An-Najah National University
Faculty of Graduate Studies

RISK PREDICTION OF TRAFFIC ACCIDENT USING MACHINE LEARNING

By
Amani Mohammed Hakawati

Supervisor
Dr. Adnan Salman

This Thesis is Submitted in Partial Fulfillment of the Requirements for the Degree of Master of Advanced Computing, in the Faculty of Graduate Studies, An-Najah National University, Nablus, Palestine.
2022

RISK PREDICTION OF TRAFFIC ACCIDENT USING MACHINE LEARNING

By
Amani Mohammed Hakawati

This Thesis was Defended Successfully on 21/07/2022 and approved by

Dr. Adnan Salman **Supervisor**	Signature
Dr. Imad Saada **External Examiner**	Signature
Dr. Samir Matar **Internal Examiner**	Signature

Dedication

Special thanks must go to my mother for without here I could do nothing

My family,

my friends,

and my partner Wajdi .

Thanks for all provided unconditional support and

encouragement through both the highs and lows of my time in graduate school.

Acknowledgment

First and foremost, I would like to acknowledge the tireless and prompt help of my supervisor, Dr.Adnan Salman. He always been here, he allowed me complete freedom to define and explore my own directions in research. And he gave me constant support and guidance.

I would like to express my sincere gratitude to staff of the Departments of Computer Sciences for their support.

To the committee for taking the time to review my work and give me their much-appreciated notes and remarks.

I would especially thank Dr. Baker Abdulhaq, Dr. Ahmad Awaad, Dr. Amjad Hawwash, and Mr. Mohammed Adas for their encouragement and helpful suggestions.

Declaration

I, the undersigned, declare that I submitted the thesis entitled:

RISK PREDICTION OF TRAFFIC ACCIDENT USING MACHINE LEARNING

I declare that the work provided in this thesis, unless otherwise referenced, is the researcher's own work, and has not been submitted elsewhere for any other degree or qualification.

Student's Name: أماني محمد حجوي

Signature: أماني

Date: ٢٠٢٣/٧/٢١

Table of Contents

List of Tables

List of Figures

List of Appendices

RISK PREDICTION OF TRAFFIC ACCIDENT USING MACHINE LEARNING

By
Amani Mohammed Hakawati
Supervisor
Dr. Adnan Salman

Abstract

Introduction: Traffic accidents imply congestion, delays, economic losses, disability persons and sometimes loss of human life. There are many factors influencing the likelihood of occurrence and severity. These factors include driver-related issues, topography and road-related issues, weather-related issues, and other accident-related issues. Predicting the severity of road accidents and understanding the factors that cause them are interesting research goals in traffic safety. This thesis analyses many traffic accidents deeply and determines the severity of accidents by using machine learning techniques. We also identified the important factors that have a direct impact on the severity of a traffic accident. This knowledge can help trainers to better educate new drivers to avoid traffic accident and can help policy maker in enforcing new laws that help reducing the number of severity of these accidents.

Methodology: Analysis has been done using Support Vector Machine (SVM), Artificial Neural Networks (ANN), and Random Forest (RF). Cross-validation with 10-fold was used to evaluate the performance. Before applying machine learning techniques, traffic accidents data was preprocessed through three stages: handling missing data, handling text and categorical attributes, and feature scaling.

Results: In this thesis, we had classified the severity of an accident into four classes based on the time delay after the accident. Our results, indicates that the most important factors that have a direct impact on the severity were time duration, end latitude, end longitude, start longitude, start latitude, and distance(mi).

Conclusion: Considering the overall accuracy, RF classifier was outperformed with (93.45%), followed by ANN (90.18%), and SVM (89.74%).

Keywords: Road accident, Support Vector Machine (SVM), Artificial Neural Networks (ANN), Random Forest(RF)

Chapter One
Introduction

Every morning and evening, we are surprised by traffic accidents, which cause us to mourn for our loved ones. We cannot accept it or become acquainted with it as humans. When the issue affects people's lives, it becomes vital to pay more attention to it and take actions.

Due to the numerous deaths, serious injuries, and economic losses that they inflict on a daily basis, traffic accidents are one of the most pressing concerns that require additional effort. According to new WHO research, road traffic deaths are on the rise, with 1.35 million people dying each year.[1]. The WHO Global status report on road safety in December 2018 highlights that road traffic injuries are now the leading killer of children and young people aged 5-29 years.

In Palestine, annual reports issued by the Palestinian Central Bureau of Statistics and the Ministry of Transport and Communication and Police indicate that number of accidents on the roads, the number of injured, and the number of fatalities is increasing from year to year [2].In addition to fatalities, road accidents cause traffic congestion, delays, and resource depletion. One thing worth noting is that the traffic accident report is regarded as the primary source of data for judicial, legal, and engineering purposes. The first step in creating an integrated database to offer relevant data to interested parties is to develop a system for recording traffic accidents [3].

The Palestinian Statistics Center publishes annual statistics on traffic accidents [4], Table 1.1 in appendix (A) displays data on fatalities in traffic accidents in the West Bank by month and governorate in the 2020 report. It is obvious that the situation is quite concerning. Accidents that result in death or major harm are not "accidents" and can be avoided. Palestine, like other countries, is committed to reducing the number of fatal and serious injury crashes to zero, and this is the most essential goal of this research. It should be noted that the numbers in this table do not include information on of Jerusalem.

One of the most significant challenges we encountered was the lack of a computerized data set regarding accidents in Palestine. To solve this issue, we obtained over 600 insurance files from a single insurance firm and extracted critical information about these accidents. These files solely contain information on drivers and do not contain information about passengers. Data from traffic accidents occurring from 2013 through 2018 was used in the initial study, as well as other accident-related information such as time, place, and date. Then, using Excel, we inserted 523 records into the computer. There are 16 attributes in each accident record. Because these data are small, we looked into other large data sets available online, one of which has 1048575 accident data from the United States [5].

Most studies reveal that road accidents are complicated and one-of-a-kind events[6][7], with various elements influencing the likelihood of occurrence and severity. These factors include driver-related issues, topography and road-related issues, weather-related issues, and other accident-related issues. Additionally, the relationship between accident frequency and road characteristics is complex. As a result, properly anticipating traffic accidents is a difficult problem. Therefore, we considered anticipating the severity of traffic accidents instead. The severity of an accident is represented by a number between 1 and 4, with 1 indicating the least impact on traffic (i.e., shortest delay).

Machine learning prediction models can be used to forecast complicated and uncertain occurrences such as car accidents [8]. As is well known, a number of factors influence a Machine Learning method's success on a given task. First and foremost, the source and quality of the instance data must be considered. Knowledge discovery becomes more difficult during the training phase when there is a lot of redundant and irrelevant information or noisy and erroneous data. Data representation is also very significant. [9]. It is commonly known that data preparation and filtering stages in ML issues consume a significant amount of processing time. Data cleaning, normalization, transformation, feature extraction and selection, and so on are all examples of data pre-processing [9].

We looked at three well-known classification algorithms in this thesis: Random Forest[11], Support Vector Machine[10], and Deep Neural Network[12]. Accurate accident prediction models are crucial in improving road safety. These characteristics, as well as trends in road accidents, can be used to develop accurate traffic safety rules.

1.1 Thesis objectives

1. **General Objective:**

Based on accident records that have occurred in the United States, the goal of this research is to design and implement a system that can predict the severity of injury caused by traffic accidents.

2. **Specific objectives:**

- Identification of the main factors that have a clear effect on accident severity.
- Identification of driver risky behavior and prediction of crash risk, which can be used to educate drivers and helps policy making.
- Prediction of the accident severity of a possible traffic accident.
- Evaluate three type of popular classification methods and compared between them.

Chapter Two
Related Works

Since the late twentieth century, the world has seen tremendous population growth, in addition, there have been more vehicles on the road and more transportation facilities. Traffic accidents have been increasing at a rapid rate. Numerous researchers have conducted studies on road accidents.

Marcin, & Ajith [13] summarize the performance of four machine learning algorithms using data from traffic accident injuries. This study analyzed data from the United States' National Automotive Sampling System (NASS) [14] and General Estimates System (GES). Their objective was to create intelligent models based on machine learning that could accurately classify the severity of injuries (5 categories). They investigated support vector machines, neural networks trained using hybrid learning techniques, decision trees, and a hybrid model combining decision trees and neural networks concurrently. The classification accuracy obtained in studies indicates that the hybrid technique outperformed neural networks, decision trees, and support vector machines for Classes of fatal injuries, incapacitating injuries, and non-incapacitating injuries. For the possible injury and no injury classes, the hybrid strategy outperformed the neural network. The classes of possible injury and no injury are best depicted directly using decision trees.

The authors in [15] demonstrated the application of the adaptive Neuro-Fuzzy Inference System (ANFIS) technique to estimate road accident frequency as a function of road geometric and environmental variables. The study area encompassed a 65-kilometer stretch of the road that was separated into fixed pieces of one kilometer in length.

The proposed ANFIS model was also compared to three parametric models, including Poisson, negative binomial, and nonlinear regression models. For this study, three performance criteria were considered: Root Mean Square Error (RMSE), Mean Relative Error (MRE) and Variance Account For (VAF). On the basis of these comparative criteria, the ANFIS model outperformed the three statistical models Moreover, we

selected variables such as shoulder width (SW), road width (RW), land use (LU), and access points (AP) that had a significant impact on the frequency of road accidents. For determining which subset is the best, three criteria are used: Bayesian Information Criterion (BIC), Mallows' Critical Point (Cp), and Akaike Information Criterion (AIC).

