PREDICTING FATALITY RATES OF ROAD ACCIDENTS IN BANGLADESH USING ARTIFICIAL NEURAL NETWORK

By

MD. SOHEL RANA

MILON KHAN

MD. MAHIM HOSSAIN KAKUN

MD. CHAD MIA

MD. BENJEER AHMED

A thesis submitted to the Department of Civil Engineering in partial fulfillment for the degree of Bachelor of Science in Civil Engineering



Department of Civil Engineering

Sonargaon University

147/I, Green Road, Dhaka-1215, Bangladesh

Section: 21B

Spring-2024

PREDICTING FATALITY RATES OF ROAD ACCIDENTS IN BANGLADESH USING ARTIFICIAL NEURAL NETWORK

By

MD. SOHEL RANA	CE2003021105
MILON KHAN	CE2003021039
MD. MAHIM HOSSAIN KAKUN	BCE2001019085
MD. CHAD MIA	CE2003021081
MD. BENJEER AHMED	CE2003021033

Supervisor

Kushal Acharja Topu
Lecturer, Department of Civil Engineering
Sonargaon University

A thesis submitted to the Department of Civil Engineering in partial fulfillment for the degree of Bachelor of Science in Civil Engineering

Department of Civil Engineering

Sonargaon University

147/I, Green Road, Dhaka-1215, Bangladesh

Section: 21B

Spring-2024

BOARD OF EXAMINERS

The thesis titled "Predicting Fatality Rates of Road Accidents in Bangladesh using Artificial Neural Network" submitted by Md Sohel Rana, Milon Khan, Md Mahim Hossain Kakun, Md. Chad Mia, Md Benjeer Ahmed. ID: CE-2003021105, CE-2003021039, BCE-2001019085, CE2003021081, CE-2003021033 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of Bachelor of Science in Civil Engineering on 17th May 2024.

1.	…… Kushal Acharja Topu Lecturer, Dept. of Civil Engineering Sonargaon University	Chairman
2.	…… Internal / External Member	Member
3.	…… Internal / External Member	Member

DECLARATION

It is hereby declared that this thesis/project or any part of it has not been submitted elsewhere for the award of any degree or diploma.

NO:	STUDENT NAME	STUDENT ID.	SIGNATURE
1.	Md. Sohel Rana	CE 2003021105	----------
2.	Milon Khan	CE 2003021039	----------
3.	Md Mahim Hossain Kakun	BCE 2001019085	----------
4.	Md. Chad Mia	CE 2003021081	----------
5.	Md Benjeer Ahmed	CE 2003021033	----------

Dedicated

To

"Our Parents"

&

"Respectable Supervisor''

ACKNOWLEDGEMENTS

1st, we are grateful to almighty ALLAH for giving us the strength and courage that we have completed the thesis successfully. We are thankful for the financial support provided by Sonargaon University which made this research possible. We extend our sincere appreciation to our research supervisor Kushal Acharja Topu for his guidance, insights and constant encouragement throughout the course of this study. Our earnest thanks go to the individuals who participated in this study. Their willingness to contribute their time and insights has been indispensable in gathering the data necessary for our research. Finally, we would like to thank our families and friends for their understanding, encouragement and responsive support during the course of this research.

ABSTRACT

In recent years, the escalation of road accidents has emerged as a significant global concern, ranking as the ninth leading cause of death worldwide. Bangladesh, in particular, grapples with a substantial burden of road accidents, a distressing reality that demands urgent attention. Recognizing the imperative to address this pressing issue comprehensively, this research paper undertakes a meticulous analysis. The aim is to delve deeper into traffic accidents in Bangladesh, employing machine learning methodologies to discern the magnitude of these incidents. Specifically, an Artificial Neural Network (ANN) is developed, utilizing the 'relu' activation function and data spanning from 1998 to 2013 encompassing accidents and casualties. This ANN is trained to forecast the severity of accidents from 2014 to 2024. The model is trained using 70 percent of the data as the training set, with the remaining 30 percent reserved for testing purposes. Through rigorous experimentation, the model is finetuned by varying the number of neurons in a single hidden layer across 1000 epochs, identifying the configuration that yields optimal performance by using the value of RMSE (Root Mean Square Error). Subsequently, the model is trained with this optimal neuron count and applied to predict accident and casualty data from N1 to N9.

TABLE OF CONTENTS

DECLERATION iv
ACKNOWLEDGEMENT vi
ABSTRACT vii
LIST OF FIGURES Viii
LIST OF TABLES ix

CHAPTER-1 (1-4)

INTRODUCTION 1
1.1 Background and Motivations 2
1.2 Research Objectives and Overview 3
1.3 Organization of the Thesis 3

CHAPTER-2 (5-6)

LITERATURE REVIEW 5
2.1 Introduction 5
2.2 Previous Legendre Works 5
2.3 Overview 6

CHAPTER-3 (7-10)

METHODOLOGY 7
3.1 Introduction 7
3.2 Artificial Neural Networks (ANN). 7
3.2.1 Overview of ANN: 7
3.2.2 Node Character: 7
3.2.3 Network Topology 8
3.2.4 Learning 9
3.2.5 Hopfield Net 9
3.2.6 Kohonen Maps 10
3.3 Summary 10

CHAPTER-4 (11-12)

DATA COLLECTION 11
4.1 Introduction 11
4.2 Summary 12

CHAPTER-5 (13-19)

DATA ANALYSIS 13
5.1 Result: ANN Accidents 13
5.2 Result: ANN Casualty 16

CHAPTER-6 (20-21)

CONCLUSIONS AND FUTURE WORKS 20
6.1 Conclusions 20
6.2 Future Works 20

REFERENCES 22

APPENDIX (23-26)

APPENDIX A.1 23
APPENDIX A.2 24

LIST OF FIGURES

Figure 2.1 A basic model of a single node 8
Figure 2.2 Transfer function 8
Figure 2.3 (a) A General topology of ANN 9
Figure 2.3 (b) Feedforward ANN (perceptron) 9
Figure 2.3 (c) Feedback ANN 9
Figure 2.4 Hopfield network 10
Figure 2.5 Kohonen maps 10
Figure 5.1 N1 to N9 Accidents predictions from 2014 to 2024 16
Figure 5.2 N1 to N9 Casualty predictions from 2014 to 2024 19

LIST OF TABLES

Table-1 Number of Accidents on Different National Highways of Bangladesh 11
Table-2 Number of Casualties on Different National Highways of Bangladesh 12

CHAPTER – 1

INTRODUCTION

Road Accident is the most undesirable and unexpected thing to occur to a road user, though they happen quite often. Unfortunately, we can see a minatory rise in road accidents in Bangladesh, conspicuously high road accidents over the past few years. It has a massive impact on society as well as on the economy of our country as there is an immense cost of fatalities and injuries. According to a recent inspection, conducted by the Accident Research Institute (ARI) of Bangladesh University of Engineering and Technology (BUET), annually on average 12,000 lives have been taken by road accidents which lead to almost 35,000 injuries [1] [2].

This record indicates that every day, approximately 32 people are killed in Bangladesh by road accidents and it is quite devastating. Besides this, according to WHO, the economic cost of road accidents in a developing country like us is 2-3% of GDP, which is a significant loss for a country like ours. Moreover, reducing this loss has become a great matter of concern for our country now.

Some researchers mainly focused on accident detection, assessment of severity and analysis of road accident patterns. The works are related to road accident analysis like road accident factors and occurrence identification, severity analysis and so on. Previously various research works had been done regarding this issue but that's not enough and is more appropriate and accurate automated system is required in the context of Bangladesh.

