array
  Initialize mask (W) randomly, Initialize bias (b) equal to 1
  For i from 1 to row
    For j from 1 to column
      Net=0
      For m from i-d to i+d      // rows and cols. of kernel, d specify kernel size
        For n from j-d to j+d    // d=1 equal to L*L kernel (W) of size 2*2
          net = net +[ V(m,n) * W(m,n) + b]
        End n
      End m
      F(net) = Max(net, 0)   // Relu activation function
      Feature-map[i,j,k] = f(net) // k feature map resulted that equal to the number of repeated filters where each feature map correspond a specific filter
    End j
  End i
End k
```

End

Accordingly, the generated feature map resulted from the convolution layer entered into the pooling layer to down-sampling it. The pooling algorithm is shown in Algorithm3.12.

Algorithm (3.12): pooling operation algorithm

Input: the features map resulted from convolution operation

Output: down sampled features map

```
Begin:
K1=0   //initial row index the features map
For  r from 1 to row of features map
    K2=0    //initial column index the features map
    For c from 1 to column of features map
        max= feature-map(r, c)
        For i from r to r+1
            For j from c to c+1
                If feature-map(I, j)>max
                    max=feature-map(i, j)
            End j
        End i
        Down-sampled-features[k1, k2]=max
        K2=k2+1
    End c
    K1=k1+1
End r
End
```

Chapter Four

The System Results

4.1 Introduction

The proposed system illustrated in the previous chapter has been tested on different parameters values. The results of the system stages are described in this chapter. The dataset of crash data from Queensland roads have been applied as case study to determine the behavior of these accidents by using ANN and CNN classifier models.

4.2 System Requirements

Hardware: Processor Intel i5, RAM 6GB, Storage 320 GB, Freq.2.50GHz.

Operating System: Ubuntu16.04 (64) bit.

Programming Language: Python3.6.

The importance python Libraries are

Tensorflow➔Pandas➔numpy➔Networkx➔matplotlib➔sklearn➔keras and another.

Waikato Environment for Knowledge Analysis (WEKA3.6): Machine Learning Package.

4.3 Description of Road Crash Dataset

The dataset of crash data from Queensland roads are available at Queensland government. The dataset involves many features that contain information about accident characteristics of Queensland for all road traffic accidents reported since January 1 in year 2001 to 31 December 2018 such as crash longitude, crash latitude, crash nature, crash type [7]. This dataset is divided into six categories as databases. First table is road crash locations contains 52 attributes, 328247 instances, second table is road casualties contains 7 attributes, 22372 instances, third table is vehicle types contains 12 attributes, 2195 instances, four table is Driver demographics contains 16 attributes, 14198 instances, five table is Seatbelt restraints and helmet use contains 8 attributes, 25689 instances, six table is Factors in road crashes contains 13 attributes, 3277 instances. For other details look at the table (4.1).

Table (4.1): A brief description of each category

Name of Category	*Description*
Road crash locations	Road crash location contains 52 attributes, 328247 instances.
Road casualties	Road casualties contain 7 attributes, 22372 instances.
Vehicle type	Vehicle types contain 12 attributes, 2195 instances.
Seatbelt restraints and helmet use	Seatbelt restraints and helmet use contains 8 attributes, 25689 instances.
Driver demographics	Driver demographics contain 16 attributes, 14198 instances.
Factors in road crashes	Factors in road crashes contain 13 attributes, 3277 instances.

4.4 Preprocessing

Preprocessing stage involves three important steps must be performed on our dataset: data cleaning, standardization and transform the categorical features to numerical. Table (4.2) shows a sample from dataset.

Table (4.2): Sample from dataset

Crash Severity	Crash Longitude	Crash Latitude	Loc_Post_Code	Count Casualty Fatality	Count Casualty MedicallyTreated
Fatal	152,960485061	-26,685052199	4555	0	1
Hospitalization	153,038262503	-27,550775838	4107	0	0
Medical treatment	149,951620433	-22,830786718	4156	0	0
Minor injury	150,515693948	-23,353654338	4032	0	1
Property damage only	151,934353976	-27,566383584	4510	0	0

In data cleaning step some features have been removed from the dataset such as the features that contain more than half of their data on zeros, which gives a fictitious accuracy of the classification result. The features like (count casualty fatality, count casualty hospitalized, count casualty medically treated, count casualty minor injury, count casualty total). Table 4.3 demonstrates a sample of the dataset after removing the features mentioned above.

Table (4.3): Data cleaning

Crash Severity	Crash Longitude	Crash Latitude	Loc_Post_Code
Fatal	152,960485061	-26,685052199	4555
Hospitalization	153,038262503	-27,550775838	4107
Medical treatment	149,951620433	-22,830786718	4156
Minor injury	150,515693948	-23,353654338	4032
Property damage only	151,934353976	-27,566383584	4510

Transformation the categorical features to numeric is performed, since most machine learning algorithms require categorical features to be transformed into numerical features, because numerical features perform better in classification models. Table 4.4 demonstrates the severity after giving a label for each individual property in it.

Table (4.4) Transformation the categorical features to numeric

Crash Severity	Crash Longitude	Crash Latitude	Loc_Post_Code
1	152,960485061	-26,685052199	4555
2	153,038262503	-27,550775838	4107
3	149,951620433	-22,830786718	4156
4	150,515693948	-23,353654338	4032

5	151,934353976	-27,566383584	4510

Because our dataset contain on spatial data (longitude and latitude) must be convert into UTM system. Table 4.5 is shown this.

Table (4.5) UTM System transformation

Crash Severity	Crash Longitude	Crash Latitude	Loc_Post_Code
1	9045709.40105	19667969,38577	4555
2	9138518.67799	19929838.12220	4107
3	9117211.22009	19937350.64279	4156
4	9120976.94208	19878852.58855	4032
5	9103482.54707	19780767.56925	4510

For our deep learning or machine learning model to work well, the data need to have the same scale in terms of the attributes, to prevent bias in the results. The method can be used to standardize the attribute value is the mean equal to zero and standard deviation equal to 1. Table 4.6 shows the sample after standardization.

Table (4.6) Standardization Process

Crash Severity	Crash Longitude	Crash Latitude	Loc_Post_Code
-1.41421356	1.01914148	-0.52376572	1.30534278
-0.70710678	1.08104878	-0.94063372	-0.76106558
0	-1.3757775	1.33216139	-0.53505216
0.70710678	-0.92680071	1.0803873	-1.10700448
1.41421356	0.20238795	-0.94814925	1.09777944

4.5 Feature Selection

To find out the patterns that cause accidents and reduce the dimensions of the data. Also, the data set contains factors that are not important in the classification process. For this reason, feature selection methods are important for finding the features that are most important and related to the target (crash severity) that classified into five class (Fatal, Hospitalization, Medical Treatment, Minor injure, Property damage only).

4.5.1 Filter Methods

Filter methods are statistical methods to select subset features from the data crash dataset dependent on the relationship between the attribute and target. Four techniques of filter feature selection methods have been used in the thesis: Information-Gain, Gain-Ratio, Relief-F and Chi-Squared, these methods also known as ranking methods because dependent on the ranker search method for giving each attribute ranked value.

These methods initially give a rank value for each feature, and then the arithmetic mean of the ranked values is calculated. The mean value is used as a threshold, and this process is repeated for all the methods mentioned above. The threshold value for each method is shown in table (3.2) in chapter 3.