The authors of [16] proposed a novel variable selection method based on Frequent Pattern trees (FP trees). In this method, every frequent pattern from the traffic accident data set is selected. Then, for each frequent pattern, introduce a new metric called the Relative Object Purity Ratio (ROPR). ROPR is then used to rank and select the variables whose significance contributes most to explaining accident patterns by determining the degree of significance of each explanatory variable. Using accident data collected on Interstate I-64 in Virginia, the study develops two traffic accident risk prediction models, a Bayesian network and a k-nearest neighbor model, to demonstrate the benefits of the variable selection method proposed in this study. Two methods of variable selection are utilized prior to model development: (1) the proposed FP tree-based variable selection method; and (2) the random forest method, which is an extensively used method that serves as a baseline for comparison. As a result of the study, FP tree-based accident risk prediction models outperformed random forest-based models, regardless of the type of datasets used, the type of prediction model (i.e., Bayesian network or k-nearest neighbor), or the parameters used. Based on FP trees, the best model found predicts 61.11 percent of accidents while having a false alarm rate of 38.16 percent.

In 2015, Ling et al. [17] Based on weather data and Microwave Vehicle Detection System (MVDS) data, a Bayesian logistic regression method is proposed for predicting expressway weaving segment crashes. There is a significant relationship between the speed difference between the start and end of weaving segments, as well as the mainline speed at the start of weaving segments, and the subsequent crash risk for weaving segments during the following five to ten minutes. Additionally, the configuration is

critical. The weaving segment, in which on- and off-ramp traffic does not need to change lanes, as a result of the increased interactions between weaving traffic and non-weaving traffic, this route has a high crash risk. On the other hand, the maximum length, which measures the distance beyond which weaving turbulence has no impact, is significantly associated with crash risk at a 95 percent level of confidence. Along with geometric and traffic factors, wet pavement surface conditions play a significant role in increasing the crash ratio by 77 percent.

In 2016, Madhar et al. [18] developed machine learning models (classifiers) capable of accurately predicting the injury severity of any new accident. The study analyzed 5973 traffic accident records from 2008 to 2013 in Abu Dhabi. Using the data mining software Waikato Environment for Knowledge Analysis (WEKA). In addition, the research aimed to develop a set of rules that would assist United Arab Emirates (UAE) Traffic Agencies to identify the primary factors that have an impact on accident severity. To model the severity of injury. The four most popular classification algorithms were used. These included the Decision Tree (DT) (J48), Rule Induction (PART), Naive Bayes (NB), and Multilayer Perceptron algorithms (MLP). The findings indicate that the most significant predictors of fatality severity were gender, age, accident year, nationality, collision type, and casualty status. All of the (PART classifiers, DT J48 classifiers, and MLP classifiers performed similarly well in predicting the severity of traffic accidents injuries. For NB, the accuracy decreased.

The authors of [19] proposed a new model for real-time crash risk prediction based on an adaptive neural network fuzzy inference system (ANFIS) and decision tree method. They compared the result to several other methods for predicting crash risk in real time, including supported vector machine (SVM), decision tree, and logistic regression. Traffic data between 0 and 30 minutes before the crash was extracted at intervals of five minutes each. A 5-minute interval of data was collected to obtain the appropriate training period using traffic detection devices such as a video detection system, a microwave sensor, and a loop detector. There is a potential for a 65% accuracy rate for

the prediction of crash occurrences, with a 7.5 percent false alarm rate.

Zhenhua et al. [20] used a deep learning model to detect traffic accidents using social media data in 2017. The purpose of this study is to detect traffic accidents in real time, which could result in improved emergency response. To begin, they conduct a thorough investigation of the over 3 million tweets sent over the course of a year in two metropolitan areas: New York City and Northern Virginia. Second, two deep learning methods are investigated and implemented on the extracted token: Long Short-Term Memory (LSTM) and Deep Belief Network (DBN). DBN classification results outperform those obtained using Support Vector Machines (SVMs) and supervised Latent Dirichlet Allocation (sLDA), with DBN achieving an overall accuracy of 85 percent. The study compared tweets about accident-related incidents with accident records from freeways and traffic data from 15,000 loop detectors installed on local roads to validate its findings. The comparison highlights several critical issues with using Twitter to detect traffic accidents, including location and time bias.

Maher et al. [21] used a Recurrent Neural Network (RNN) to develop a deep learning model for predicting the severity of traffic accidents. During the six-year period between 2009 and 2015, they analyzed 1130 accidents on the North-South Expressway (NSE) in Malaysia. When dealing with sequential data, RNN is more effective than conventional Neural Networks (NNs). Additionally, to illustrate the proposed RNN model's strengths and weaknesses, it was compared to Bayesian Logistic Regression (BLR) and Multilayer Perceptron (MLP) models. Comparative analyses revealed that the RNN model outperformed both the BLR and MLP models. The RNN model achieved a validation accuracy of 71.77 percent, while MLP achieved 65.48 percent of the test, and BLR achieved 58.30 percent.

They predict whether or not an accident will occur on each road segment during each hour in [22]. That is, they approach the problem as a binary one. In the state of Iowa, large data sets containing all motor vehicle crashes from 2006 to 2013, a detailed road network, and various weather attributes with a one-hour granularity have been collected

and map-matched. They compare Random Forest, Decision Tree, Support Vector Machine (SVM), and Deep Neural Network (DNN) classification models. The results indicate that the overall performance of RF and DNN is comparable to and significantly superior to that of SVM and DT. With a 0.9612 Area Under Curve and a 0.9511 accuracy value, DNN scores the highest.

By analyzing the spatial and temporal patterns of traffic accident frequency, Honglei, You, Jingwen, Yucheng, and Jinzhi [23] developed a deep learning model based on LSTM for predicting traffic accident risk using recurrent neural networks. Each record contains the date and time of the accident, as well as the GPS (Global Positioning System) coordinates. The model is capable of inferring intricate relationships between traffic accidents and their spatial-temporal patterns. The findings indicate that the risk of traffic accidents is not uniformly distributed across space and time. It exhibits strong periodical temporal patterns and spatial correlation at the regional level.

The authors of [24] intend to compare the predictive performance of various machine learning and statistical methods with distinct modeling logic for crash severity analysis, including prediction accuracy and estimation of variable importance. The data on traffic flow, road geometry, and crash severity were collected at Florida freeway diverge areas. They estimated two widely used statistical methods, ordered multinomial logit and probit (OP) models, as well as four widely used machine learning methods, Random Forest (RF), Decision Tree, K-Nearest Neighbor, and Support Vector Machine. The outcome indicates that machine learning techniques outperformed statistical methods. The RF method predicted the most accurately in both overall and severe crashes, whereas the OP method predicted the least accurately.

Recently, in 2020, Natalia-Casado-Sanz et al. [25] used a multinomial logit (MNL) model to determine the most significant factors affecting the severity of driver injuries. Based on 1064 accidents that occurred on Spanish crosstown roads between 2006 and 2016. According to statistical analysis, factors such as low traffic volumes, infractions,

lateral crosstown roads, a higher proportion of heavy vehicles, wider lanes, and finally, the absence of road markings all contribute to the severity of drivers' injuries.

Ali Ghandour et al. [26] proposed a hybrid ensemble classifier model based on sequential minimal decision trees to determine what factors contribute most to fatal road accidents. They analyzed 8482 road crash incidents from the Lebanese Road Accidents Platform (LRAP) database, in this model, fatality occurrence is the outcome variable. A significant association was found between seven out of the nine independent variables and fatalities.

The authors of [27] employed the random undersampling of the majority class (RUMC) technique to deal with imbalanced accident datasets and improve minority class prediction. To deal with imbalanced data in a classification problem, they used an imbalanced and a balanced RUMC-based training set. They propose training, validation, and evaluation for four widely used machine learning methods: random forest, logistic regression, k-nearest neighbor, and random tree. To evaluate the performance, the accuracy, F1-score, false positive rate, true positive rate (recall), true negative rate, precision and confusion matrix were calculated. According to our findings, applying RUMC models to classifiers improves their ability to detect fatal and injury-causing collisions.

Chapter Three
Methodology

Our approach to predicting the risk factors that contribute to the severity of an accident consists of three steps: 1) collecting a large number of accident data sets, including information about drivers and road conditions, 2) preprocessing the data and feature extraction, and 3) the application of a classification algorithm to extract patterns. In the following subsections, we describe these steps in more details.

3.1 Data Set

The most important problem that we encountered is the lack of a data set about accidents in Palestine. To address this problem, we considered two data sets in this thesis. The first one is a small data set (523 accidents) that we gathered from written reports obtained from a local insurance company in Palestine. The goal of using this small data set is to keep the study focused on Palestine Road accidents. However, since the data set is small and its source is only from a single insurance company, the prediction model can be inaccurate. Therefore, to evaluate the performance of our approach, we considered another large data set available online. This data set contains 10485747 accident data in the United States. Regarding the machine learning algorithm, we compared the performance of three well-known classification methods, Support vector machine (SVM), Random Forest (RF), and Deep Neural Network (DNN) and evaluated the effectiveness of each method in predicting accident severity.

In the following subsections we explained the main attributes of these data sets.

3.1.1 Data from Palestine

For this study, we obtained over 600 insurance records from a single insurance company in Palestine, from which we extracted key information regarding the accidents. These files solely contain information on drivers and do not include information about passengers. The study's initial sample includes traffic accident information from 2013 to 2018. After that, we used Excel to insert 523 records into the

computer. There are 16 attributes in each accident report. The severity of the accidents was divided into two categories: heavy and light. Table (3.1) shows the list of these attributes.