In [3], M. M. L. Elahi et al. proposed a method that can learn traffic patterns through video data captured from the roadside and can track various features of a vehicle by using a vision-based technique. They used a probabilistic model Parzen Probabilistic Neural Network (PPNN). For the prediction of road accidents, they detected the anomaly of vehicles by their model. Their proposed model got an accuracy of almost 85% for detecting unnatural special occurrences or situations. As features, they just used the velocity of different vehicles and the distance between them. They did not measure the severity of a particular accident with these characteristics.

In [4], Md. S. Satu et al. analyzed N5 national highway specifically the Dhaka Banglabandha highway and found out the traffic accident pattern. For that, they used the Decision Tree induction algorithm. The performance of twelve Decision Tree classifiers was

compared by them. Among them, the CDT (unpruned) was found to be the best classifier. The AUC curve and precision of their findings were fine, but they got a low accuracy and recall. They had operated in a particular region of the N5 highway with very few reports of 892 incidents.

1.1 Background and Motivations

Road accidents pose a significant threat to public safety and well-being, particularly in densely populated countries like Bangladesh. Despite concerted efforts by authorities to improve road infrastructure and enforce traffic regulations, the incidence of road accidents remains alarmingly high, resulting in substantial human and economic costs.

Bangladesh, with its burgeoning population and rapid urbanization, faces unique challenges in managing road safety. The country's road network is characterized by a mix of modern highways and rural roads, each presenting its own set of risks and hazards. Factors such as inadequate road maintenance, poor traffic management, and non-compliance with safety regulations further exacerbate the problem.

Traditional methods of road safety management rely heavily on reactive measures, such as accident investigation and enforcement of traffic laws after incidents occur. While these approaches are essential, they often fall short of preventing accidents proactively and mitigating their severity. There is a pressing need for innovative solutions that leverage advanced technologies to predict road accidents and assess their potential severity.

Machine learning, a subset of artificial intelligence, offers promising capabilities in analyzing vast amounts of data and identifying complex patterns and relationships. By leveraging historical accident data along with various contextual factors such as weather conditions, road characteristics, and vehicle attributes, machine learning techniques can help forecast the likelihood of accidents and their potential severity.

The motivation behind this research stems from the desire to harness the power of machine learning to enhance road safety in Bangladesh. By developing predictive models that can anticipate accidents before they occur and assess their severity, policymakers and transportation authorities can take proactive measures to mitigate risks and allocate resources more effectively. Furthermore, the insights gained from this study can inform evidence-based decision-making and guide the development of targeted interventions aimed at reducing the frequency and severity of road accidents in Bangladesh.

2019 saw at least 7,855 fatalities and 13,330 injuries nationwide from 5,516 traffic incidents, according to the "Jatri Kalyan Samity" annual report. Every day, more than 4,000 people traverse the streets of Bangladesh. The country has 85 fatalities per 10,000 enlisted engine vehicles, which is likely the highest rate. That is much higher than the average rate in most Western nations. Automobile accidents do severe harm to defenseless households and drain billions from Bangladesh's economy. According to the World Health Organization (WHO), road traffic injuries cost Bangladesh's economy 2% of its GDP, or over £1.2 billion, annually. This is comparable to our completely foreign guide during a financial year. The losses include direct and indirect costs, such as medical expenses, security losses, property damage, lost family income, and gridlock.

1.2 Research Objectives

We have tried to focus on the busy highways of Bangladesh. We must obtain information from ARI before analyzing each piece of data.

The primary objectives of our study are

[1] To determine the accident characteristics of National Highways.

[2] To conduct a comprehensive analysis of accidents and causalities of the existing 15 years data and predict the 10 years data from 2014 to 2024.

The first stage is to conduct research into the pertinent sources to gain a better understanding of the various accident characteristics, especially as they affect pavement layout and structure. In addition, after evaluating prior research on the related topic of crashes, the methodology for these studies has been condensed in the following ways.

1.3 Organization of the thesis

This section should have a brief description of the thesis outline of the thesis. It should contain chapter no. with a title and brief descriptions of the content of each chapter. An example guide is provided below.

Chapter 1:

Introduction and Objective. This chapter provides the background and motivations of the research. The overall objectives and expected outcomes are also described in this chapter.

Chapter 2:

Literature Review. This chapter reviews the related works in the Literature Review field with a special focus on Introduction and previous Legendre Works.

Chapter 3:

Methodology. This chapter describes the methodology adopted to carry out the research.

Chapter 4:

Data Collection. This chapter describes the collected time and date of our data collection.

Chapter 5:

Data Analysis. This chapter summarizes the conclusions and major contributions of this study and provides recommendations for future studies.

Chapter-6:

Conclusions and Future Works. This chapter includes all the possible future works and summary of our thesis work.

CHAPTER - 2

LITERATURE REVIEW

2.1 Introduction

The analysis of fatality rates in road accidents in Bangladesh is a critical endeavor with far-reaching implications for public safety, economic development, and policy formulation. As the country continues to experience rapid urbanization and infrastructure expansion, the challenges associated with road safety have become increasingly pronounced. In this literature review, we embark on a comprehensive examination of existing research to shed light on the factors influencing road accidents and fatalities in Bangladesh.

The motivation behind this thesis work stems from the alarming statistics surrounding road accidents in Bangladesh. Despite concerted efforts to improve road infrastructure, enhance driver education, and strengthen enforcement measures, the country continues to grapple with high fatality rates on its roads. Understanding the underlying causes of these accidents is imperative for devising effective strategies to mitigate their impact and save lives.

This literature review serves as a foundational component of the thesis work, providing a synthesis of existing knowledge and insights into the complex dynamics of road safety in Bangladesh. Through a structured analysis of key themes and research findings, we aim to elucidate the interconnected factors contributing to road accidents, including road infrastructure, driver behavior, vehicle safety standards, and emergency response systems.

By critically assessing the existing literature, we seek to identify gaps in knowledge and areas where further research is warranted. Moreover, we aim to highlight the relevance and importance of the thesis work in addressing these gaps and advancing understanding in the field of road safety in Bangladesh.

Through this literature review, we endeavor to lay the groundwork for subsequent chapters of the thesis, which will delve into empirical analysis, data interpretation, and policy recommendations. Ultimately, our goal is to contribute to evidence-based decision-making and interventions aimed at reducing road accidents and fatalities, thereby fostering a safer and more sustainable transportation environment in Bangladesh.

2.2 Previous Legendre Works

Below is the list of those who have previously worked on Analyzing Fatality Rates In Road Accidents In Bangladesh.

- **KNN:** KNN is a classification algorithm which is based on feature similarity. It analyzes the data and measure the distance and similarities between data and cluster them based on K values. Distance is calculated in many ways, for this research, we used Euclidean distance measurement. The class of new input data is classified by calculating the distance between the clusters and assigned it to the closest one. [4] Euclidean distance is calculated by,

$$\boldsymbol{d(x, x') = \sqrt{(x_1 - x'_1)^2 + (x_2 - x'_2)^2 + \ldots + (x_n - x'_n)^2}}$$

- **Naïve Byes:** Naïve Byes is another classification technique based on Bayes theorem. It predicts the probability of different class based on several attributes and assigns the new class to the highest probability. The algorithm working rule:

$$\boldsymbol{Posterior\ Probability, P\left(\frac{a}{b}\right) = \frac{p\left(\frac{a}{b}\right) \times p(a)}{p(b)}}$$

The posterior probability is mainly the probability of "a" being true given that "b" is true. [5]

- **AdaBoost:** AdaBoost is mainly a boosting algorithm which is used with short decision trees. Every instance is weighted in the training dataset. At first, the weight is set to,

$$\boldsymbol{Weight, w_i = \frac{1}{n}}$$

Where wi is the it's training instance weight and n is the number of training instances. Further, the first tree is created, the performance of the tree on each training instance is used. After that, it evaluates overall errors. Next iteration weights are calculated by the errors. More weight is given where hard to predict, whereas less weight is given where easy to predict. [6]

2.3 Overview:

Below is a list of what we will discuss in this chapter and what we will discuss in the next chapter.