Tables (4.7), (4.8), (4.9), (4.10) from application the Information-Gain, Gain-Ratio, Relief-F and Chi-Squared respectively.

Table (4.7) Ranked attributes from applied IG.

Ranked Valu	Feature No.	Feature Name
1.3588	10	Crash_Street
1.0959	13	Loc_Suburb
0.8725	21	Loc_ABS_Statistical_Area_2
0.5519	15	Loc_Post_Code
0.419	16	Loc_Police_Division

0.3142	11	Crash_Street_Intersecting
0.2942	37	Crash_DCA_Description
0.2487	25	Loc_State_Electorate
0.2237	22	Loc_ABS_Statistical_Area_3
0.164	12	State_Road_Name
Selected Features: 10,13,21,15,16,11,37,25,22,12 : 10		

Table (4.8) Ranked attributes from applied GR.

Ranked value	Feature No.	Feature Name
0.1553	10	Crash_Street
0.1533	45	Count_Unit_Pedestrian
0.1298	13	Loc_Suburb
0.1249	44	Count_Unit_Bicycle
0.108	21	Loc_ABS_Statistical_Area_2
0.1062	41	Count_Unit_Motorcycle_Moped
0.077	15	Loc_Post_Code
0.0639	16	Loc_Police_Division
0.0596	37	Crash_DCA_Description
0.0576	36	Crash_DCA_Code
0.0563	7	Crash_Type
0.0475	6	Crash_Nature
0.0402	25	Loc_State_Electorate
0.0378	22	Loc_ABS_Statistical_Area_3
0.0377	11	Crash_Street_Intersecting
Selected attributes: 10,45,13,44,21, 41,15,16, 37,36,7,6,25,22,11 : 15		

Table (4.9) Ranked attributes from applied Relief-F.

Ranked value	Feature No.	Feature Name
0.0299	6	Crash_Nature
0.02346	7	Crash_Type
0.02305	37	Crash_DCA_Description
0.02292	38	Crash_DCA_Group_Description
0.02032	14	Loc_Local_Government_Area
0.02015	17	Loc_Police_District
0.01988	23	Loc_ABS_Statistical_Area_4
0.01905	30	Crash_Speed_Limit
0.01692	3	Crash_Month
0.01501	20	Loc_Main_Roads_Region

0.01483	22	Loc_ABS_Statistical_Area_3
0.01421	26	Loc_Federal_Electorate
0.01394	33	Crash_Lighting_Condition
0.0125	18	Loc_Police_Region
0.01174	41	Count_Unit_Motorcycle_Moped
0.01093	35	Crash_Road_Vert_Align
0.01053	19	Loc_Queensland_Transport_Region
0.01041	29	Crash_Traffic_Control
0.00997	24	Loc_ABS_Remoteness
0.00983	36	Crash_DCA_Code
0.00899	16	Loc_Police_Division
Selected attributes: 6,7,37,38,14,17,23,30,3,20,22,26,33,18,41,35,19,29,24,36,16:21		

Table (4.10) Ranked attributes from applied Chi squared.

Ranked value	Feature No.	Feature Name
2345.8807	10	Crash_Street
2185.2995	13	Loc_Suburb
1436.7574	21	Loc_ABS_Statistical_Area_2
837.3951	15	Loc_Post_Code
767.2168	16	Loc_Police_Division
547.6642	11	Crash_Street_Intersecting
476.6942	37	Crash_DCA_Description
370.0458	25	Loc_State_Electorate
323.3384	22	Loc_ABS_Statistical_Area_3
316.2585	12	State_Road_Name
Selected attributes: 10,13,21,15,16,11,37,25,22,12 : 10		

There are several threshold values tried in this thesis every one give specific number of features where take the best one for classifier result. This values are show in the figure 4.1 for filtering methods.

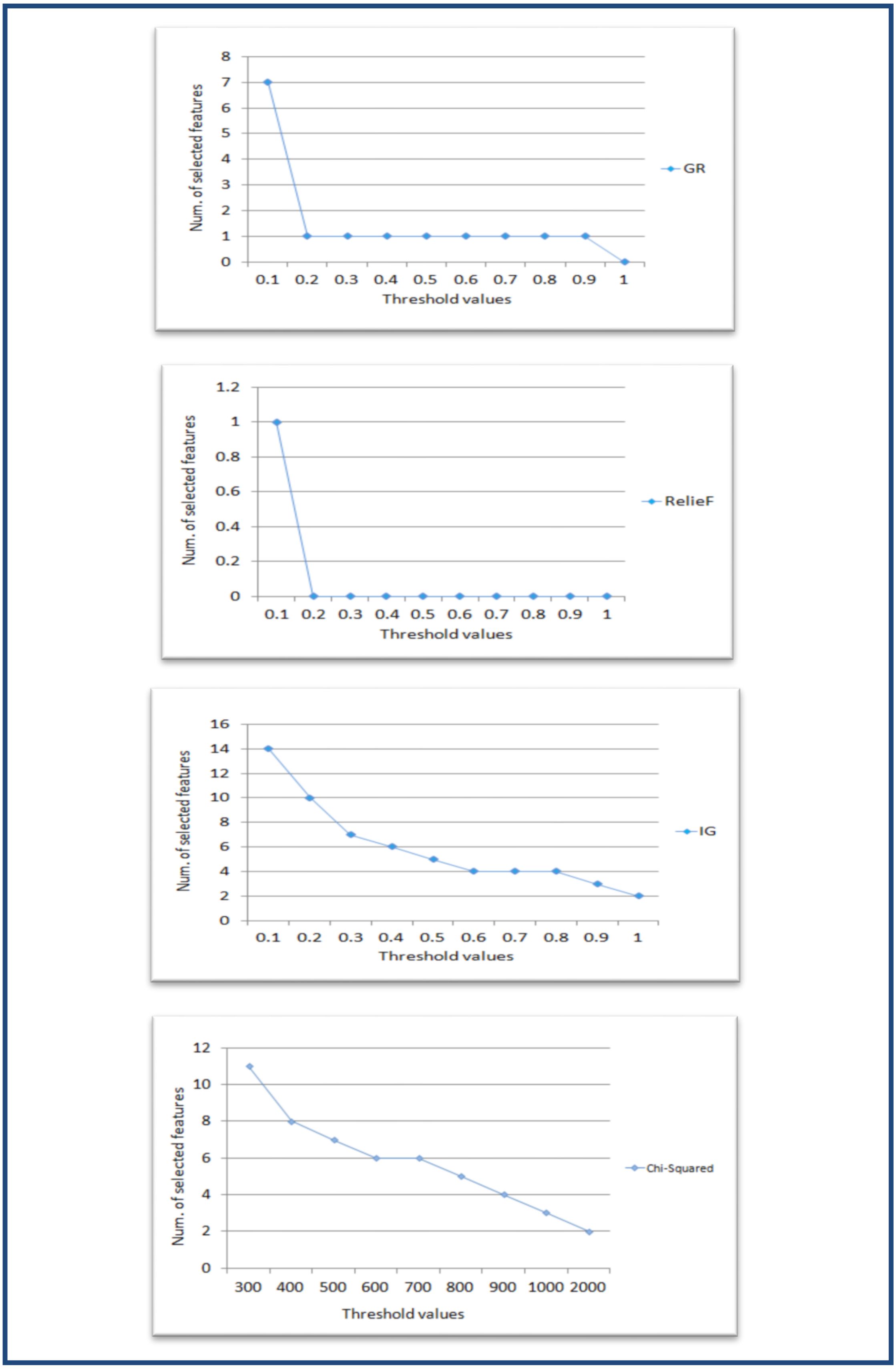


Figure (4.1) features selected by different threshold values

4.6 Graph Generation

We represent its data as a graph where every accident in dataset as nodes in graph and connect each node with other nodes and the relationship (edges) between nodes and the weight of edges calculate by using Euclidean distance of specific features such as (longitude and latitude) after converting this to the UTM system, in addition to five other features like (crash street, loc post code, crash street intersection, loc police division, loc-suburb). Figure (4.2) illustrate the graph that generate from a sample of dataset (contain 14 accident).