Table (3.1)

16 Specific Attribute Class Labels With Data Type And Description.

#	Attributes Name	Description	Data Type
1	Accident number	Unique number given for each traffic accident.	Numeric
2	Date of accident	The date of accident (D/M/Y).	Numeric
3	Year of accident	The year of accident.	Numeric
4	Place	The place where the accident occurred.	Nominal
5	Time	The time of the accident occurs.	Numeric
6	Type of car	Type of car (Peugeot, Mercedes, Golf,).	Nominal
7	Model of car	The year it was manufactured.	Nominal
8	Gender	Gender of the driver.	Nominal
9	Age	Age of the driver.	Numeric
10	Number of Years Possession License	Number of years that the driver has a license.	Numeric
11	Type of insurance required	Bodily Injury, comprehensive, third-party liability.	Nominal
12	Injured in the accident	Number of people injured in the accident.	Numeric
13	Responsible for the accident	The driver whom causing the accident.	Nominal
14	Severity	Its expressed severity of the accident by the money which cost.	Numeric
15	Vehicles affected	Number of vehicles involved in the accident.	Numeric

The following Figures shows some characteristics of the Palestine dataset

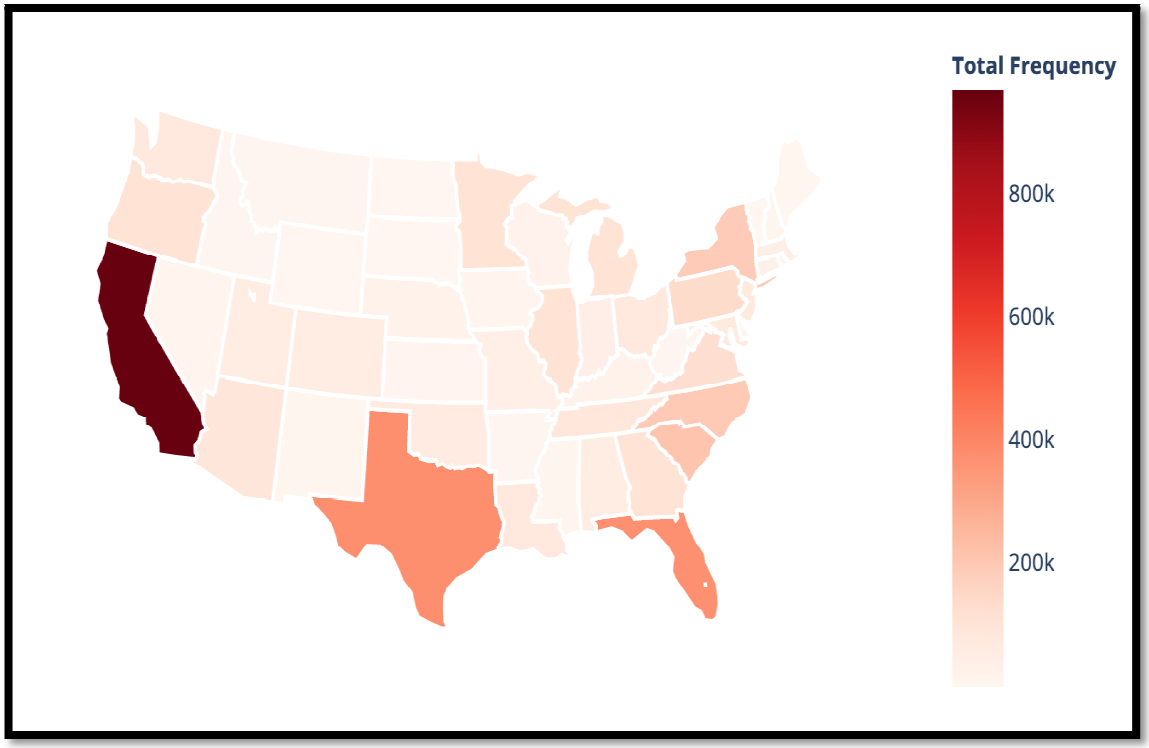
3.1.2 The U.S. government's data:

From February 2016 to March 2020, the data was collected.

This data set, which covers 49 states in the United States, was compiled from a variety of data sources, including APIs that provide streaming traffic event data. The APIs broadcast information gathered from various sources, including traffic cameras, state transportation departments, and road network traffic sensors. The diagram in Figure (3.1) in Appendix (B) depicts the current data distribution among all states. The data collection includes 49 states in the United States.

Figure (3.4)

The current Data Distribution Over All State[13].



The data is provided in the form of a CSV file. It contains 47 columns (attributes), and 1048575 entries. Each row represents one crash. Table (3.2) in Appendix (A) describes the list of the data attributes.

We solely used data from road accidents in the United States for this investigation. Numerical and nominal data characteristics are the two types of data attributes (categorical attribute). There are 21 numerical and 26 category features of this data set.

3.2 Preprocessing of data and Feature selection

Data is pre-processed in this step by addressing missing data and removing outliers. In addition, the data set is prepared for machine learning techniques, which comprises text and category attribute encoding, as well as feature scaling, as shown below.

3.2.1 Handling Missing values

Missing values for some attributes are another key issue for machine learning. Data samples with missing values are removed, resulting in a significant reduction in the size of the training data set, which may result in poor performance if the data set is not large enough. There are several approaches to handle missing data includes include filling in with some values.

In some applications, it is permissible to input missing values as zero. In the absence of a meaningful value for zero, the model learns that these values represent missing data, and they are ignored. The network will not be able to disregard missing values if you expect missing values in the test data, and the network was trained on data without missing values. For these cases, it is preferable to create training samples with missing entries artificially by reusing some training samples many times and removing some of the features that are expected to not be present in the test data. [28].

Missing data techniques can be divided into simple traditional techniques and advanced techniques:

1. Traditional methods where we can divide it into two types:

- Deletion: The best avoidable method in many cases is deletion.
- Imputation: The practice of replacing missing data with approximated values based on information in the data set is known as imputation methods. There are a variety

of forms, ranging from simple methods like mean imputation to more powerful methods based on attribute connections [29].

2. Advanced methods: Machine learning approaches provide computers the ability to learn from data and handle missing values without having to be explicitly programmed [29].

In this thesis, we adopted traditional methods because these methods are faster than the advanced imputation methods[30]. Also, because our data is large enough, we used the deletion technique as well.

3.2.2 Handling text and categorical attributes

Because all input and output variables in machine learning models must be numeric, categorical data must be transformed into numerical forms before being employed in machine learning algorithms. Label encoding and one-hot encoding are the most frequent methods for encoding categorical data. Using n-dimensional binary vectors, one-hot encoding expresses categorical variables with n classes. The binary vector assigns a value of 1 to the class label and a value of 0 to all other places. Dummy variables are the output of one-hot encoding in classification issues.

3.2.3 Feature Scaling

There are certain machine learning algorithms that are highly sensitive to features with large magnitudes or fluctuations. When the features are scaled, all gradient descent optimization methods, including the Logistic, ANN, and Linear Regression, converge faster. A measure of distance is skewed toward larger values, so algorithms such as KNN and SVM, which measure similarity by distance, are heavily dependent on feature scaling. Algorithms that rely on trees, such as decision trees or random forests, are resistant to feature scaling since they divide each node based on only one characteristic - the one leading to impurities - and do not use other factors.

Literature describes two different types of feature scaling: standardization and normalization. Normalizing features is typically performed when they do not follow a specific distribution and when using algorithms, such as KNN and ANN, do not assume any distribution [12]. Standardization is typically utilized when it is reasonable to assume that the data follows a normal distribution.

A typical technique for normalization is min-max scaling, in which data are shifted and rescaled to fall within the range of 0 to 1 or -1 to 1. This is accomplished by subtracting the min value from the max value and dividing by the max minus the min. The standardization procedure is different, first it subtracts the mean value, then it divides that value by the variance so that the distribution will have a unit variance and a zero mean. Standardization is different from min-max scaling because values are not bound to a range, which may present a problem for some algorithms (e.g., neural networks converges faster when the input values are in the range of 0 to 1. However, outliers have a significantly smaller effect on standardization[31] .

3.2.4 Classification method

Classifying items into one of a predefined set of categories or classes is a characteristic of intelligence that has attracted the interest of computer scientists. Identifying the "core" features shared by a group of objects that are typical of their class is extremely useful for focusing the interest of a person or computer program.

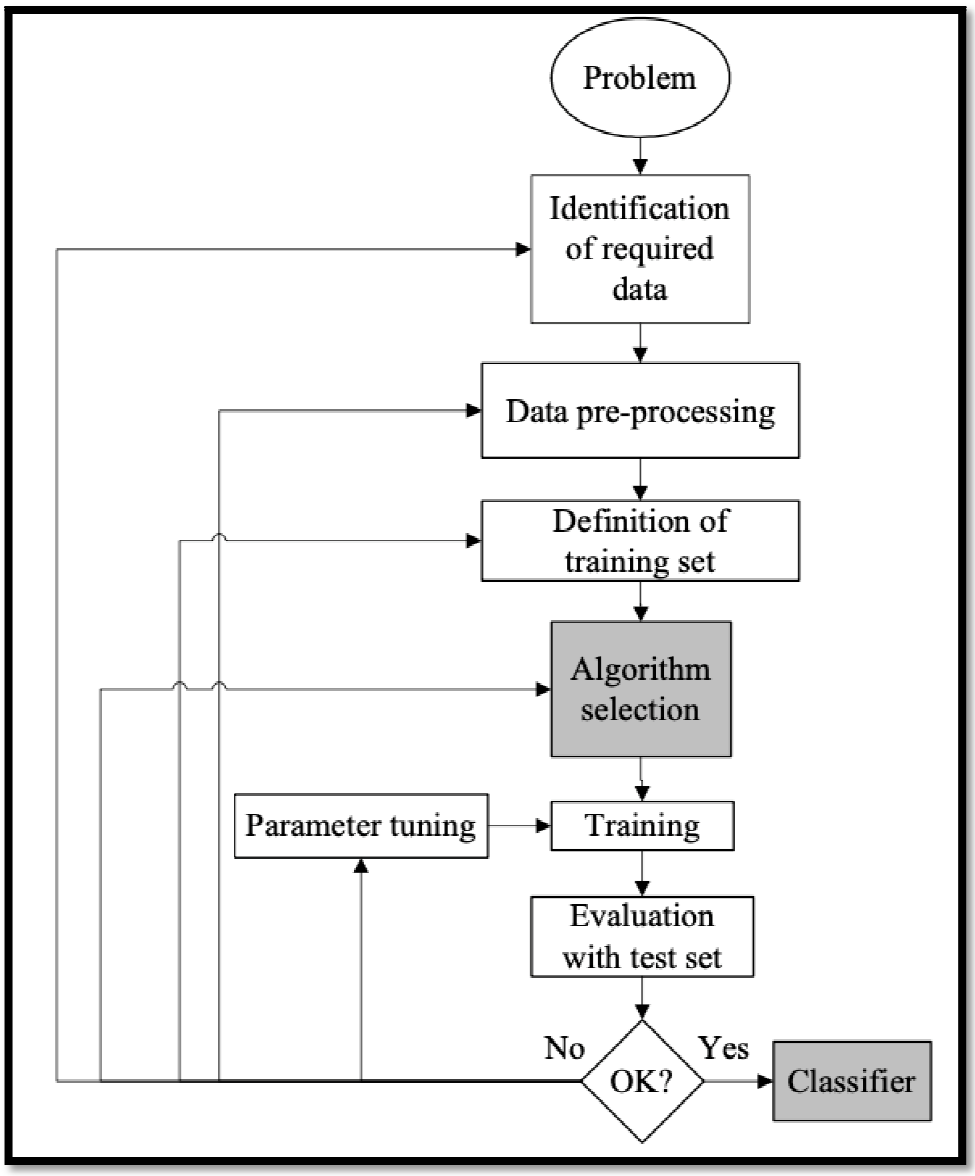
Our technique is a classification task that entails a detailed analysis of known risk factors such as driver features, vehicle attributes, risk exposure (traffic volumes), traffic control, weather conditions, and route design characteristics.