- **Chapter-2:** The second chapter is a review of relevant literature. This chapter explains the introduction, Previous Legendre Works, overview, etc., which have been used in comparable works previously.
- **Chapter-3:** The third chapter discussed of about, Methodology introduction, Decision Tree, and Artificial neural networks.

CHAPTER - 3

METHODOLOGY

3.1 Introduction

This section outlines the methodologies employed to analyze fatality rates in road accidents in Bangladesh. The study utilizes two distinct approaches: Decision Trees and Artificial Neural Networks (ANN).

Artificial Neural Networks are computational models inspired by the structure and function of biological neural networks. They consist of interconnected nodes organized in layers, capable of learning complex patterns and relationships from data.

Finally, a comparative analysis is conducted between the performance of Decision Trees and Artificial Neural Networks in predicting fatality rates in road accidents, considering factors such as accuracy, interpretability, and computational efficiency.

3.2 Artificial Neural Networks (ANN)

3.2.1 Overview of ANN:

The artificial neural network is modeled on the biological neural network. Like the biological neural network, the ANN is an interconnection of nodes, analogous to neurons. Each neural network has three critical components: node character, network topology, and learning rules. weight associated with each input and output, and the activation function. Network topology determines the ways nodes are organized and connected. Learning rules determine how the weights are initialized and adjusted. The following sections explain each of the components in detail.

3.2.2 Node Character:

The basic model for a node in the ANN is shown in Fig. 2.1. Each node receives multiple inputs from others via connections that have associated weights, analogous to the strength of the synapse. When the weighted sum of inputs exceed the threshold value of the node, it activates and passes the signal through a transfer function and sends it to neighboring nodes. This process can be expressed as a mathematical model [7]

$$y = f\left(\sum_{i=0}^{n} w_i x_i - T\right)$$

Where y is the output of the node, f is the transfer function, $\boldsymbol{w_i}$ is the weight of input $\boldsymbol{x_i}$, and T is the threshold value. The transfer function has many forms. A non-linear transfer function is more useful than linear ones [8], since only a few problems are linearly separable. The simplest one is the step function (see Fig. 2.2)

$$\boldsymbol{y} = \begin{cases} \boldsymbol{1 \ldots if \sum_{i=0}^{n} w_i x_i > T} \\ \boldsymbol{1 \ldots if \sum_{i=0}^{n} w_i x_i < T} \end{cases}$$

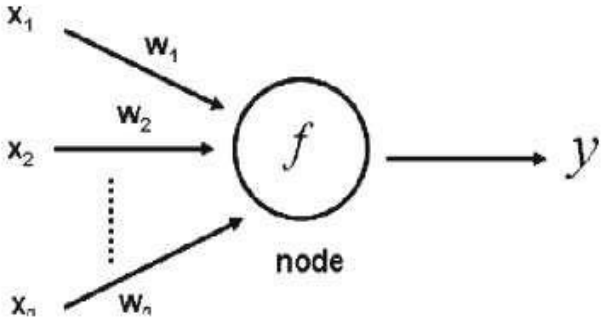


Fig-2.1: A basic model of a single node: x_i = input, w_i= weight, f= transfer function, y= output.

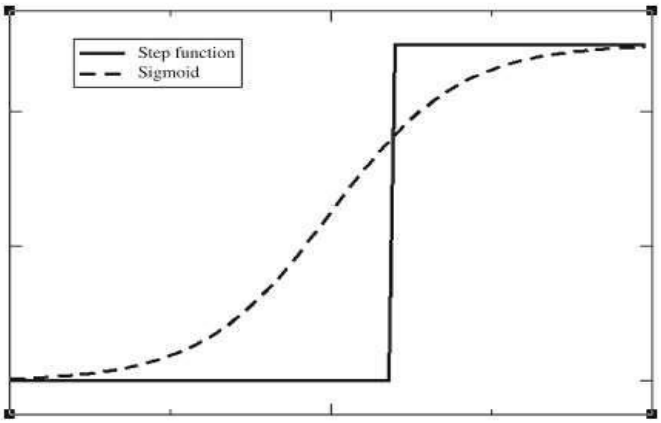


Fig-2.2: Transfer function.

$$\boldsymbol{y = \frac{1}{1 + exp(-\beta x)}}$$

3.2.3 Network Topology

Figure 2.3(a) shows the general architecture of an ANN. The nodes are organized into linear arrays, called layers. Usually, there are input layers, output layers, and hidden layers. There can be none to several hidden layers. Designing the network topology involves the determining the number of nodes at each layer, the number

of layers in the network, and the path of the connections among the nodes. Usually, those factors are initially set by intuition and optimized through multiple cycles of experiments. Also, some rational methods can be used to design a neural network. For example, the genetic neural network (GNN) uses a generic algorithm to select the input features for the neural network solving QSAR problems [9].

3.2.4 Learning

The ANN uses a learning process to train the network. During the training, weights are adjusted to desired values. The learning can be classified into two major categories supervised learning and unsupervised learning. In supervised learning, a training set, that is, examples of inputs and corresponding target outputs, is provided. The weights are adjusted to minimize the error between the network output and the correct output. Error correction methods normally have a back propagation mechanism. Let $\boldsymbol{y_{kn}}$ be the output of the $\boldsymbol{k}$ th output node at step n and $\boldsymbol{y_k}$ $*$, be the target output for the $\boldsymbol{k}$ th node. An error function can be defined as the difference betwen the node output and target output:

$$\boldsymbol{e_k = y_{kn} - y_k} *$$

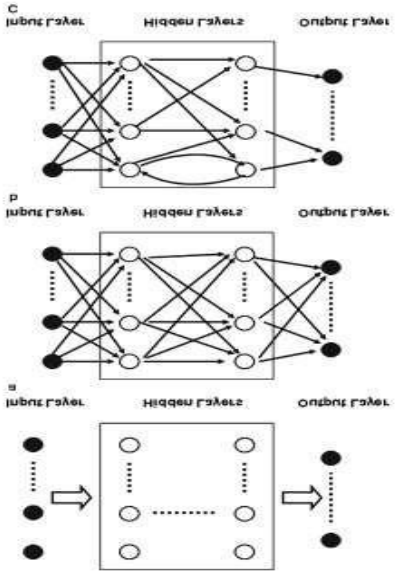


Fig-2.3: (a) A general topology of ANN. (b) Feedforward ANN (perceptron). (c) Feedback ANN.

3.2.5 Hopfield Net

The Hopfield net [10] is a feedback neural network with only a single layer. A threenode Hopfield net is shown in Fig. 2.4. In the Hopfield net, every node is connected with all other nodes (fully connected), but not with itself. Given an

input, the state of the Hopfield net is updated until it converges to an equilibrium state, called the attractor state.

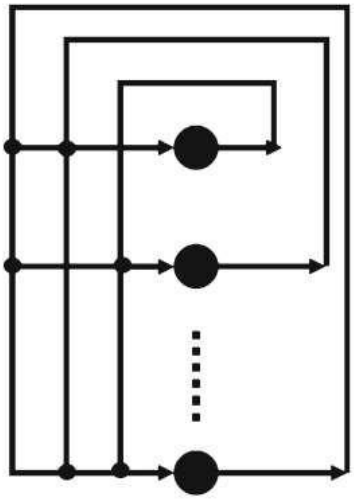


Fig-2.4: Hopfield network.