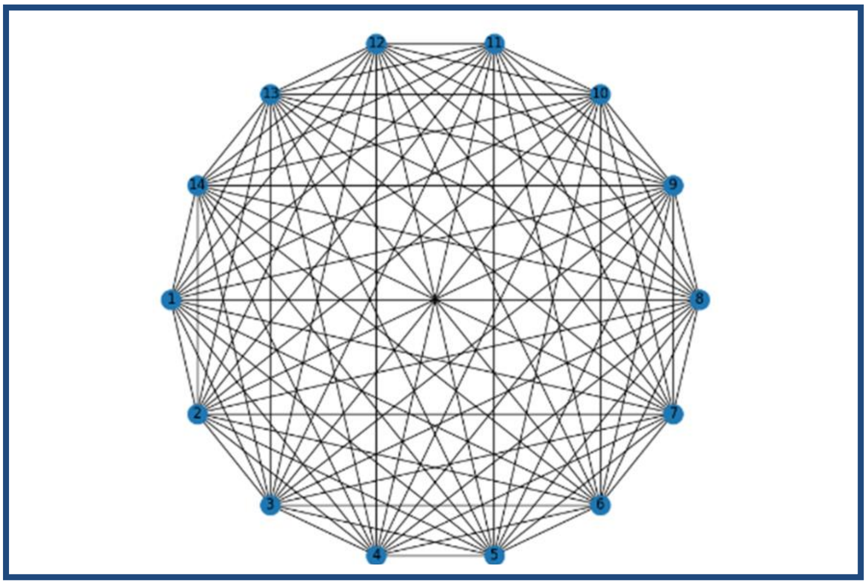


Figure (4.2) Generated Graph

4.7 Graph Learning by Using Node2Vec Algorithm

Learning can be used to convert from highly structured such as a graph, into a low-dimensional representation for node in the graph as vectors of neighboring nodes.

By using the Node2Vec algorithm a fixed-length features vector is generated from the neighboring nodes for each node in the graph randomly. Length vector is one of the parameters of the algorithm known as the output dimension, in addition to the other parameters mentioned in the chapters 2 and 3, and we will notice the effect of each parameter with the other on the used classifiers (ANN, CNN). Table 4.10 shows the neighboring nodes vectors for each node in the graph according to a sample from 14 nodes and the vector length 64.

Table(4.11) Embedded nodes representation

Dim Accidents	0	1	2	.....	62	63
6	-0.131470	0.043013	0.222784	.....	0.011749	0.147778
1	-0.139904	0.008133	0.200867	.....	0.023875	0.146130
12	-0.127695	0.025423	0.206218	.....	0.009938	0.148100
5	-0.140196	0.031723	0.214900	.....	0.012779	0.139968
11	-0.133449	0.021164	0.216468	.....	0.033051	0.141114
10	-0.124237	0.035539	0.210864	.....	-0.010802	0.150688
3	-0.122053	0.025905	0.223480	.....	0.024763	0.131629
4	-0.139986	0.034200	0.213412	.....	0.022022	0.154420
13	-0.137338	0.020259	0.206109	.....	0.010780	0.161045
7	-0.137374	0.032744	0.207215	.....	0.010179	0.159079
8	-0.126347	0.036082	0.221516	.....	0.009021	0.139578
9	-0.128470	0.043533	0.215564	.....	0.019806	0.149192
14	-0.121200	0.051447	0.215276	.....	-0.006727	0.140549
2	-0.128198	0.033167	0.220684	.....	0.021849	0.147782

4.8 Results of System Classifiers

Reliable evaluation for the classification model should be based on test data that has never seen before during training stage. According to road crash dataset is split into two parts in which the test data is 30% from original data and the rest for training.

The training ANN and CNN models are implemented using the softmax activation function for multi-classification on the road crash dataset to find the best results by using accuracy. Table 4.12 demonstrates the training accuracy results with feature selection and Tables 4.13 through 4.16 show the results of the classifiers on training data according to the accuracy metric, so, the effect of parameters of the Node2Vec algorithm on the classifiers.

Table (4.12) Accuracy of ANN with feature selection methods

Filter Method Name	Evaluation Metrics
	Accuracy (%)
Information Gain	80.05
Gain Ratio	80.23
Relief-F	80.39
Chi-Squared	80.06

Table (4.13) Training Accuracy for ANN with Extracted features from Node2Vec1

Input dimensio	Walk length	Num. walk	Evaluation Metrics
			Accuracy(%)
64	20	5	83.63
	70	5	88.43
	20	20	92.32
	70	20	94.77
	50	20	95.66

Table (4.14) Accuracy for ANN with Extracted features from Node2Vec2

Input dimension	Walk length	Num. walk	Evaluation Metrics
			Accuracy(%)
784	20	5	83.53
	70	5	90.74
	20	20	94.57
	5	70	95.33
	50	20	98.88

Table (4.15) Accuracy for CNN with Extracted features from Node2Vec1

Input dimension	Walk length	Num. walk	Evaluation Metrics
			Accuracy(%)
64 the input is (8,8,1)	70	5	80.39
	10	20	80.49
	5	70	80.95
	70	20	81.01
	50	20	81.11

Table (4.16) Accuracy for CNN with Extracted features from Node2Vec2

Input dimension	Walk length	Num. walk	Evaluation Metrics
			Accuracy(%)
784 the input is (28,28,1)	20	20	83.90
	70	5	84.41
	90	10	85.02
	50	20	85.24
	70	20	86.67

For comparing the proposed methodology with other methodology. Accordingly, the proposed method to detect high accident areas suggested by the proposed methodology in this thesis is ANN model with filtering methods features input and with features extracted from DGL, and the other methodology is the CNN model, that trained on some accidents and utilizing the model's weights to predict severity of accidents. After that, the matching method used to predict the severity of accidents is CNN. The datasets used in this thesis are Crash data from the Queensland roads dataset. Finally, the best results that have been achieved from several experiments as shown in the above tables are 80.39 % for filtering methods, 98.88 % for DGL with ANN model, and 86.67 with CNN model. Figures 4.3 through 4.5 show the highest results.