The categorization task mentioned above is typically referred to as supervised learning in machine learning. In supervised learning, a predefined set of classes is used, and sample objects are labeled appropriately. The objective is to generalize (using class descriptions) from the training examples in order to identify fresh things as belonging to

one of the classes. The process of applying supervised ML to areal-world problem is described in Figure (3.2).

Figure (3.5)

The Process Of Supervised Machine Learning[32].



A faulty algorithm and bad data are two problems in machine learning that could result in inaccuracies in the output. If the training data contains errors, outliers, and noise, the method won't work well. This makes it more challenging for the system to identify the underlying patterns. The work spent cleaning the training data set is frequently more than worthwhile.

Three common machine learning methods are used throughout this thesis: Artificial Neural Networks (ANN),Random Forests (RF), and Support Vector Machines (SVM). These techniques are briefly detailed in the sections below.

3.3.1 Artificial Neural Network (ANN):

Mathematical models of neural networks use learning algorithms that mimic brain activity to store information. Research in machine learning aims to teach computer systems to recognize complex patterns and make intelligent decisions based on data. Figure 1 shows an example of a neural network (3.3). An artificial neural network that works only from input to output is known as a feed-forward neural network. Show figure (3.6) in appendix (B).

The primary reason for utilizing artificial neural networks in our thesis is their capability to analyze data with a high number of dimensions. Many of the interactions between inputs and outputs in real life are both non-linear and intricate, and ANNs can learn and simulate non-linear and complex relationships. For an ANN system to learn, it has to update its settings so it can execute a certain task more effectively. The network must be able to learn the weights of its connections using accessible training patterns.

Iteratively updating the weights in the network improves performance. The capacity of ANNs to learn automatically from examples makes them attractive and interesting. Rather than adhering to a set of rules provided by human experts, it appears as though the system learns underlying rules (such as input-output relationships) from a collection of representative cases. This is a significant advantage that neural networks have over traditional expert systems [33].

Various challenges arise while developing networks, including the following: The primary challenge in implementing ANNs is determining the number of hidden layers and neurons required for a specific task. The question of how many hidden layers and nodes should be present in a neural network arises in any classification task involving remotely sensed input. There has been no definitive solution to this problem until today.

We chose only one hidden layer in this thesis because, for the vast majority of issues, one hidden layer is sufficient, and there is no theoretical reason to employ more than two hidden layers[34].

The ideal number of neurons in the hidden layer must be determined. An inadequate number of neurons may result in either under- or over-fitting. The number of neurons in the hidden layer is determined using a trial-and-error procedure in which the ANN accuracy is assessed for various hidden layer densities and the number of neurons with the highest output accuracy is chosen for further investigation. In addition, it ought to be noted that the structures of ANN models are affected by repeated training, because initial weights are generated randomly and training data is extracted randomly, resulting in varying reliability of projections. To minimize oscillations caused by random initializations, the trained ANN model is repeated ten times.

3.3.2 Random forest (RF)

Decision trees, as the name implies, can be used to visually and explicitly display information on the decision made regarding a query case. In order to classify a query q, we traverse the tree to reach its root, which is the highest decision node.Each internal node contains a test which determines which of its subtrees is traversed for q, which belongs to a leaf node, which represents a classification or decision[35]. Datasets for classification research are usually divided into training sets and testing sets. A decision tree model is developed by training a set of data while testing a set of data to determine its predictive accuracy.

As a computationally efficient method that works rapidly across large datasets, random forest builds a lot of decision trees that can then be used to classify a new instance by majority vote. This method is used to build many decision trees in recent research projects and real-world applications. In each node of the decision tree, a subset of attributes is randomly selected from the original attributes. Further, each tree is bootstrapped based on a different set of sample data in the same manner as bagging: by

replacing each set of data with a new one. There are no guidelines on how many trees should be used in a Random Forest in the relevant literature [36] .

For each observation, each individual tree votes in the random forest spits out a class prediction and the class with most votes become our model's prediction. The Gini[37]index is a splitting criterion that determines how a decision tree was split. The largest possible tree is grown and not pruned. As we said above the root node of each tree in the forest contains a bootstrap sample from the original data as the training set.

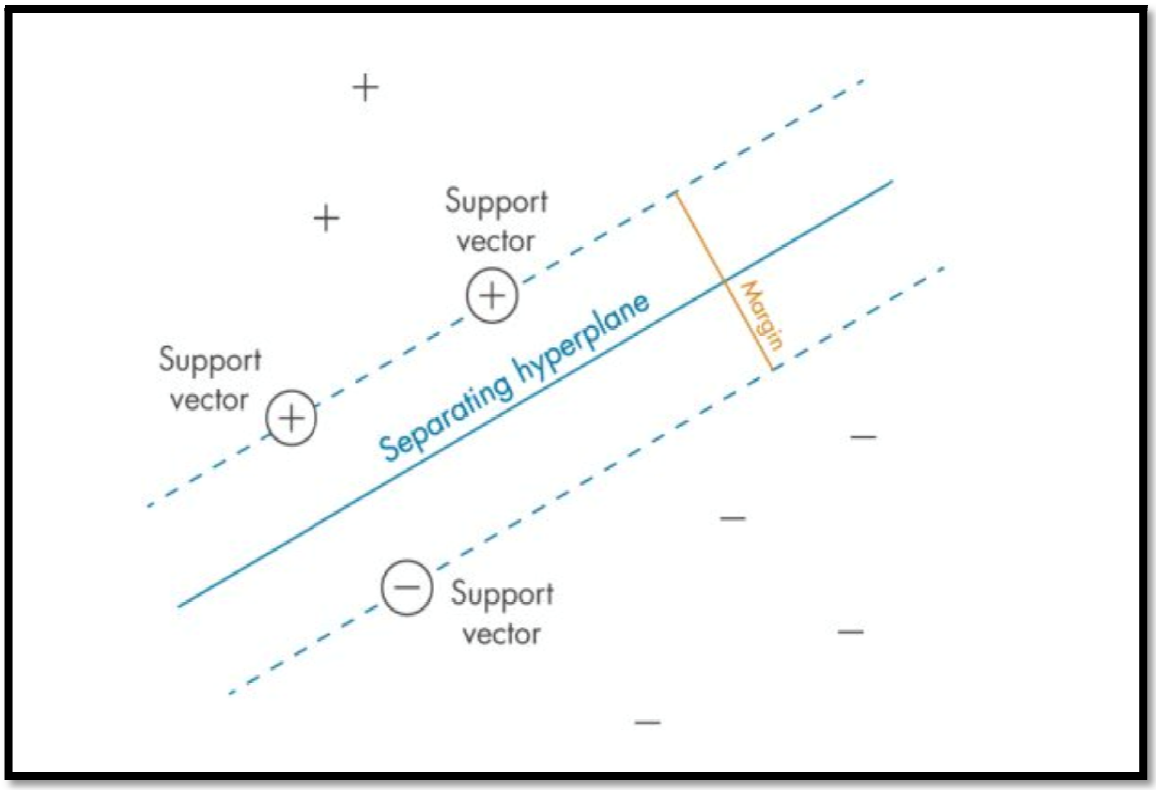
3.3.3 Support Vector Machine (SVM)

A support vector machine (SVM) is a supervised learning algorithm used for many classification and regression problems. The basic concepts of SVM can be summarized as follows [10]:

1. The separating hyperplane: the goal of the SVM algorithm is to find a hyperplane that, to the best degree possible, separates data point of one class from the other class.
2. The maximum- margin hyperplane: we seek to find a hyperplane that separates two classes so that the margin between them is as large as possible, in the Figure (3.4) represented by plus versus minus. The margin means the maximum width of the slab parallel to the hyperplane without internal data points.
3. The soft margin hyperplane: This hyperplane can only be found for problems that can be linearly separated, in most specific problems the algorithm allowing a few numbers of misclassifications by maximizes the soft margin.
4. The kernel function: the kernel function is a mathematical trick projects data from a low-dimensional space to a space of higher dimension.

Figure (3.7)

Defining the 'margin' between classes (the standard that SVMs aim to optimize) [22].



3.4 Tuning hyperparameters

The simplest technique for tuning the hyperparameters is to perform a grid search and select a set of candidate values for each hyperparameter. This is accomplished by training models with all possible values and selecting the configuration that produces the lowest validation error [20].