3.2.6 Kohonen Maps

The Kohonen map [11] is another feedback network. It has a single hidden layer. The nodes in the hidden layer are organized typically as two-dimensional rectangular grids, as shown in Fig. 2.5. Every node in the hidden layer is connected to all the input nodes. The Kohonen map uses unsupervised learning with the winner-takes all learning rule. Only the node with the highest response and its neighboring nodes get their weights updated, so that they are more likely to response to similar input patterns. Kohonen maps can be used for projection of high-dimensional data into two-dimensional space, clustering of data, and the like

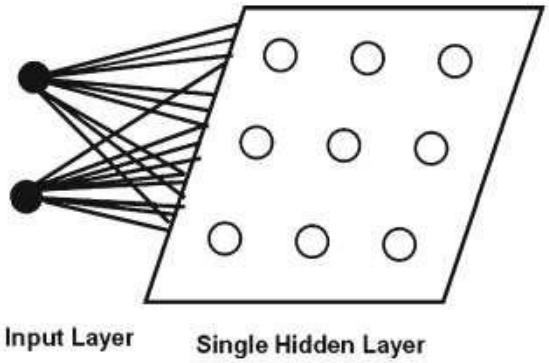


Fig.-2.5: Kohonen maps.

3.3 Summary

This chapter should contain a summary section of 4-5 sentences long.

CHAPTER - 4

DATA COLLECTION

4.1 Introductions

The Accident Research Institute (ARI) affiliated to Bangladesh University of Engineering and Technology (BUET) has obtained data on accidents related to national highways in Bangladesh. The Accident Research Institute (ARI) of Bangladesh University of Engineering and Technology (BUET) is collecting and analyzing data on accidents recorded in Bangladesh. Law enforcement agencies and other agencies collect and document this information. Here is the information we got from the Accident Research Institute (ARI):

Table-1: Number of Accidents on Different National Highways of Bangladesh.

Accident									
Year	N1 (455km)	N2 (286km)	N3 (112km)	N4 (146km)	N5 (507km)	N6 (150km)	N7 (252km)	N8 (191km)	N9 (43.1km)
	Accident	Accident	Accident	Accident	Accident	Accident	Accident	Accident	Accident
1998	150	137	36	82	38	0	16	285	97
1999	233	237	94	121	57	44	104	279	96
2000	192	182	54	189	50	57	93	218	65
2001	128	256	56	133	36	21	82	138	81
2002	262	133	81	195	42	42	131	273	77
2003	126	115	90	138	66	42	123	270	66
2004	411	180	83	116	38	6	54	287	80
2005	343	134	27	83	71	18	113	291	51
2006	288	115	80	72	118	42	97	316	57
2007	258	131	103	63	100	17	90	273	50
2008	229	178	79	91	110	29	79	187	64
2009	124	158	62	46	64	39	70	201	23
2010	106	109	32	39	41	25	49	140	22
2011	38	41	52	39	95	22	32	40	20
2012	75	67	19	34	140	24	31	43	24
2013	112	2	5	2	82	11	24	35	8
Total=	**3075**	**2175**	**953**	**1443**	**1148**	**439**	**1188**	**3276**	**881**

Table-2: Number of Casualties on Different National Highways of Bangladesh.

Casualty									
Year	N1 (455km)	N2 (286km)	N3 (112km)	N4 (146km)	N5 (507km)	N6 (150km)	N7 (252km)	N8 (191km)	N9 (43.1km)
	Casualty	Casualty	Casualty	Casualty	Casualty	Casualty	Casualty	Casualty	Casualty
1998	293	222	79	29	583	161	149	76	0
1999	417	395	152	228	444	187	270	101	94
2000	406	337	116	184	385	102	372	89	149
2001	299	294	117	181	227	130	246	71	43
2002	518	258	166	297	529	146	322	74	98
2003	231	253	178	317	526	137	295	141	152
2004	811	351	152	141	591	155	276	92	11
2005	674	257	74	237	543	39	133	138	27
2006	555	151	130	156	543	109	132	192	85
2007	520	183	144	132	393	79	78	145	28
2008	397	350	165	131	325	104	144	178	76
2009	212	275	119	176	378	40	108	132	56
2010	191	238	69	119	277	35	95	87	54
2011	69	111	79	74	177	43	98	80	49
2012	168	138	22	64	257	44	51	73	50
2013	187	3	8	11	155	31	56	64	14
Total=	**5948**	**3816**	**1770**	**2477**	**6333**	**1542**	**2825**	**1733**	**986**

4.2 Summary

This chapter should contain a summary section of 4-5 sentences long.