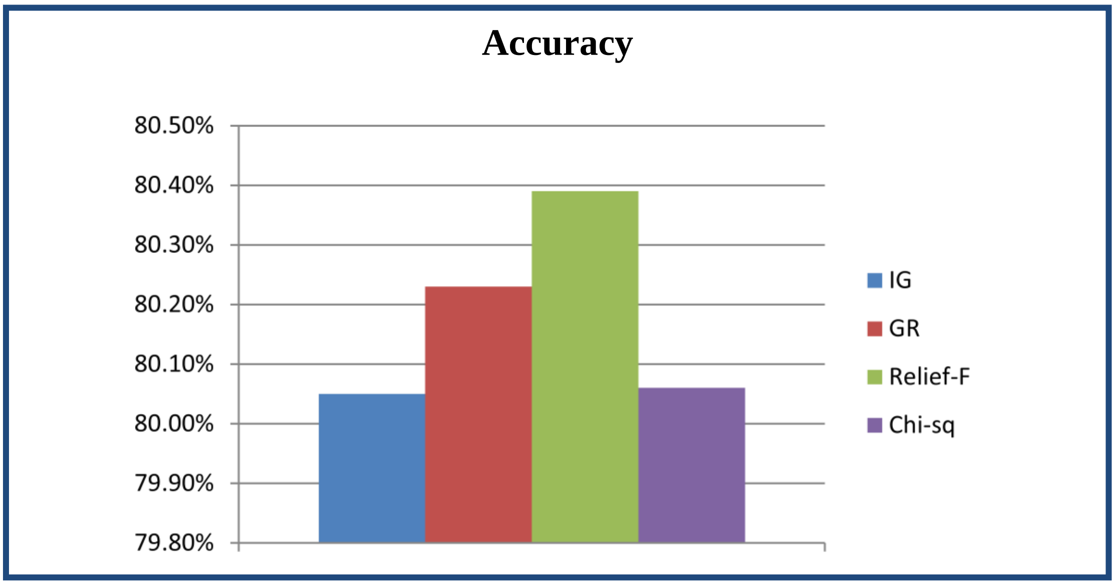


Figure (4.3) The accuracy of ANN with feature selection method

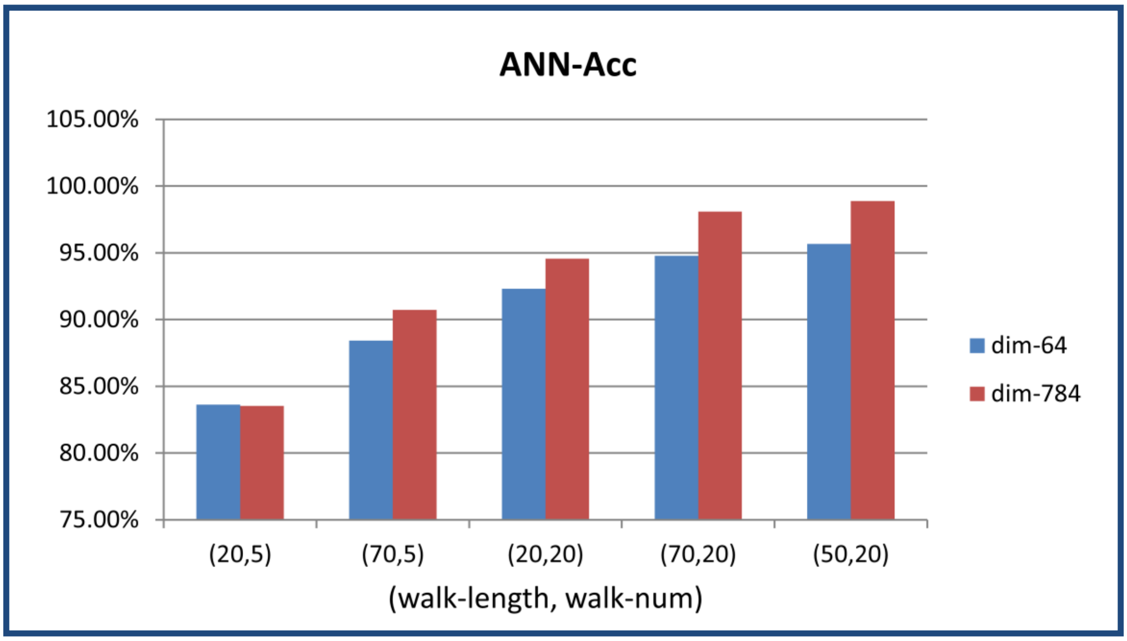


Figure (4.4) the accuracy of ANN with features from GL

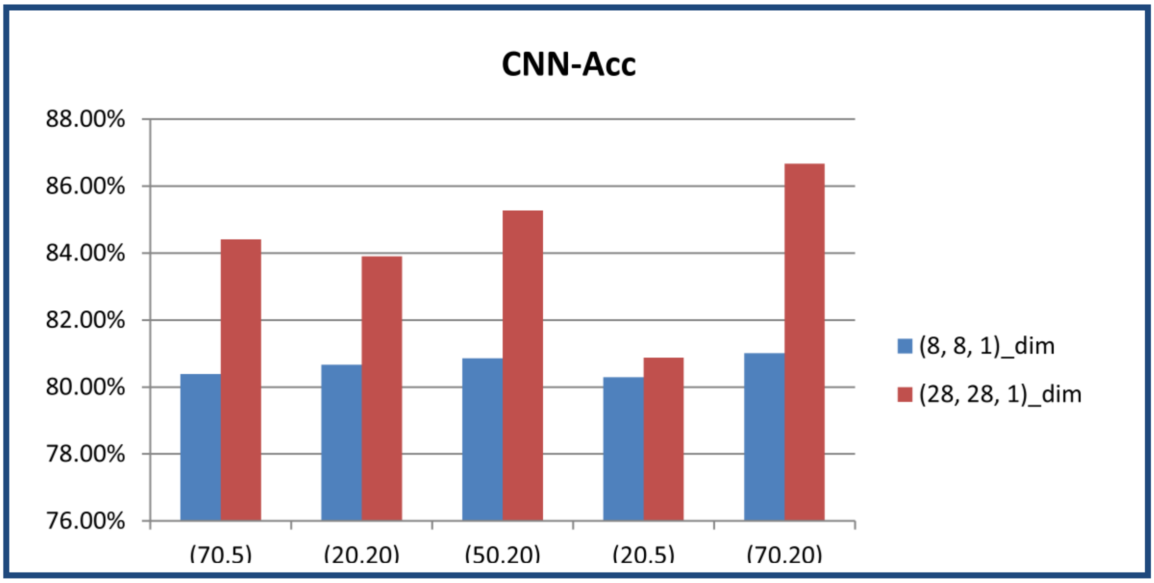


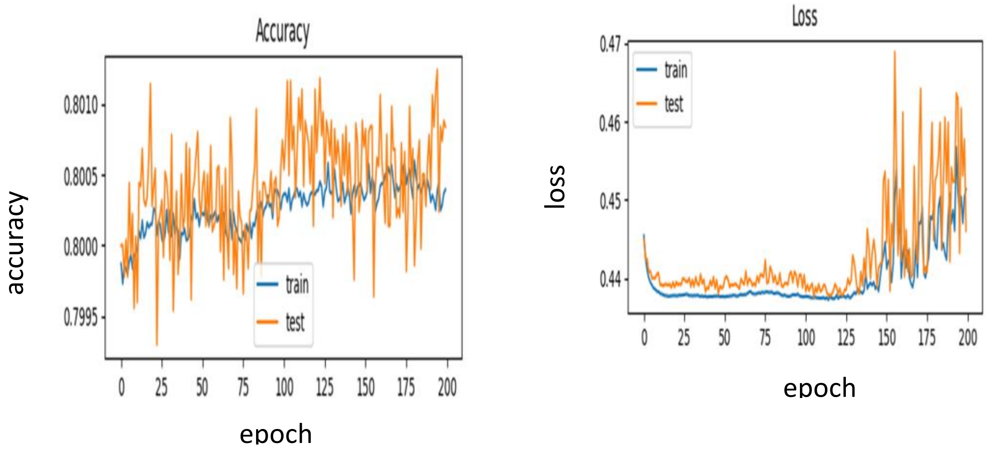
Figure (4.5) the accuracy of CNN with extracted features from DGL

4.9 System Testing

After the system has been trained on the training data of crash data from Queensland roads and saves weights of them, it becomes capable to be tested on the test data that is not seen before. There are some stages to be followed to achieve the final system decision. The testing phase consists of two main steps, which are the severity of accident prediction, high accident area detection.