3.5 Techniques for Reducing overfitting

The generalization error can be improved by reducing overfitting. Several strategies exist that are both effective in practice and have robust theoretical foundations. A decent neural network will typically integrate many of these strategies.

- Early stopping: the training error and the validation errors generally continue to improve as the training progress. When the training error starts to increase that means the model start to overfit. At this point the training should stop.

- Regularization: regularization is one of the most fundamental concepts in machine learning, and many theoretical justifications have been proposed. Regularizes are sometimes viewed as penalizing the “complexity” of a network, or favoring explanations which are “more likely”.
- Stochastic regularization: the most popular form of stochastic regularization is dropout. The algorithm itself is simple: we drop out each individual unit with some probability by sitting its activation to zero.

Chapter Four

Results

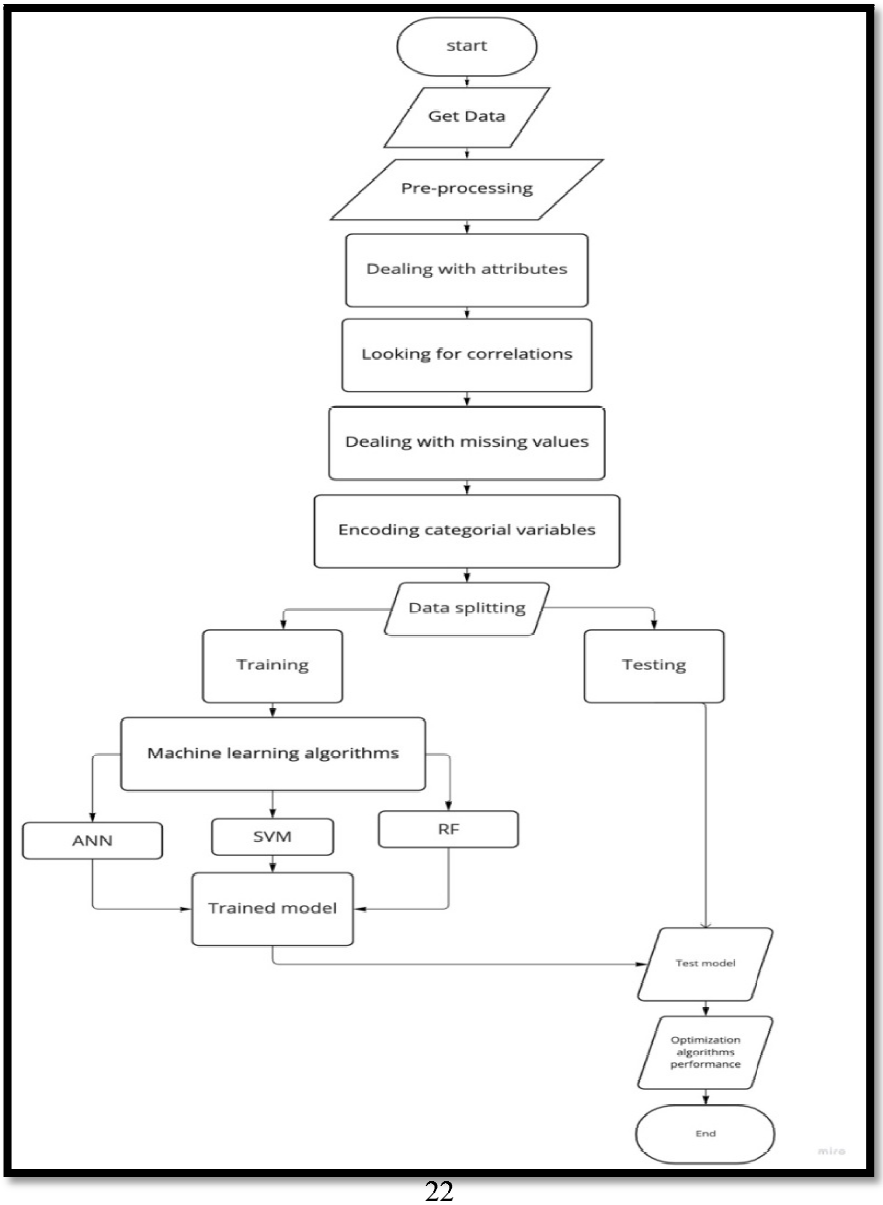
In this thesis, we present a method consisting of three stages:

1. Preprocessing.
2. Tuning Parameters.
3. Classification.

Figure (4.1) defines the steps of the supervised machine learning technique that we used in this thesis.

Figure (4.1)

Supervised machine learning process

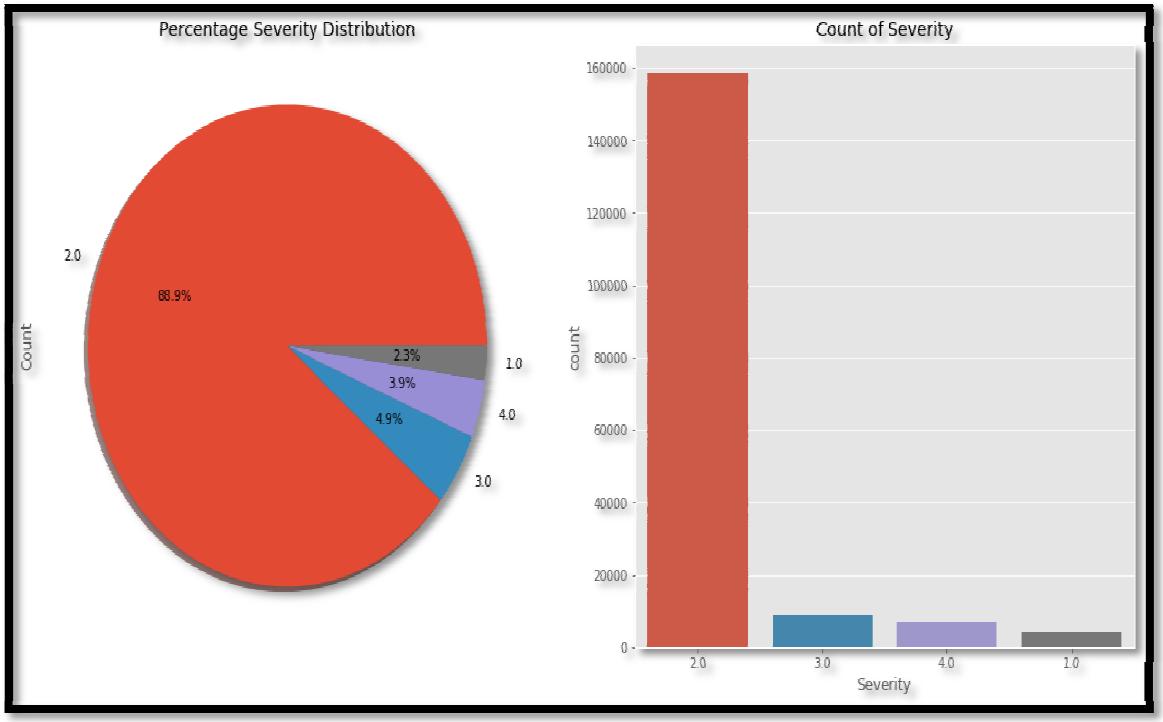


A nationwide dataset covering 49 states of the country was used to apply the methodology. The data was collected utilizing "APIs" that collect incident data from 49 states between February 2016 and December 2020. The data is gained by a variety of sources, including the US Department of Transportation, state transportation departments, police, traffic sensors, and traffic cameras on the highways. The dataset contains approximately a million data points (incidents).

Figure (4.2) shows the percentage severity distribution. In Table (3.2), In this study, we describe the characteristics of the dataset. The dataset is available on an online website [5]. The dataset is in a CSV file. Apart from the column ID, which is an identifier of the accident record, there are 46 different features. The variable "severity" is the observation outcome with a number between 1 and 4, where 1 indicates the least impact on traffic (i.e., a short delay), and it is the variable that we would like to predict. There are 20 attributes with object datatype, 13 attributes with Boolean datatype, and 13 attributes with float datatype. The total number of observations in the CSV file that was used in this study is 1048575.

Figure (4.2)

The Percentage Severity Distribution



4.1 Pre-processing in Python:

The Python programming language was used in this project for computing mathematical and statistical information. Data analysis and predictive modeling can be done with this platform, which is flexible and powerful. It contains 47 columns and 1048575 entries (rows) in a data frame format.

4.1.1 Combining features:

We occasionally need to look closely at the attributes and may need to change the way some of them are formed.