CHAPTER – 5

DATA ANALYSIS

5.1 Result: ANN Accidents

RMSE for model with 3 neurons predicting N1(455km): 134.87028266908385

RMSE for model with 4 neurons predicting N1(455km): 132.00374492152656

RMSE for model with 5 neurons predicting N1(455km): 131.77312872459174

RMSE for model with 6 neurons predicting N1(455km): 116.23412623010849

RMSE for model with 7 neurons predicting N1(455km): 138.09045202143813

RMSE for model with 8 neurons predicting N1(455km): 124.50868285632656

RMSE for model with 9 neurons predicting N1(455km): 121.61476437514928

RMSE for model with 10 neurons predicting N1(455km): 123.7179292548985

RMSE for model with 11 neurons predicting N1(455km): 116.61293904810589

RMSE for model with 12 neurons predicting N1(455km): 126.62336530647488

The best model for N1(455km) has 6 neurons with an RMSE of 116.2341

RMSE for model with 3 neurons predicting N2(286km): 127.87062022161555

RMSE for model with 4 neurons predicting N2(286km): 131.7037788200653

RMSE for model with 5 neurons predicting N2(286km): 134.5840681084765

RMSE for model with 6 neurons predicting N2(286km): 125.91852398746327

RMSE for model with 7 neurons predicting N2(286km): 124.59668326736838

RMSE for model with 8 neurons predicting N2(286km): 121.87507056177965

RMSE for model with 9 neurons predicting N2(286km): 118.12278557018129

RMSE for model with 10 neurons predicting N2(286km): 111.5131384671928

RMSE for model with 11 neurons predicting N2(286km): 129.66038110991795

RMSE for model with 12 neurons predicting N2(286km): 120.86701006639089

The best model for N2(286km) has 10 neurons with an RMSE of 111.5131

RMSE for model with 3 neurons predicting N3(112km): 61.7992843050748

RMSE for model with 4 neurons predicting N3(112km): 53.71861768092222

RMSE for model with 5 neurons predicting N3(112km): 62.78900143793937

RMSE for model with 6 neurons predicting N3(112km): 53.917078082754195

RMSE for model with 7 neurons predicting N3(112km): 56.71178500418579

RMSE for model with 8 neurons predicting N3(112km): 50.8186590791579

RMSE for model with 9 neurons predicting N3(112km): 47.97900162478906

RMSE for model with 10 neurons predicting N3(112km): 51.49674752725351

RMSE for model with 11 neurons predicting N3(112km): 54.66531362145639

RMSE for model with 12 neurons predicting N3(112km): 45.553864771002075

The best model for N3(112km) has 12 neurons with an RMSE of 45.5539

RMSE for model with 3 neurons predicting N4(146km): 91.95656398036715

RMSE for model with 4 neurons predicting N4(146km): 85.85531020225395

RMSE for model with 5 neurons predicting N4(146km): 88.18754098697644

RMSE for model with 6 neurons predicting N4(146km): 79.56713540769839

RMSE for model with 7 neurons predicting N4(146km): 80.36689291522157

RMSE for model with 8 neurons predicting N4(146km): 79.74117993148955

RMSE for model with 9 neurons predicting N4(146km): 70.99478600945284

RMSE for model with 10 neurons predicting N4(146km): 71.9922805461892

RMSE for model with 11 neurons predicting N4(146km): 75.28591157319738

RMSE for model with 12 neurons predicting N4(146km): 74.27235992498633

The best model for N4(146km) has 9 neurons with an RMSE of 70.9948

RMSE for model with 3 neurons predicting N5(507km): 74.71405651575722

RMSE for model with 4 neurons predicting N5(507km): 74.7010315599345

RMSE for model with 5 neurons predicting N5(507km): 75.6804169456131

RMSE for model with 6 neurons predicting N5(507km): 70.2862137878711

RMSE for model with 7 neurons predicting N5(507km): 80.49373640453278

RMSE for model with 8 neurons predicting N5(507km): 72.57688255216102

RMSE for model with 9 neurons predicting N5(507km): 61.8623873879065

RMSE for model with 10 neurons predicting N5(507km): 67.86706828811315

RMSE for model with 11 neurons predicting N5(507km): 67.7126936092153

RMSE for model with 12 neurons predicting N5(507km): 58.09805119665961

The best model for N5(507km) has 12 neurons with an RMSE of 58.0981

RMSE for model with 3 neurons predicting N6(150km): 25.693942078253496

RMSE for model with 4 neurons predicting N6(150km): 24.706902463668985

RMSE for model with 5 neurons predicting N6(150km): 26.521981851004007

RMSE for model with 6 neurons predicting N6(150km): 22.842945410945116

RMSE for model with 7 neurons predicting N6(150km): 22.07755714545767

RMSE for model with 8 neurons predicting N6(150km): 21.609605718798644

RMSE for model with 9 neurons predicting N6(150km): 18.267901514564223

RMSE for model with 10 neurons predicting N6(150km): 19.005197685456697

RMSE for model with 11 neurons predicting N6(150km): 17.982250913517728

RMSE for model with 12 neurons predicting N6(150km): 18.40517364938198

The best model for N6(150km) has 11 neurons with an RMSE of 17.9823

RMSE for model with 3 neurons predicting N7(252km): 74.28942512476995

RMSE for model with 4 neurons predicting N7(252km): 67.83316678904342

RMSE for model with 5 neurons predicting N7(252km): 67.73156014710139

RMSE for model with 6 neurons predicting N7(252km): 63.803162468295525

RMSE for model with 7 neurons predicting N7(252km): 61.08579422812982

RMSE for model with 8 neurons predicting N7(252km): 61.186618453298266

RMSE for model with 9 neurons predicting N7(252km): 65.0681051512823

RMSE for model with 10 neurons predicting N7(252km): 57.56471646986728

RMSE for model with 11 neurons predicting N7(252km): 65.81770308774351

RMSE for model with 12 neurons predicting N7(252km): 60.110226941084996

The best model for N7(252km) has 10 neurons with an RMSE of 57.5647

RMSE for model with 3 neurons predicting N8(191km): 204.78648342926502

RMSE for model with 4 neurons predicting N8(191km): 215.01452262124843

RMSE for model with 5 neurons predicting N8(191km): 206.63933203134616

RMSE for model with 6 neurons predicting N8(191km): 207.01075674081937

RMSE for model with 7 neurons predicting N8(191km): 202.59801328803619

RMSE for model with 8 neurons predicting N8(191km): 198.90750702892848

RMSE for model with 9 neurons predicting N8(191km): 204.5358243382547

RMSE for model with 10 neurons predicting N8(191km): 193.99687048835486

RMSE for model with 11 neurons predicting N8(191km): 194.0001226410926

RMSE for model with 12 neurons predicting N8(191km): 210.72447038586998

The best model for N8(191km) has 10 neurons with an RMSE of 193.9969

RMSE for model with 3 neurons predicting N9(43.1km): 56.68727027557357

RMSE for model with 4 neurons predicting N9(43.1km): 61.43070606352397

RMSE for model with 5 neurons predicting N9(43.1km): 62.00064579024541

RMSE for model with 6 neurons predicting N9(43.1km): 61.269014157485934