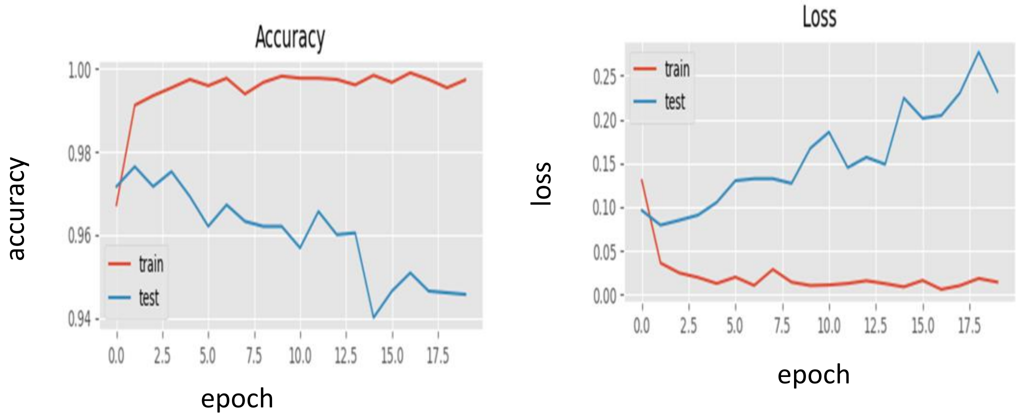
4.10 System Evaluation

After the testing dataset achieved accuracy results are 80%, 95%, and 79% for ANN with filtering methods, ANN with DGL, and CNN respectively, they are now to be matched with all training data weights of the corresponding ANN. But it is important to mention to the testing set part of the database consists of the same patterns of training data. The resulted accuracy and loss curve achieved from training the model on the crash data from Queensland roads databases appear in the following figures, Figure 4.6, Figure 4.7, and Figure 4.8 respectively, and the different between these figures are the number of epoch where, the epoch of classifier with FS methods 200 and the epoch of classifier with deep learning is 20 :



(a) (b)

Figure 4.6: The train/test curves of ANN with filtering method. (a) The train/test accuracy curve of ANN with filtering method. (b) The train/test loss curve of ANN with filtering method.



(a) (b)

Figure 4.7: The train/test curves of ANN with DGL. (a) The train/test accuracy curve of ANN with DGL. (b) The train/test loss curve of ANN with DGL.

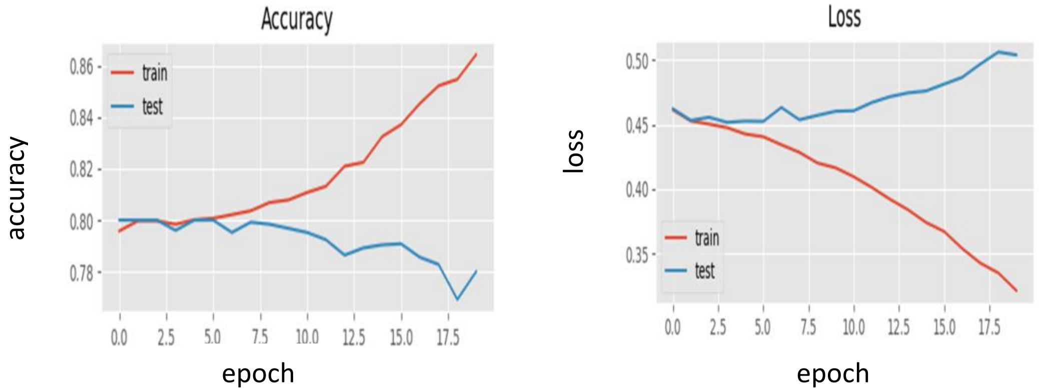


Figure 4.8: The train/test curves of CNN with DGL. (a) The train/test accuracy curve of CNN with DGL. (b) The train/test loss curve of CNN with DGL.

Chapter Five

Conclusions and Future Works

5.1 Conclusions

The most important conclusions are discovered during the design and implementation of the proposed system and achieving its results are as follows:

1- Location-based datasets like NDD can be used in my information system and its location data is very compromised.

2- In this thesis, we find that the latent features that can be extracted from the data are better than using selected related directly.

3- Data transformation into graph is a big challenge and we pass over by using some features like latitude, longitude, and some other ones to generate the graph representation in our proposed model.

4- The deep graph learning gives high accuracy to our system. The algorithm of Node2Vec and its Random Walk gives promised feature extraction techniques to extract latent features in NDD.

5- ANN feed-forward classifier gives acceptable accuracy to system evaluators by using the extracted features from the DGL better than the selected features from the filtering feature selection methods.

6- Unfortunately CNN classifier doesn't work with this environmental of such dataset, we tried to reform the features into appropriate form to be accepted in CNN classifier but unrelational data didn't get better accuracy while the training the CNN classifier.

5.2 Future works

Some suggestions may improve the results of the system and reduce the required training time if they are applied, and some suggestions for future work are presented in the following:

1- Applying the proposed system on other road crash dataset.

2- Divided the fully connected graph into sub-graphs to reduce the computation time for edges.

3- The support vector machine can be used to classify the severity of accidents.

References

[1] J. Guo, Y. Liu, L. Zhang, and Y. Wang, “Driving behaviour style study with a hybrid deep learning framework based on GPS data,” *Sustain.*, vol. 10, no. 7, pp. 1–16, 2018, doi: 10.3390/su10072351.

[2] L. Yasaswini, G. Mahesh, R. S. Shankar, and L. V. Srinivas, “Identifying Road Accidents Severity using Convolutional Neural Networks,” *Int. J. Comput. Sci. Eng.*, vol. 6, no. 7, pp. 354–360, 2018, doi: 10.26438/ijcse/v6i7.354360.

[3] J. Bärgman, “On the Analysis of Naturalistic Driving Data Department of Applied Mechanics,” no. May, 2015.

[4] R. Eenink, Y. Barnard, M. Baumann, X. Augros, and F. Utesch, “UDRIVE: The European naturalistic driving study,” *Transp. Res. Arena*, vol. 32, no. 2, pp. 1–10, 2017, [Online]. Available: http://eprints.whiterose.ac.uk/93078/1/Paper - UDRIVE the European naturalistic driving study %283%29.pdf.

[5] “4,” *meek.* .

[6] E. Muthoni Njeru and A. Imwati, “GPS & GIS In Road Accident Mapping And Emergency Response Management,” *IOSR J. Environ. Sci. Toxicol. Food Technol.*, vol. 10, no. 10, pp. 75–86, 2016, doi: 10.9790/2402-1010017586.