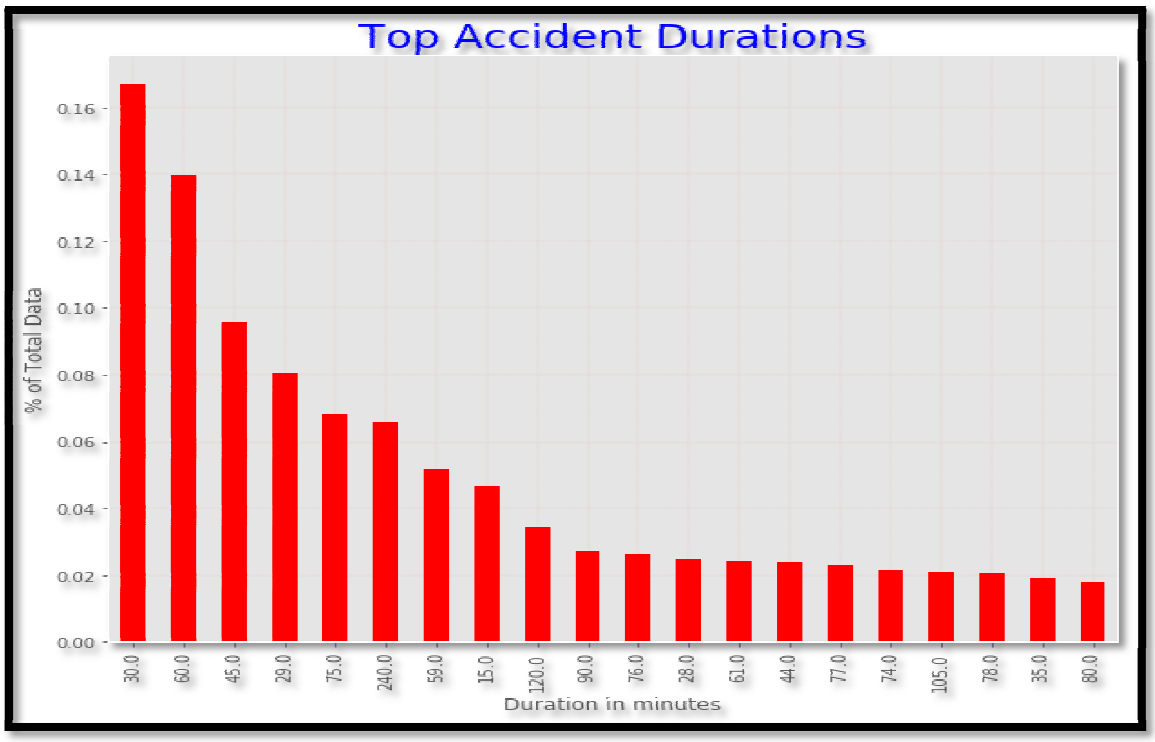
1. The time duration refers to the time difference between the end and start of an accident in minutes. We derived this attribute Time as follows:

Time_Duration= (End_time) - (Start_time),

Figure (4.3) Shows the top 20 longest accidents correspond to 84.6% of the data.

Figure (4.3)

Top 20 Longest Accidents Correspond To 84.6% Of The Data



1. We divided the Start Time variable into the following attributes: year, month, day, hour, and weekday.

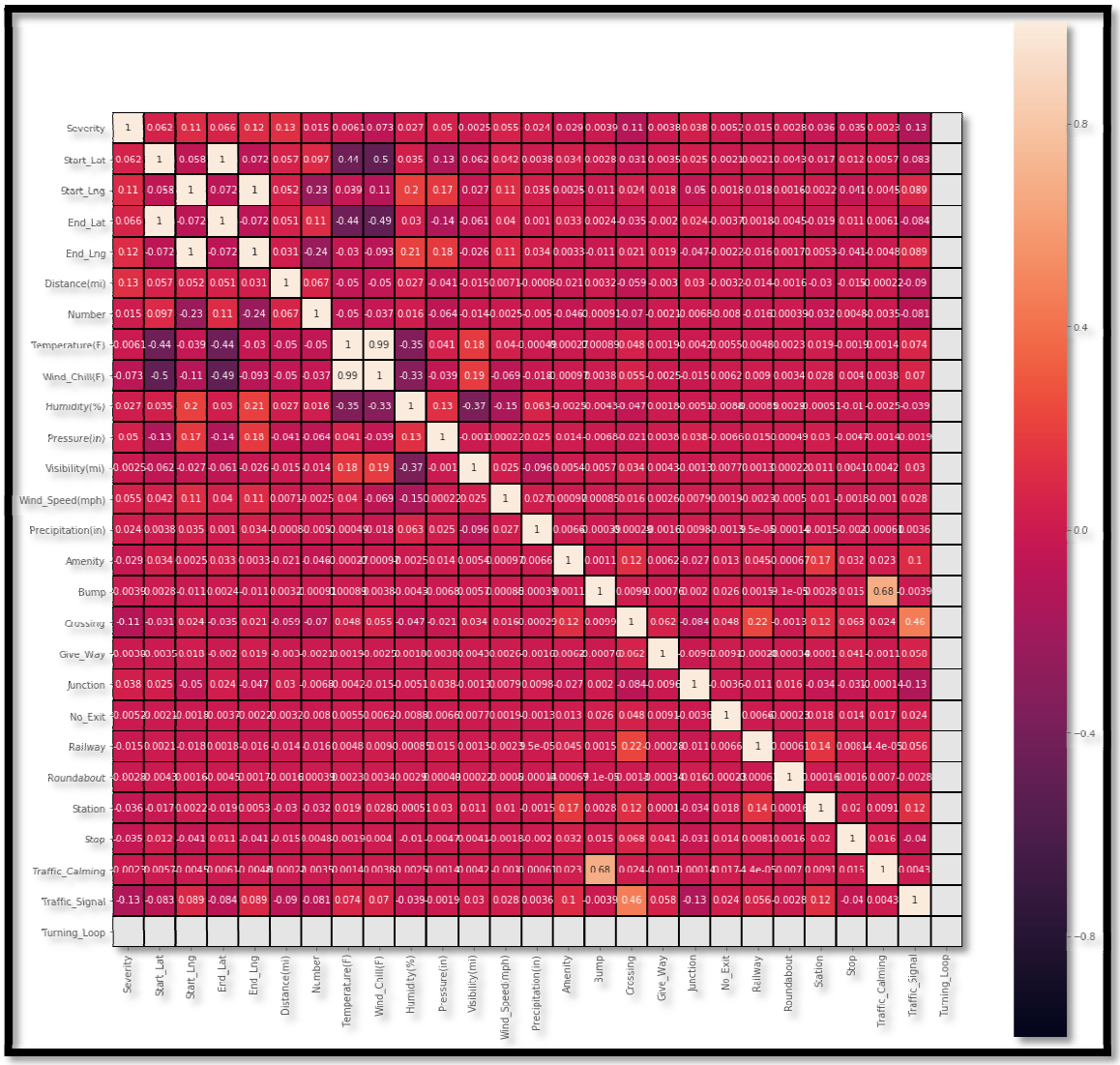
2. Eliminating unnecessary attributes: since the entire dataset belongs to a single country, we eliminated the attribute "Country" because it contains just a single value. We also removed the attribute "Turing loop" from the entire dataset because it has a single value. Because the ID columns serve only as a unique identifier and does not contain any information on the incidents themselves, we eliminated the "ID" characteristics. And as we can see from the correlation map, there is a strong link between "Wind Chill(F)" and "Temperature(F)", we can eliminate either "Wind Chill(F)" or "Temperature(F)" column, we eliminated "Temperature(F)" attribute.

4.1.2 Looking for correlations

The correlation between features can be used to pick features for classification tasks in machine learning. This method of feature selection is advantageous for standard machine learning algorithms. Therefore, the next step is to exclude those attributes that are significantly correlated with others. This is accomplished by computing the standard correlation coefficient (a.k.a. person r) between any two attributes using Python's corr() function. This is seen in Figure (4.5), where the data value in each cell is represented by the white and black colors, respectively, which correspond to the positive and negative relationships.

Figure (4.4)

Shows Correlation Map Between Every Pairs.



We excluded those attributes having a correlation of 0. between the target attribute "Severity" and the other attributes based on the correlation between the target attribute "Severity" and the other attributes. Figure (4.5) depicts the correlation map for the remaining attributes. The other attributes are listed in Table (4.1), along with their correlation coefficients with the target attributes.

Figure (4.5)

Correlation Map Between The Rest Attributes

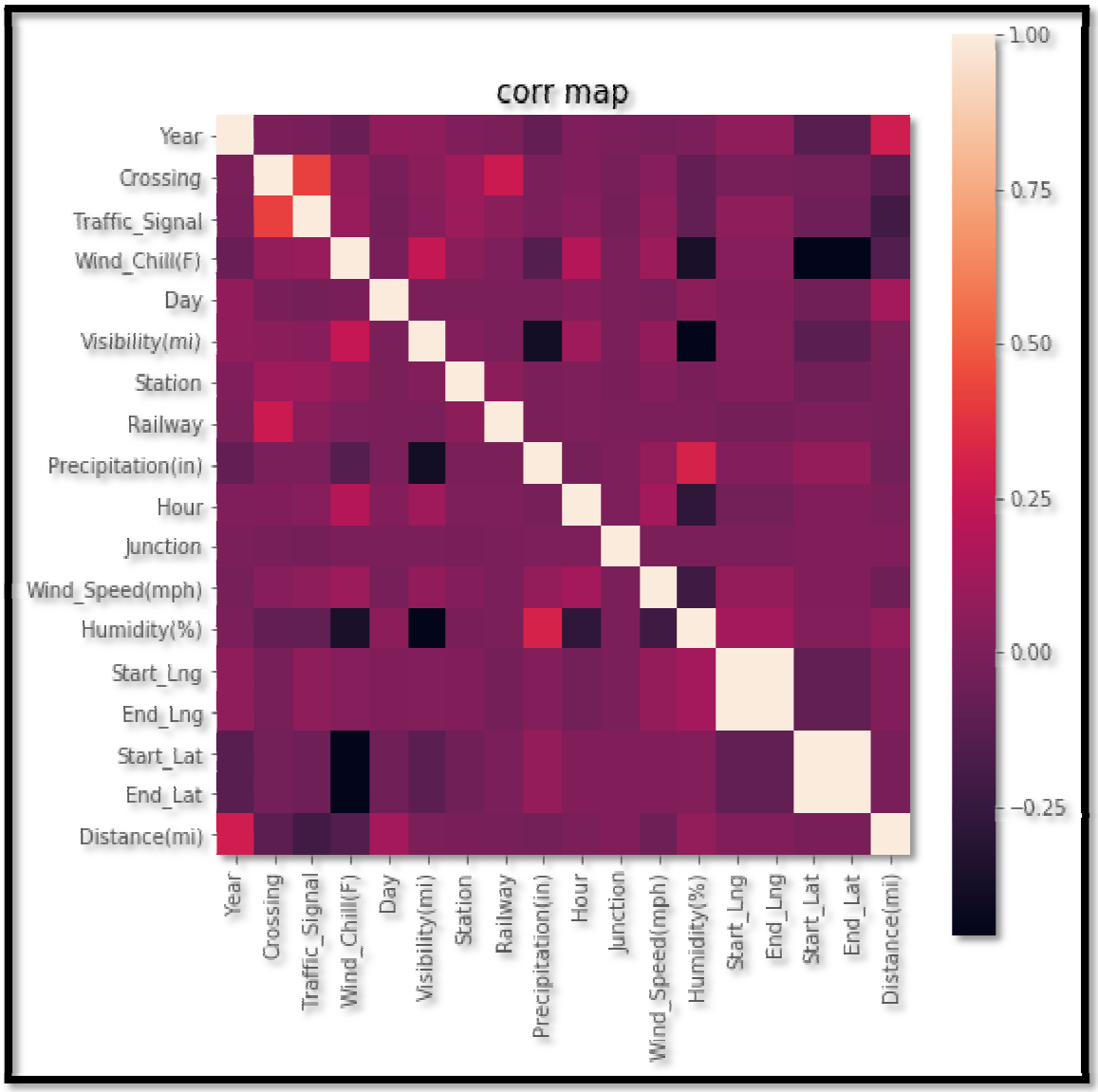


Table (4.1)