RMSE for model with 7 neurons predicting N9(43.1km): 58.56467558250274

RMSE for model with 8 neurons predicting N9(43.1km): 48.682412483924864

RMSE for model with 9 neurons predicting N9(43.1km): 52.07096685803889

RMSE for model with 10 neurons predicting N9(43.1km): 54.79234174297879

RMSE for model with 11 neurons predicting N9(43.1km): 55.63077916643792

RMSE for model with 12 neurons predicting N9(43.1km): 50.757007943429166

The best model for N9(43.1km) has 8 neurons with an RMSE of 48.6824

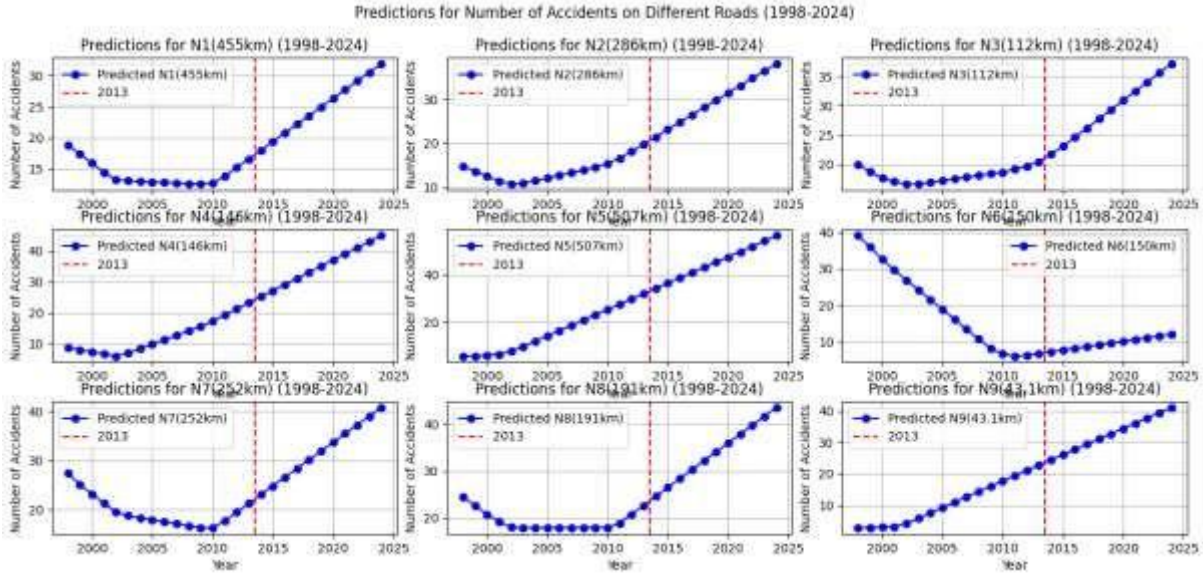


Figure 5.1: N1 to N9 Accidents predictions from 2014 to 2024

5.2 Result: ANN Casualty

RMSE for model with 3 neurons predicting N1(455km): 249.9412605992405

RMSE for model with 4 neurons predicting N1(455km): 255.07367121422115

RMSE for model with 5 neurons predicting N1(455km): 255.5191309837858

RMSE for model with 6 neurons predicting N1(455km): 248.23792419741818

RMSE for model with 7 neurons predicting N1(455km): 243.36373231496782

RMSE for model with 8 neurons predicting N1(455km): 258.6518615974653

RMSE for model with 9 neurons predicting N1(455km): 238.98705071112002

RMSE for model with 10 neurons predicting N1(455km): 249.4255700658633

RMSE for model with 11 neurons predicting N1(455km): 257.8147690613783

RMSE for model with 12 neurons predicting N1(455km): 235.17434244758914

The best model for N1(455km) has 12 neurons with an RMSE of 235.1743

RMSE for model with 3 neurons predicting N2(286km): 238.46392383498545

RMSE for model with 4 neurons predicting N2(286km): 244.3616826580616

RMSE for model with 5 neurons predicting N2(286km): 236.2627057320624

RMSE for model with 6 neurons predicting N2(286km): 232.24810174537453

RMSE for model with 7 neurons predicting N2(286km): 229.91690604239795

RMSE for model with 8 neurons predicting N2(286km): 233.22588029896167

RMSE for model with 9 neurons predicting N2(286km): 232.76474010368932

RMSE for model with 10 neurons predicting N2(286km): 232.37874478248406

RMSE for model with 11 neurons predicting N2(286km): 226.9638578816891

RMSE for model with 12 neurons predicting N2(286km): 232.09626022300645

The best model for N2(286km) has 11 neurons with an RMSE of 226.9639

RMSE for model with 3 neurons predicting N3(112km): 113.9485637239858

RMSE for model with 4 neurons predicting N3(112km): 110.47731420232661

RMSE for model with 5 neurons predicting N3(112km): 106.72381990644737

RMSE for model with 6 neurons predicting N3(112km): 106.6203921376784

RMSE for model with 7 neurons predicting N3(112km): 105.82419656460621

RMSE for model with 8 neurons predicting N3(112km): 100.7492607768658

RMSE for model with 9 neurons predicting N3(112km): 108.42988252504514

RMSE for model with 10 neurons predicting N3(112km): 102.62007859448681

RMSE for model with 11 neurons predicting N3(112km): 97.68819961980972

RMSE for model with 12 neurons predicting N3(112km): 98.12637969151697

The best model for N3(112km) has 11 neurons with an RMSE of 97.6882

RMSE for model with 3 neurons predicting N4(146km): 176.11421643618624

RMSE for model with 4 neurons predicting N4(146km): 174.96392336744006

RMSE for model with 5 neurons predicting N4(146km): 169.05849933192874

RMSE for model with 6 neurons predicting N4(146km): 164.8697825502624

RMSE for model with 7 neurons predicting N4(146km): 171.20694687036152

RMSE for model with 8 neurons predicting N4(146km): 169.8890980812577

RMSE for model with 9 neurons predicting N4(146km): 167.81914809466826

RMSE for model with 10 neurons predicting N4(146km): 167.1213004428101

RMSE for model with 11 neurons predicting N4(146km): 160.66684281550417

RMSE for model with 12 neurons predicting N4(146km): 163.73101891224337

The best model for N4(146km) has 11 neurons with an RMSE of 160.6668

RMSE for model with 3 neurons predicting N5(507km): 419.46852086099904

RMSE for model with 4 neurons predicting N5(507km): 419.43695703192583

RMSE for model with 5 neurons predicting N5(507km): 409.25748290075677

RMSE for model with 6 neurons predicting N5(507km): 416.75592261527197

RMSE for model with 7 neurons predicting N5(507km): 403.2427839462204

RMSE for model with 8 neurons predicting N5(507km): 410.3163276232061

RMSE for model with 9 neurons predicting N5(507km): 409.8363921936662

RMSE for model with 10 neurons predicting N5(507km): 414.6984236589113

RMSE for model with 11 neurons predicting N5(507km): 412.20888328156735

RMSE for model with 12 neurons predicting N5(507km): 396.73141904239884

The best model for N5(507km) has 12 neurons with an RMSE of 396.7314

RMSE for model with 3 neurons predicting N6(150km): 128.31698119963468

RMSE for model with 4 neurons predicting N6(150km): 118.22701273189436

RMSE for model with 5 neurons predicting N6(150km): 118.24804998599345

RMSE for model with 6 neurons predicting N6(150km): 124.25897193119162

RMSE for model with 7 neurons predicting N6(150km): 110.62492592229508

RMSE for model with 8 neurons predicting N6(150km): 114.65844449483552

RMSE for model with 9 neurons predicting N6(150km): 109.38086263062343

RMSE for model with 10 neurons predicting N6(150km): 115.81048293014447

RMSE for model with 11 neurons predicting N6(150km): 114.0772242738045

RMSE for model with 12 neurons predicting N6(150km): 106.85332790787398

The best model for N6(150km) has 12 neurons with an RMSE of 106.8533

RMSE for model with 3 neurons predicting N7(252km): 190.7911619884531

RMSE for model with 4 neurons predicting N7(252km): 191.01228132256037

RMSE for model with 5 neurons predicting N7(252km): 183.504946097274

RMSE for model with 6 neurons predicting N7(252km): 185.74552434843397

RMSE for model with 7 neurons predicting N7(252km): 189.05040381283249

RMSE for model with 8 neurons predicting N7(252km): 185.7919916788746

RMSE for model with 9 neurons predicting N7(252km): 178.98836048423436

RMSE for model with 10 neurons predicting N7(252km): 173.72005948307392

RMSE for model with 11 neurons predicting N7(252km): 170.8972919767058

RMSE for model with 12 neurons predicting N7(252km): 180.540846364514

The best model for N7(252km) has 11 neurons with an RMSE of 170.8973

RMSE for model with 3 neurons predicting N8(191km): 94.23244220593186

RMSE for model with 4 neurons predicting N8(191km): 90.16584601971357

RMSE for model with 5 neurons predicting N8(191km): 92.84718537944657

RMSE for model with 6 neurons predicting N8(191km): 86.2056820687907

RMSE for model with 7 neurons predicting N8(191km): 82.85185138121867

RMSE for model with 8 neurons predicting N8(191km): 89.48143574977823

RMSE for model with 9 neurons predicting N8(191km): 84.06602664381857