[7] J. Balsa-Barreiro, P. M. Valero-Mora, J. L. Berné-Valero, and F. A. Varela-García, “GIS mapping of driving behavior based on naturalistic driving data,” *ISPRS Int. J. Geo-Information*, vol. 8, no. 5, 2019, doi: 10.3390/ijgi8050226.

[8] “intro_deep[7.” [Online]. Available: https://www.freecodecamp.org/news/how-to-think-about-your-data-in-a-different-way-b84306fc2e1d/.

[9] “intro_graph[8.” [Online]. Available: https://towardsdatascience.com/node2vec-graph-embedding-method-f306ac87004e.

[10] “Dset_Crash data from Queensland roads - Datasets _ Open Data Portal _ Queensland Government.” [Online]. Available: https://www.data.qld.gov.au/dataset/crash-data-from-queensland-roads.

[11] S. Alkheder, M. Taamneh, and S. Taamneh, “Severity Prediction of Traffic Accident Using an Artificial Neural Network,” *J. Forecast.*, vol. 36, no. 1, pp. 100–108, 2017, doi: 10.1002/for.2425.

[12] F. Knoefel, B. Wallace, R. Goubran, and S. Marshall, “Naturalistic driving: A framework and advances in using big data,” *Geriatr.*, vol. 3, no. 2, pp. 1–10, 2018, doi: 10.3390/geriatrics3020016.

[13] M. I. Sameen and B. Pradhan, “Severity prediction of traffic accidents with recurrent neural networks,” *Appl. Sci.*, vol. 7, no. 6, 2017, doi: 10.3390/app7060476.

[14] M. N. Khan, A. Ghasemzadeh, and M. M. Ahmed, “Investigating the impact of fog on freeway speed selection using the SHRP2 naturalistic driving study data,” *Transp. Res. Rec.*, vol. 2672, no. 16, pp. 93–104, 2018, doi: 10.1177/0361198118774748.

[15] Z. Zhang, Q. He, J. Gao, and M. Ni, “A deep learning approach for detecting traffic accidents from social media data,” *arXiv*, 2018.

[16] Z. Yuan, X. Zhou, and T. Yang, “Hetero-ConvLSTM: A deep learning approach to traffic accident prediction on heterogeneous spatio-temporal data,” *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, pp. 984–992, 2018,

doi: 10.1145/3219819.3219922.

[17] H. Shi, H. Li, D. Zhang, C. Cheng, and X. Cao, “An efficient feature generation approach based on deep learning and feature selection techniques for traffic classification,” *Comput. Networks*, vol. 132, pp. 81–98, 2018, doi: 10.1016/j.comnet.2018.01.007.

[18] D. Xu, H. Dai, Y. Wang, P. Peng, Q. Xuan, and H. Guo, “Road traffic state prediction based on a graph embedding recurrent neural network under the SCATS,” *Chaos*, vol. 29, no. 10, 2019, doi: 10.1063/1.5117180.

[19] I. van Schagen and F. Sagberg, “The Potential Benefits of Naturalistic Driving for Road Safety Research: Theoretical and Empirical Considerations and Challenges for the Future,” *Procedia - Soc. Behav. Sci.*, vol. 48, pp. 692–701, 2012, doi: 10.1016/j.sbspro.2012.06.1047.

[20] J. Bärgman, *Methods for Analysis of Naturalistic Driving Data in Driver Behavior Research*. 2016.

[21] L. C. Datasets *et al.*, “applied sciences E ffi cient Distributed Preprocessing Model for Machine Learning-Based Anomaly Detection over,” 2020.

[22] P. J. M. Ali, “Data Normalization and Standardization,” *Bmbolstad.Com*, no. 1, pp. 1–3, 2012, doi: 10.13140/RG.2.2.28948.04489.

[23] A. Clark, “The machine learning audit- CRISP-DM Framework,” *ISACA J.*, vol. 1, pp. 42–47, 2018.

[24] K. B. and N. B. A. Jović*, “A review of feature selection methods with applications,” in *Computers in Biology and Medicine*, vol. 112, no. May, 2015, pp. 25–29.

[25] M. W. Mwadulo and Department, “A Review on Feature Selection Methods For Classification Tasks,” *Int. J. Comput. Appl. Technol. Res.*, vol. 5, no. 6, pp. 395–402, 2016.

[26] K. James Mathai and K. Agnihotri, “Optimization techniques for feature selection in classification,” *Int. J. Eng. Dev. Res.*, vol. 5, no. 3, p. 1167, 2017.

[27] R. M. and W. A. Banu, “Feature Selection for Optimization Algorithm: Literature Survey,” *J. Eng. Appl. Sci.*, vol. 12, no. 1, pp. 5735–5739, 2017.

[28] P. Kumbhar, “A Survey on Feature Selection Techniques and Classification Algorithms for Efficient Text Classification,” *Int. J. Sci. Res.*, vol. 5, no. 5, pp. 1267–1275, 2016, doi: 10.21275/v5i5.nov163675.

[29] Dinakaran S and Dr. P. Ranjit Jeba Thangaiah, “Role of Attribute Selection in Classification Algorithms,” *Int. J. Sci. Eng. Res.*, vol. 4, no. 6, pp. 67–71, 2013, doi: June 2013.

[30] Y. B. Wah, N. Ibrahim, H. A. Hamid, S. Abdul-Rahman, and S. Fong, “Feature selection methods: Case of filter and wrapper approaches for maximising classification accuracy,” *Pertanika J. Sci. Technol.*, vol. 26, no. 1, pp. 329–340, 2018.

[31] A. Sarkar, G. Sahoo, and U. C. Sahoo, “Feature selection in accident data: An analysis of its application in classification algorithms,” *Int. J. Data Anal. Tech. Strateg.*, vol. 8, no. 2, pp. 108–121, 2016, doi: 10.1504/IJDATS.2016.077484.

[32] R. J. Urbanowicz, M. Meeker, W. La Cava, R. S. Olson, and J. H. Moore, “Relief-based feature selection: Introduction and review,” *J. Biomed. Inform.*, vol. 85, pp. 189–203, 2018, doi: 10.1016/j.jbi.2018.07.014.

[33] R. P. L. DURGABAI and R. B. Y, “Feature Selection using ReliefF Algorithm,” *Ijarcce*, vol. 3, no. 10, pp. 8215–8218, 2014, doi: 10.17148/ijarcce.2014.31031.

[34] G. Gavisiddappa, S. Mahadevappa, and C. M. Patil, “Multimodal biometric authentication system using modified relief feature selection and multi support vector machine,” *Int. J. Intell. Eng. Syst.*, vol. 13, no. 1, pp. 1–12, 2020, doi: 10.22266/ijies2020.0229.01.

[35] K. K. Vasan and B. Surendiran, “Feature subset selection for intrusion detection using various rank-based algorithms,” *Int. J. Comput. Appl. Technol.*, vol. 55, no. 4, pp. 298–307, 2017, doi: 10.1504/IJCAT.2017.086017.

[36] M. Shardlow, “An Analysis of Feature Selection Techniques,” *Univ. Manchester*, vol. 14, no. 1, pp. 1–7, 2016.