The Reset Attributes Used In This Study

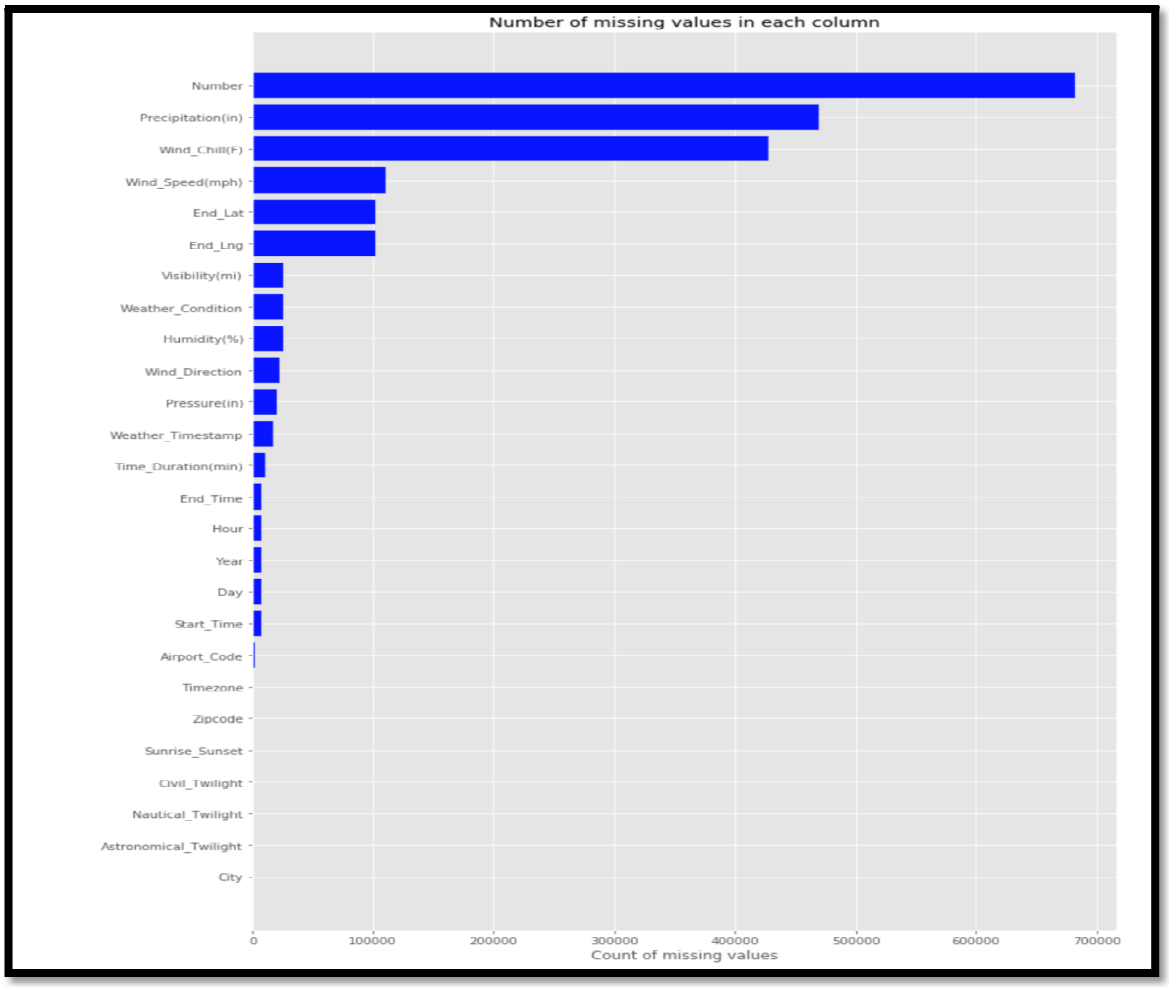
Columns labels	Correlation values
Year	-0.0119442
Crossing	-0.081033
Traffic_signal	-0.055001
Wind_chill(F)	-0.053728
Day	-0.021813
Visibility	-0.021361
Station	-0.018061
Railway	-0.011697
Precipitation(in)	0.015644
Hour	0.020009
Junction	0.025773
Wind_Speed(mph)	0.028030
Humidity(%)	0.042282

4.1.3 Handling missing values

There are various approaches to addressing missing values, and because the data set is sufficiently large, we have deleted each observation with a missing value. A bar plot of missing values is depicted in Figure (4.6).

Figure (4.6)

Missing Values Bar Plot



4.1.4 Handling text and categorical attributes

One of the most widely used encoding methods is one-hot encoding. It compares each categorical variable level to a predetermined reference level. Each observation has a binary vector with only one value, presence (1), and the remaining values, absence (0). If a single variable has n observations and d different values, one hot encoding produces d binary variables with n observations each [38].

Using one-hot-encoding , we can represent any number of categories by introducing one new feature per category. For simplicity, we encode each category with a different binary feature.

When we work with pandas and machine learning models, it is common to use the *get_dummies()* method to convert the categorical variables to dummies. The get_dummies() function automatically transforms all columns that have object type.

4.2 Data splitting

Increasing the amount of training samples improves the performance of machine learning algorithms in general. Adding more varied samples would increase performance and reduce variation in the classification model's predictions.

If the training data is noisy or unrepresentative, the model's performance may suffer and deteriorate. At some point, adding more data will not result in a significant change in the model fit. Most models can attain a stable state with a sufficient number of samples, but adding more samples would need more time and computational resources.

The data set is split into 80 percent training data (826148 samples) and 20 percent testing data for our implementation of the Support Vector Machine, Random Forest, and Artificial Neural Networks algorithms (206537 samples).

4.3 Performance Metrics

As previously stated, a wide variety of performance metrics allows for a thorough assessment and comparison of machine learning algorithms.

For a binary classification problem, the confusion matrix is defined as shown in Figure (4.9) in Appendix (B), which is a table that helps us visualize a classification models performance on a set of test data for which the actual values are known, where

- TP, is the number of True Positives, i.e., occurrences accurately identified as belonging to positive class (class 1)
- TN, is the number of True Negative instances, i.e., examples correctly categorized as belonging to the negative class (class 0).
- FP, is the number of instances from the negative class that were mistakenly classified as belonging to the positive class.

- FN, is the number of instances from the positive class that were mistakenly classified as belonging to the positive class.

Equations (1)-(7) describe the classifier's overall accuracy and precision, as well as the TPR, FPR, TNR, FNR, and F1-score.

$$\text{Accuracy} = TP + TN/(TP + FP + TN + FN) \quad (1)$$

$$\text{Precision} = TP/(TP + FP) \quad (2)$$

$$\text{TPR} = TP/(TP + FN) \quad (3)$$

$$\text{FPR} = FP/(FP + TN) \quad (4)$$

$$\text{TNP} = TN/(TN + FP) \quad (5)$$

$$\text{FNR} = FN/(FN + TP) \quad (6)$$

$$F_1 = \frac{TP}{TP + \frac{FN+FP}{2}} \quad (7)$$

The classifier's overall accuracy is meant to reflect the classifier's overall performance. The precision metric indicates the accuracy of positive predictions. The recall of a classifier refers to its ability to correctly identify positive instances its essentially the ratio of true positives (TPR) to all positives in ground truth. F1-scores are used for comparing classifiers because they combine precision and recall.

4.4 Machine learning algorithms implementation

Following analysis of the pre-processed data, three common machine learning algorithms were developed (Support Vector Machine, Artificial Neural Network, and Random Forest), and the resulting performance of the built models is quantified and visually evaluated.

4.4.1 Artificial Neural Networks(ANN)

A neural network is a mathematical model that stores information by using learning algorithms based on brain activity. Machine learning research is primarily concerned with automatically learning to recognize complex patterns and making intelligent decisions based on data [12].

There are various packages and methods in Python for creating models of artificial neural networks, such as sequential model (), which includes several methods such as add (), compile (), and fit (). The sequential model is a stack of layers in a linear fashion. By using the model.add () method, we can easily add layers. Once the model is constructed, it can be configured with different sorts of losses and metrics in compile () method. To train the model, use the model.fit () method, and to predict, use the model.predict() method.

After creating the ANN model, the summary () function can be used to obtain various model-related information. Figure (4.8) in Appendix (B) shows the model, which is made up of three layers: the input layer has 88 nodes, the hidden layer has 10 nodes, and the out layer has four nodes, with 934 parameters.