RMSE for model with 10 neurons predicting N8(191km): 83.55481310277072

RMSE for model with 11 neurons predicting N8(191km): 85.03686029507463

RMSE for model with 12 neurons predicting N8(191km): 81.83727958892592

The best model for N8(191km) has 12 neurons with an RMSE of 81.8373

RMSE for model with 3 neurons predicting N9(43.1km): 82.83410029268927

RMSE for model with 4 neurons predicting N9(43.1km): 74.72674618698323

RMSE for model with 5 neurons predicting N9(43.1km): 78.29886278284232

RMSE for model with 6 neurons predicting N9(43.1km): 76.10596512886896

RMSE for model with 7 neurons predicting N9(43.1km): 78.27849963046947

RMSE for model with 8 neurons predicting N9(43.1km): 75.51619103261105

RMSE for model with 9 neurons predicting N9(43.1km): 77.02836279353261

RMSE for model with 10 neurons predicting N9(43.1km): 77.21602700093935

RMSE for model with 11 neurons predicting N9(43.1km): 71.73595245374415

RMSE for model with 12 neurons predicting N9(43.1km): 67.83119322177592

The best model for N9(43.1km) has 12 neurons with an RMSE of 67.8312

Casualty

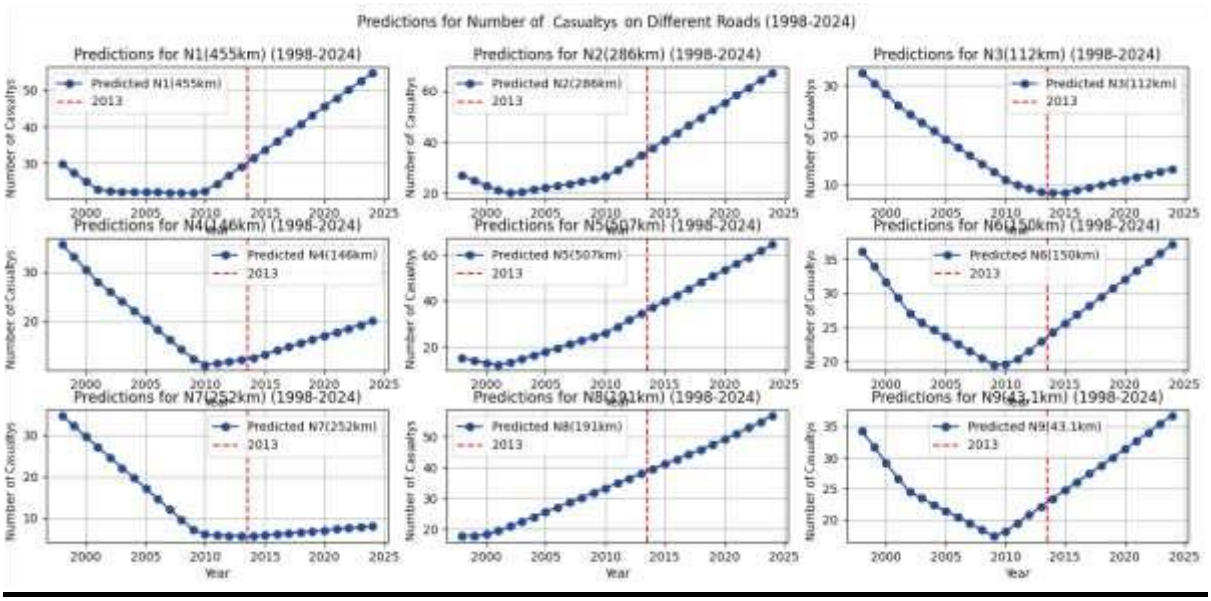


Figure 5.2: N1 to N9 Casualty predictions from 2014 to 2024.

CHAPTER – 6

CONCLUSIONS AND FUTURE WORKS

6.1 Conclusions

1. Model Performance: The artificial neural network (ANN) demonstrated promising performance in predicting fatality rates of road accidents in Bangladesh, showcasing its potential as a valuable tool for road safety analysis.
2. Key Findings: Our study identified several significant factors contributing to road accident fatalities, as highlighted by the ANN model. These insights provide valuable information for policymakers and stakeholders to prioritize interventions effectively.
3. Implications: The findings underscore the importance of targeted interventions and policy initiatives to mitigate road accident fatalities in Bangladesh. Implementing evidence-based strategies informed by our study can lead to substantial improvements in road safety outcomes.
4. Limitations: Despite its strengths, our study is subject to certain limitations, including data availability constraints and the inherent complexity of modeling road accident dynamics. Addressing these limitations in future research will enhance the robustness and generalizability of our findings.

6.2 Future Works:

1. Enhanced Data Collection: Future research should focus on enhancing the quality and granularity of accident data, incorporating additional variables such as weather conditions, road infrastructure, and vehicle characteristics to improve predictive accuracy.
2. Model Refinement: Further refinement of the ANN model is warranted, exploring alternative architectures, hyper parameters, and optimization techniques to enhance its predictive capabilities and reliability.
3. Exploratory Analysis: Investigating the causal relationships between various factors and road accident fatalities through advanced statistical techniques will deepen our understanding of the underlying mechanisms and inform targeted interventions.
4. Geospatial Analysis: Integrating geospatial data and geographic information systems (GIS) into the analysis can facilitate spatial visualization and hotspot identification, enabling policymakers to prioritize resources and interventions in high-risk areas.

5. Policy Evaluation: Assessing the effectiveness of different policy interventions and road safety initiatives through scenario analysis and simulation modeling will provide valuable insights into their potential impact and guide evidence-based decision-making.

In summary, our study lays the groundwork for future research endeavors aimed at improving road safety outcomes in Bangladesh through data-driven approaches and policy interventions informed by predictive modeling and analysis.

REFERENCES

S. O. Report, “Road crashes kill 7,855 people in 2019,” The Daily Star, 11-Jan2020. [Online]. Available: https:// www.thedailystar.net/ country/ road-accidents kill-7855people-in-2019 -in-bangladesh -1852684. [Accessed: 24-Feb-2020].

[1] “Bangladesh:Alarming rise in road crashes,”Anadolu Ajansı.[Online].Available: https://www.aa.com.tr/en/asia-pacific/ bangladesh-alarmingrise- in-road-crashes/ [Accessed: 24-Feb-2020].

[2] M. M. L. Elahi, R. Yasir, M. A. Syrus, M. S. Q. Z. Nine, I. Hossain and N. Ahmed, “Computer vision based road traffic accident and anomaly detection in the context of Bangladesh,” 2014 International Conference on Informatics, Electronics Vision (ICIEV), Dhaka, 2014, pp. 1-6, doi: 10.1109/ICIEV.2014.6850780.

[3] Z. Zhang, “Introduction to machine learning: k-nearest neighbors,” Annals of Translational Medicine (2016), vol 4, issue no. 11, pp.218- 218.

[4] I. Rish, "An Empirical Study of the Naïve Bayes Classifier," IJCAI 2001 Work Empir Methods Artif Intell (2001), vol 3.

[5] R. Wang, “AdaBoost for Feature Selection, Classification and Its Relation with SVM”, A Review. Physics Procedia (2012), vol. 25, pp. 800-807.

[6] McCulloch WS, Pitts W (1943) A logical calculus of ideas immanent in nervous activity, Bull Mathematical Biophysics 5:115–133.

[7] Minsky ML, Papert SA (1969) Perceptrons. MIT Press, Cambridge, MA.

[8] Chiu T-L, So S-S (2003) QSAR Comb Sci 22:519–526.

[9] Hopfield JJ (1982) Neural networks and physical systems with emergent collective computa-tional abilities. Proc Nat Acad Sci 79:2554–2558.

[10] Kohonen T (1989) Self Organization and associative memory, 3 edn. SpringerVerlag, New York.

APPENDIX

Appendix A.1: Code for predicting number of accidents from N1 to N9 import os os.environ['TF_ENABLE_ONEDNN_OPTS'] = '0'

 # Step 1: Data Preprocessing import pandas as pd import numpy as np from sklearn.model_selection import train_test_split from sklearn.preprocessing import StandardScaler from sklearn.metrics import mean_squared_error import matplotlib.pyplot as plt import tensorflow as tf

 # Load the dataset from Excel starting from the 3rd row and ending at the 18th row data = pd.read_excel('Accident Data.xlsx', skiprows=1, nrows=16)

 # Extracting input features and target variables

X = data.iloc[:, 0].values.reshape(-1, 1) # Extract the first column as features (years) outputs = data.iloc[:, 1:] # Extract output columns from the second column (N1 to N9)

 # Scale the features scaler = StandardScaler()

X_scaled = scaler.fit_transform(X)

 # Function to create and train the ANN model def create_and_train_model(X_train, y_train, hidden_neurons):

 # Define the architecture of the ANN model = tf.keras.Sequential([tf.keras.layers.Dense(hidden_neurons, activation='relu'), tf.keras.layers.Dense(1)