[37] D. Bacciu, F. Errica, A. Micheli, and M. Podda, “A gentle introduction to deep learning for graphs,” *Neural Networks*, vol. 129, pp. 203–221, 2020, doi: 10.1016/j.neunet.2020.06.006.

[38] Y. L. and S. J. Annamalai Narayanan, Mahinthan Chandramohan, Rajasekar Venkatesan, Lihui Chen, “graph2vec: Learning Distributed Representations of Graphs,” *28th Mod. Artif. Intell. Cogn. Sci. Conf. MAICS 2017*, pp. 189–190, 2017, doi: 10.1145/1235.

[39] Z. Shen, F. Chen, L. Yang, and J. Wu, “Node2vec Representation for Clustering Journals and as A Possible Measure of Diversity,” *J. Data Inf. Sci.*, vol. 4, no. 2, pp. 79–92, 2019, doi: 10.2478/jdis-2019-0010.

[40] L. Meng and N. Masuda, “Analysis of node2vec random walks on networks: Node2vec random walks on networks,” *Proc. R. Soc. A Math. Phys. Eng. Sci.*, vol. 476, no. 2243, 2020, doi: 10.1098/rspa.2020.0447.

[41] A. Grover and J. Leskovec, “Node2Vec,” pp. 855–864, 2016, doi: 10.1145/2939672.2939754.

[42] F. S. Rizi and M. Granitzer, “Properties of vector embeddings in social networks,” *Algorithms*, vol. 10, no. 4, 2017, doi: 10.3390/a10040109.

[43] I. M. Nasser and S. S. Abu-Naser, “Artificial Neural Network for Predicting Animals Category,” *Int. J. Acad. Appl. Res.*, vol. 3, no. 2, pp. 18–24, 2019.

[44] Y. Guo, X. Cao, B. Liu, and M. Gao, “Solving partial differential equations using deep learning and physical constraints,” *Appl. Sci.*, vol. 10, no. 17, 2020, doi: 10.3390/app10175917.

[45] I. M. Nasser, M. O. Al-Shawwa, and S. S. Abu-Naser, “A Proposed Artificial Neural Network for Predicting Movies Rates Category,” *Int. J. Acad. Eng. Res.*, vol. 3, no. 2, pp. 21–25, 2019, [Online]. Available: www.ijeais.org/ijaer.

[46] O. M. Al-Mubayyed, B. S. Abu-Nasser, and S. S. Abu-Naser, “Predicting Overall Car Performance Using Artificial Neural Network,” *Int. J. Acad. Appl. Res.*, vol. 3, no. 1, pp. 1–5, 2019.

[47] and S. S. A.-N. Nesreen Samer El_Jerjawi, “Diabetes prediction using artificial neural network,” *Int. J. Adv. Sci. Technol. Vol.121*, vol. 121, pp. 55–64, 2018, doi: 10.1016/B978-0-12-819061-6.00014-8.

[48] N. Y. A. Yadav and M. Kumar, *SPRINGER BRIEFS IN APPLIED SCIENCES AND An Introduction to Neural Network Methods for Differential Equations*. 2015.

[49] Z. Wang, “The Applications of Deep Learning on Traffic Identification,” *Black Hat USA*, 2015.

[50] A. Mathew, P. Amudha, and S. Sivakumari, “Deep learning techniques: an overview,” *Adv. Intell. Syst. Comput.*, vol. 1141, no. January, pp. 599–608, 2021, doi: 10.1007/978-981-15-3383-9_54.

[51] D. A. Bashar, “Survey on Evolving Deep Learning Neural Network

Architectures," *J. Artif. Intell. Capsul. Networks*, vol. 2019, no. 2, pp. 73–82, 2019, doi: 10.36548/jaicn.2019.2.003.

[52] "No Title."

[53] L. Wenqi, L. Dongyu, and Y. Menghua, "A model of traffic accident prediction based on convolutional neural network," *2017 2nd IEEE Int. Conf. Intell. Transp. Eng. ICITE 2017*, pp. 198–202, 2017, doi: 10.1109/ICITE.2017.8056908.

[54] M. A. Wani, F. A. Bhat, S. Afzal, and A. I. Khan, *Advances in Deep Learning*, vol. 57. 2020.

[55] 3 & Richard Kinh Gian Do2 & Kaori Togashi1 Rikiya Yamashita1, 2 & Mizuho Nishio1, "Convolutional Neural Networks: An Overview and Its Applications in Pattern Recognition," *Springer*, vol. 195, pp. 21–30, 2018, doi: 10.1007/978-981-15-7078-0_3.

[56] Y. Liu, Y. Zhou, S. Wen, and C. Tang, "A Strategy on Selecting Performance Metrics for Classifier Evaluation," *Int. J. Mob. Comput. Multimed. Commun.*, vol. 6, no. 4, pp. 20–35, 2014, doi: 10.4018/IJMCMC.2014100102.

[57] R. J. Cascaro, B. D. Gerardo, and R. P. Medina, "Filter selection methods for multiclass classification," *ACM Int. Conf. Proceeding Ser.*, pp. 27–31, 2019, doi: 10.1145/3366650.3366655.

[58] A. Setiawan and E. Sediyono, "The use of google maps and universal transverse mercator (UTM) coordinate in land measurement of region in different zone," *J. Theor. Appl. Inf. Technol.*, vol. 96, no. 23, pp. 8071–8080, 2018.

[59] M. Rudnicki and T. H. Meyer, "Methods to convert local sampling coordinates into geographic information system/global positioning systems (GIS/GPS)-compatible coordinate systems," *North. J. Appl. For.*, vol. 24, no. 3, pp. 233–238, 2017, doi: 10.1093/njaf/24.3.233.

Appendix-A/ Features Description of Location Table

No.	Attribute Name	Description
1	Crash_Ref_Number	Contain id number of crash .
2	Crash_Severity	Contains how dangerous the crash like (Property damage only, Hospitalization , Medical treatment, Minor injury).
3	Crash_Year	Contain the crash years from the 2001 to 2018 .
4	Crash_Month	Contains the month in which the incident occurred.
5	Crash_Day_Of_Week	Contains any day of the week in which the incident occurred.
6	Crash_Hour	Contains at any hour the accident occurred.
7	Crash_Natur	Contains the nature of the crash type Is it Angle, Hit object, Rear-end, Overturned, Sideswipe, Fall from vehicle, Hit pedestrian, Hit parked vehicle, Hit animal, Collision - miscellaneous.
8	Crash_Type	Contains the crash type Is it Multi-Vehicle, Single Vehicle, Hit pedestrian and other.
9	Crash_Longitude_GDA9	GPS Longitude
10	Crash_Latitude_GDA9	GPS Latitude
11	Crash_Street	Contains information about the incident in any street such as Woombye - Montville Rd , Allandale St, Gateway Art Rd Ramp Xk, Hamilton Rd and other.
12	Crash_Street_Intersecting	Contain information about crash street intersecting
13	State_Road_Name	Contain information about state of road name
14	Loc_Suburb	Contain information about Loc_suburb
15	Loc_Local_Government_Area	Contain information about Location of local government area
16	Loc_Post_Code	Contain information about Location of post code
17	Loc_Police_Division	Contain information about Location of police division
18	Loc_Police_District	Contain information about Location of police district
19	Loc_Police_Region	Contain information about Location of police region
20	Loc_Queensland_Transport_Region	Contain information about Location of Queensland transport region
21	Loc_Main_Roads_Region	Contain information about the location of main road region
22	Loc_ABS_Statistical_Area_2	Contain information about Loc ABS statistical area _2
23	Loc_ABS_Statistical_Area_3	Contain information about Loc ABS statistical area _3
24	Loc_ABS_Statistical_Area_4	Contain information about Loc ABS statistical area _4

No.	Attribute Name	Description
25	Loc_ABS_Remoteness	Contain information about Loc ABS remoteness
26	Loc_State_Electorate	Contain information about Loc state electorate
27	Loc_Federal_Electorate	Contain information about Federal Electorate
28	Crash_Controlling_Authority	Contain information about controlling Authority
29	Crash_Roadway_Feature	Contain information about the feature of road way such as (No Roadway Feature, Intersection - T-Junction, Intersection – Cross, Intersection – Roundabout).
30	Crash_Traffic_Control	Include information about Crash of Traffic Control such as (Give way sign, No traffic control, Operating traffic lights).
31	Crash_Speed_Limit	Contains information about the limit speed measured km/h
32	Crash_Road_Surface_Condition	Contain information about Crash Road Surface Condition such as (Sealed - Dry, Sealed - Wet, Unsealed – Dry, unknown).
33	Crash_Atmospheric_Condition	Include information about Crash Atmospheric Condition such as (clear, Raining).
34	Crash_Lighting_Condition	Include information about Crash Lighting Condition such as (Daylight, Darkness, Darkness – Lighted, Darkness - Not lighted , unknown).
35	Crash_Road_Horiz_Align	It is Curved - view open, Curved - view obscured, Straight
36	Crash_Road_Vert_Align	It is Level, Grade
37	Crash_DCA_Code	Contain information about crash DCA code
38	Crash_DCA_Description	Contain information about crash DCA description
39	Crash_DCA_Group_Description	Contain information about crash DCA group desc.
40	DCA_Key_Approach_Dir	Contain information about DCA key approach dir
41	Count_Casualty_Fatality	Count number of Casualty Fatality
42	Count_Casualty_Hospitalised	Count number of Casualty Hospitalized
43	Count_Casualty_MedicallyTreated	Count number of Casualty Medically Treated
44	Count_Casualty_MinorInjury	Count number Casualty Minor Injury
45	Count_Casualty_Total	Count the number of all casualty
46	Count_Unit_Car	How many cars are there
47	Count_Unit_Motorcycle_Moped	How many Motorcycle Moped are there
48	Count_Unit_Truck	How many Truck are there
49	Count_Unit_Bus	How many Bus are there
50	Count_Unit_Bicycle	How many Bicycle are there
51	Count_Unit_Pedestrian	How many Pedestrian are there
52	Count_Unit_Other	How many other type of unit are there

المستخلص

حوادث المرور على الطرق هي حوادث شائعة يمكن أن تؤدي إلى إصابات خطيرة وتؤثر بشكل مباشر على أفراد المجتمع. لقد تغير تركيز أبحاث السلامة المرورية من منع الإصابات في التصادم إلى التدابير المتخذة قبل وقوع الحادث لتجنب آثاره تمامًا. هذه التدابير المتخذة تحتاج إلى معرفة العوامل البيئية وغيرها من العوامل التي تسبب الحادث. لذلك ، لتحقيق هذه المتطلبات والتحليل المتعلق بها ؛ ظهرت بيانات القيادة الطبيعية (NDD) كمصدر مهم للبيانات يتمتع بصلاحية بيئية عالية.

في هذه الرسالة ، نقوم بتحليل وتوقع مدى خطورة إصابات حوادث المرور وفقًا لقاعدة بيانات كوينزلاند التي تتكون من ٣٢٨٢٤٧ سجلًا لحوادث المرور التي حدثت خلال فترة ١٩ عامًا (من ٢٠٠١ إلى ٢٠١٩). لكل حادث ٥٢ ميزة خاصة تم جمعها وقت وقوع الحادث. تُستخدم طرق filtering Feature Selection مثل (Information-Gain ، Gain-Ratio ، Relief-f، Chi-Squared)لتقليل عدد الميزات التي سيتم استخدامها لاحقًا في نموذج التصنيف ، لتصنيف خمسة أنواع من خطورة الإصابة.

تم تصميم النظام المقترح بطريقتين، اعتمد الأول مصنف ANN لتحقيق هدف التنبؤ بخطورة الحوادث باستخدام الميزات الناتجة من خطوة طرق اختيار الميزات السابقة ، والهدف الرئيسي هو رفع دقة التصنيف. النموذج المطبق هو مُصنِّف (ANN Feed-Forward) لتصنيف وتوقع خطورة الإصابة ، وكيف يمكننا استخدام هذه النتائج لتحديد مناطق الحوادث الشديدة.

الطريقة الثانية المقترحة يتم تنفيذها لتحسين التصنيف ودقة التنبؤ على نتائج تصنيف الشبكات العصبية الاصطناعية. يتم تمثيل بيانات حوادث المرور (fully connected graph) يستخدم للتعلم العميق (deep graph) باستخدام خوارزمية Node2Vec يستخدم هذا النموذج لاستخراج الميزات الكامنة من البيانات الأولية. هذه الخوارزمية تحول ال (graph) إلى مصفوفات احادية منخفضة الأبعاد حيث يتم تمثيل كل حادث مصفوفة احادية من الخصائص ، ثم تمثل مصفوفة تضمين العقدة كمدخل لمصنف ANN مرة أخرى. بعد استخدام (deep graph learning) ، أشارت الميزات الناتجة إلى تحسن كبير في دقة تنبؤ مصنف ANN ؛ حقق ٩٥.٨٨بالمئة مقارنة بالدقة السابقة باستخدام الميزات الناتجة من طرق اختيار الميزة ، كانت النتائج ٨٠.٣٩ بالمئة.

للتحقق من أداء نموذج ANN ، تم اقتراح نموذج اخر في هذه الرسالة. يُقترح نموذج CNN هنا لتصنيف وتوقع شدة الحادث. يتم استخدام كلتا الطريقتين السابقتين لاختيار و / أو استخراج الميزات الجديدة كبيانات إدخال إلى نموذج CNN المقترح. أظهرت النتائج أن الميزات المستخرجة من (deep graph learning) باستخدام نموذج ANN تتمتع بدقة أعلى من تلك التي تم الحصول عليها من نموذج CNN ، فهي تعطي ٧٨.٦٧٪ باستخدام نفس الميزات المستخرجة وفقًا للتقييمات

المطبقة على كلا المصنفين. و للأسف لم يعطي CNN دقة عالية على البيانات المستخدمة لعدم وجود ترابط او صلة مشتركة بين الحوادث كما موجود صفة الترابط بين بيانات الصورة الواحدة.

جمهورية العراق
وزارة التعليم العالي والبحث العلمي
جامعة بابل
كلية تكنولوجيا المعلومات-قسم البرمجيات

اكتشاف منطقة الحوادث باستخدام تعلم الرسم البياني العميق بناءً على بيانات القيادة الطبيعية

رسالة
مقدمة إلى مجلس كلية تكنولوجيا المعلومات - جامعة بابل وهي جزء
من متطلبات نيل شهادة الماجستير في تكنولوجيا المعلومات/برمجيات

من قبل
ميقات حسن علي حسين

باشراف
د. وضاح رزوقي بيعي

١٤٤٣هـ ٢٠٢١م