 # One output neuron for the number of accidents])

 # Compile the model model.compile(optimizer='adam', loss='mse')

 # Mean Squared Error loss for regression

 # Train the model

model.fit(X_train, y_train, epochs=1000, batch_size=32, validation_split=0.2, verbose=0) return model

 # List of neurons to test neuron_counts = list(range(3, 13)) # 3 to 12 neurons

 # Train and evaluate models for each target variable all_models = {} for i, target in enumerate(outputs.columns): y = outputs.iloc[:, i].values.reshape(-1, 1)

 # Extract the target column

 # Split the target data into training and testing sets

X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.3, random_state=42)

 # Dictionary to store RMSE values for each model rmse_dict = {} for neurons in neuron_counts:

 # Create and train the model model = create_and_train_model(X_train, y_train, neurons)

 # Predict on the test set y_test_pred = model.predict(X_test)

 # Calculate RMSE for the model rmse = np.sqrt(mean_squared_error(y_test, y_test_pred)) rmse_dict[neurons] = rmse print(f"RMSE for model with {neurons} neurons predicting {target}: {rmse}")

 # Determine the best model based on RMSE best_neurons = min(rmse_dict, key=rmse_dict.get) best_model = create_and_train_model(X_train, y_train, best_neurons) all_models[target] = best_model print(f"The best model for {target} has {best_neurons} neurons with an RMSE of {rmse_dict[best_neurons]:.4f}")

 # Make Predictions for All Years and plot them all_years = np.array([year for year in range(1998, 2025)]).reshape(-1, 1) all_years_scaled = scaler.transform(all_years) fig, axs = plt.subplots(3, 3, figsize=(18, 12)) fig.suptitle('Predictions for Number of Accidents on Different Roads (1998-2024)') for i, target in enumerate(outputs.columns): model = all_models[target] all_predictions = model.predict(all_years_scaled) row, col = divmod(i, 3) axs[row, col].plot(all_years, all_predictions, color='blue', marker='o', label=f'Predicted {target}') axs[row, col].axvline(x=2013.5, color='red', linestyle='--', label='2013')

 # Marking the separation between historical and future axs[row, col].set_title(f'Predictions for {target} (1998-2024)') axs[row, col].set_xlabel('Year') axs[row, col].set_ylabel('Number of Accidents') axs[row, col].legend() axs[row, col].grid(True) plt.tight_layout(rect=[0, 0.03, 1, 0.95]) plt.show()

Appendix A.2: Code for predicting number of Casualty from N1 to N9

import os os.environ['TF_ENABLE_ONEDNN_OPTS'] = '0'

 # Step 1: Data Preprocessing import pandas as pd import numpy as np from sklearn.model_selection import train_test_split from sklearn.preprocessing import StandardScaler from sklearn.metrics import mean_squared_error import matplotlib.pyplot as plt import tensorflow as tf

 # Load the dataset from Excel starting from the 3rd row and ending at the 18th row data = pd.read_excel('Causalities Data.xlsx', skiprows=2, nrows=16)

 # Extracting features and target variables X = data.iloc[:, 0].values.reshape(-1, 1) # Extract the first column as features (years) y = data.iloc[:, 1].values.reshape(-1, 1) # Extract the first target column (N1) as the number of accidents

 # Split the data into training (70%) and testing (30%) sets

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

 # Scale the features scaler=StandardScaler()

X_train_scaled=scaler.fit_transform(X_train)

X_test_scaled = scaler.transform(X_test)

 # Function to create and train the ANN model def

create_and_train_model(hidden_neurons):

 # Define the architecture of the ANN model = tf.keras.Sequential([

tf.keras.layers.Dense(hidden_neurons, activation='relu',

input_shape=(X_train_scaled.shape[1],)), tf.keras.layers.Dense(1)

 # One output neuron for the number of accidents])

Compile the model.compile(optimizer='adam', loss='mse') # Mean Squared Error loss for regression

 # Train the model model.fit(X_train_scaled, y_train, epochs=1000, batch_size=32,

validation_split=0.2, verbose=0) return model

 # List of neurons to test neuron_counts = list(range(3, 13)) # 3 to 12 neurons

Dictionary to store RMSE values for each model rmse_dict = {} import os

os.environ['TF_ENABLE_ONEDNN_OPTS'] = '0'

 # Step 1: Data Preprocessing import pandas as pd import numpy as np from

sklearn.model_selection import train_test_split from sklearn.preprocessing import StandardScaler from sklearn.metrics import mean_squared_error import matplotlib.pyplot as plt import tensorflow as tf

 # Load the dataset from Excel starting from the 3rd row and ending at the 18th row

data = pd.read_excel('Causalities Data.xlsx', skiprows=1, nrows=16)

 # Extracting input features and target variables

X = data.iloc[:, 0].values.reshape(-1, 1) # Extract the first column as features (years) outputs = data.iloc[:, 1:] # Extract output columns from the second column (N1 to N9)

 # Scale the features scaler = StandardScaler() X_scaled = scaler.fit_transform(X)

 # Function to create and train the ANN model def create_and_train_model(X_train, y_train, hidden_neurons):

 # Define the architecture of the ANN model = tf.keras.Sequential([

tf.keras.layers.Dense(hidden_neurons, activation='relu'), tf.keras.layers.Dense(1) # One output neuron for the number of accidents])

 # Compile the model model.compile(optimizer='adam', loss='mse') # Mean Squared Error loss for regression

 # Train the model model.fit(X_train, y_train, epochs=1000, batch_size=32,
validation_split=0.2,
verbose=0) return model
 # List of neurons to test neuron_counts = list(range(3, 13))
 # 3 to 12 neurons
 # Train and evaluate models for each target variable all_models = {} for i, target in
enumerate(outputs.columns): y = outputs.iloc[:, i].values.reshape(-1, 1)
 # Extract the target column
 # Split the target data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.3, random_state=42)
 # Dictionary to store RMSE values for each model rmse_dict = {} for neurons in
neuron_counts:
 # Create and train the model
model = create_and_train_model(X_train, y_train, neurons)
 # Predict on the test set y_test_pred = model.predict(X_test)
 # Calculate RMSE for the model rmse = np.sqrt(mean_squared_error(y_test,
y_test_pred)) rmse_dict[neurons] = rmse print(f"RMSE for model with {neurons} neurons predicting {target}: {rmse}")
 # Determine the best model based on RMSE best_neurons = min(rmse_dict,
key=rmse_dict.get) best_model = create_and_train_model(X_train, y_train, best_neurons) all_models[target] = best_model print(f"The best model for {target} has {best_neurons} neurons with an RMSE of {rmse_dict[best_neurons]:.4f}")
 # Make Predictions for All Years and plot them all_years = np.array([year for year in
range(1998, 2025)]).reshape(-1, 1) all_years_scaled = scaler.transform(all_years) fig, axs = plt.subplots(3, 3, figsize=(18, 12)) fig.suptitle('Predictions for Number of Causalities on Different Roads (1998-2024)') for i, target in enumerate(outputs.columns): model = all_models[target] all_predictions = model.predict(all_years_scaled) row, col = divmod(i, 3) axs[row, col].plot(all_years, all_predictions, color='blue', marker='o', label=f'Predicted {target}') axs[row, col].axvline(x=2013.5, color='red', linestyle='--', label='2013') # Marking the separation between historical and futureaxs[row,col].set_title(f'Predictions for {target} (1998-2024)' axs[row, col].set_xlabel('Year') axs[row, col].set_ylabel('Number of Accidents') axs[row,col].legend()axs[row,col].grid(True)plt.tight_layout(rect=[0,0.03,1,0.95]) plt.show()