4.4.1.1 Selecting Tuning Parameters For ANN

Manual parameter tuning can be extremely time consuming, especially when the learning algorithm includes a large number of parameters. Numerous tuning parameters are required for an ANN model, including the number of hidden layers, batch size, optimizer, and so on. One approach to accomplish this is to manually adjust the hyperparameters until you find an optimal combination of hyperparameter values. This would be quite time consuming, and you may not have the time to investigate numerous combinations. Rather of that, we rely on Scikit-Learns GridSearchCV to perform the search for us. Initially, the grid search was an exhaustive search of a subset of the hyperparameter space [39]. We provide it with hyperparameters and values to experiment with, and it evaluates all possible combinations of hyperparameter values.

Each combination's performance is evaluated using a variety of performance indicators; in this thesis, we used cross-validation.

A specific candidate batch size is tuning (8, 16,32, and64), and three different optimizers (Adam, Nadam and SGD). Figure (4.9) in Appendix (B) shows that by running one round of the fit () function, 12 ANN models would be generated, each with different tuning parameters.

Where Keras and sklearn is an open source deep learning framework for the python programming language.

Grid search optimizes the ANN parameters (batch size, optimizer, etc.) based on the performance metric cross validation (CV). The objective is to develop effective hyperparameter combinations that enable the classifier to reliably predict unknown data. As is well known, non-deterministic algorithms, such as those in ANNs, can produce completely different results depending on the initialization parameters, even when the parameters are constant. As a result, before determining the batch size and optimizer, we divide the available data into k subsets (we set k=10). One subset is utilized as testing data,for the rest k-1 of the training subsets, the model is evaluated. Then, we calculate the CV error for the ANN classifier using the split error for various batch size and optimizer values. An ANN is trained on the entire dataset using a number of hyperparameter combinations, with the one with the highest cross validation accuracy (or the lowest CV error) being used.

In this thesis, the fit () function was running ten times independently to determine the tuning settings with the highest quality. Each run, the optimal outcome with the greatest score (or the minimum loss) was chosen among 12 ANN models.

The following Table contains all 10 optimal solutions for the ten instances of the fit () function (4.2). As can be seen, the batch size = 16 is mentioned nine times. Additionally, the optimizer= Adam is mentioned six times. The largest number is the best score of the ANN model with batch size = 16 and optimizer = Adam.

This thesis chooses as the tuning parameters of the ANN model the batch size = 16 and the optimizer = Adam, which are based on the optimal results of the fit () function.

Table (4.2)

Fit ()'s Optimal Results

K Number	Best Parameters		Best Score
	Optimizer	batch_size	
1	Adam	16	0.9081
2	SGD	16	0.9061
3	SGD	16	0.9103
4	Adam	16	0.9151
5	Adam	32	0.9096
6	Adam	16	0.9101
7	Nadam	16	0.9051
8	Adam	16	0.9122
9	Adam	16	0.9144
10	SGD	16	0.9091

4.4.1.2 Building ANN Model

As we mention in methodology chapter we choose only one hidden layer, and following a trial-and-error procedure, it is determined how many neurons are in the hidden layer based on the ANN accuracy. This entails comparing the ANN accuracy of the different numbers of neurons in the hidden layer, and selecting the neuron number that produces the best outputs. As a result of the random generation of initial weights and random extraction of training data, the construct ANN models obtained from multiple repeating training are different, resulting in different accuracy of the predictions. ANN models are thus trained 10 times so that random initializations are eliminated from the model. Table (4.3) in appendix (A) shows the average accuracy for each number of neurons.

Using the statistics illustrated in Table (4.3), the optimal number of neurons in the hidden layer was taken to be 10, 60, and 50.

An artificial neural network with three layers is trained in this study using the fit () algorithm. The input layer of the ANN model contains 88 nodes, the hidden layer contains 10 nodes, and the output layer contains 4 nodes. As shown in Figure (4.10) in Appendix (B), we choose (***relu)*** activation function enhances the performance of an overfit learning neural network model. Regularizes enable the application of penalties to layer parameters or activities while optimizing. These penalties are added together and used to optimize the network's loss function. The following function is used to calculate the L2 regularization penalty:

*loss = l2 * reduce_sum(square(x))*

l2: float; L2 regularization factor. In our case l2=0.0001.

Input_dim is the input shape that specifies the dimensionality of our data.

The two parameters batch size and optimizer are set 16 and Adam (learning rate = 0.01) which have been chosen in the last section. We choose random uniform distribution to set the initial random weights of keras layers.

4.4.2 Random Forest(RF)

There are several tree classifiers in the random forest classifier, and each of them casts a unit vote for the most popular class to categorize input images. Each of the tree classifiers is constructed using a random vector sampled independently from the input vector [11]. The keras package's primary function fit () requires various tweaking parameters: The number of trees in the forest *n_estimatores*, the maximum depth of the tree *max_depth* and so on.

4.4.2.1 Selecting Tuning Parameters For RF

Aiming at tuning the two parameters, *n_estimatores* and *max_depth* the GridSearchCV is still suggested to be used. We can train over this parameter in Figure (4.11) in Appendix (B) as follows:

Depending on GridSearchCV first, we split the available dataset into k subsets (we set k=10) randomly one subset is used as a testing dataset, and then evaluated using the remaining (k-1) training subsets. Finally, we calculate the CV error using the split error for the RF classifier using different values of n_estimatores andmax_depth.

Therefore, the fit () Function is also called ten times independently to ensure that the table tuning parameter is selected with the greatest accuracy possible. To find the optimal model with the greatest value, the accuracy parameter is additionally used in the single fit () function (4.4). The optimal values of n_etimatores = 500 and max_depth = 55 is repeated three time with the best score = 0.93223 for accuracy so we adopted these values to the final parameters for our model.

Table (4.4)

Fit ()'s Optimal Results

K Number	Best Parameters		Best Score
	Num_estimators	Maximum_depth'	
1	100	55	0.93154
2	500	40	0.93099
3	500	40	0.93091
4	500	55	0.93216
5	500	55	0.93223
6	500	55	0.93137
7	100	55	0.93148
8	400	55	0.93095
9	500	70	0.93098
10	400	60	0.93097

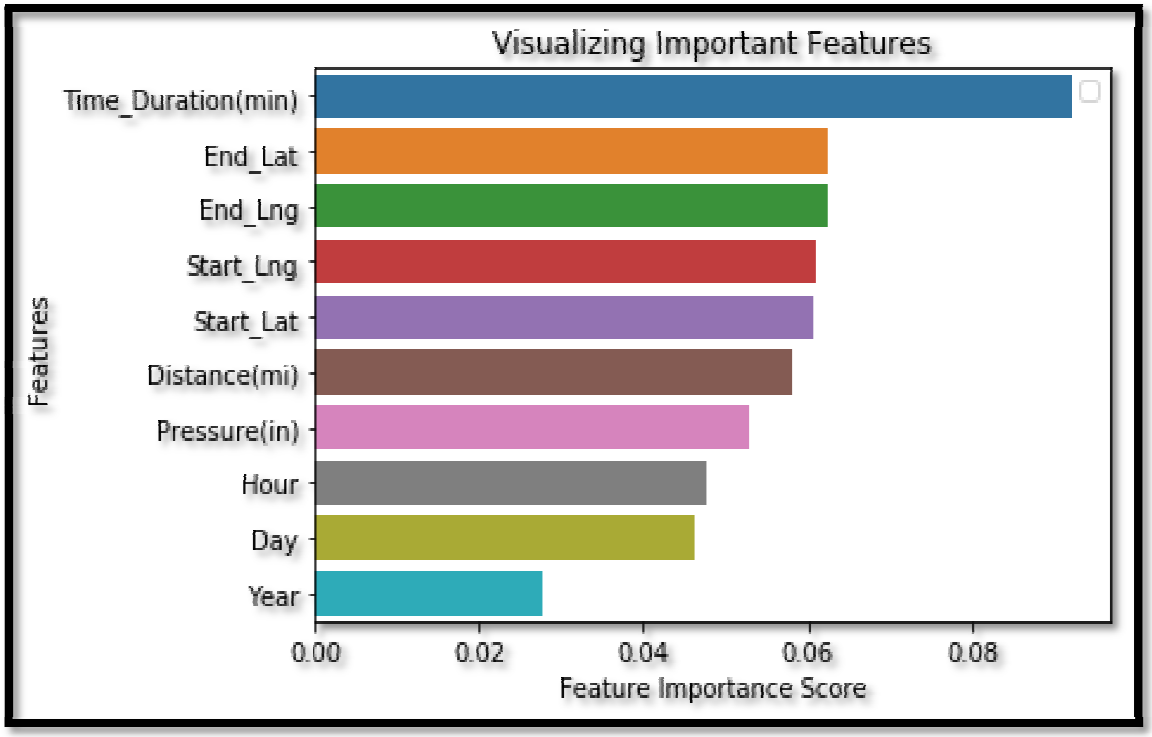
4.4.2.2 Building RF model

Following the selection of tuning parameters, Figure (4.12) in Appendix (B) illustrates the fundamental execution of the random forest classification model. Apart from the two tuning settings n etimatores = 500 and max depth = 55, there are no further configuration parameters.

As one of the specific features and a critical area of application for a random forest, the feature _importances Attribute can be used to extract variable importance scores. The mean and standard deviation of the accumulation of the impurity decrease within each tree can be used to calculate the feature relevance. The related variable significance scores are depicted in Figure (4.13)

Figure (4.13)

The Corresponding Of Variable Important Scores



4.4.3 Support Vector Machines (SVMs)

SVMs are based on statistical learning theory and are used to determine the optimal position of decision borders for class separation [10]. Unfortunately, the performance of the SVM is greatly dependent on the parameter settings and kernel choices. The quality of the SVM parameters and kernel functions that are selected has an effect on learning and generalization.

4.4.3.1 Selecting Tuning Parameters For SVM

Additionally, we performed grid search with cross-validation to update the SVM parameters. There are two critical parameters to optimize in the RBF kernel: C and

gamma. The C parameter adds a penalty for each misclassified data point; a modest C value results in a low penalty for misclassified points. When C is big, SVM attempts to reduce the number of incorrect classifications. While the gamma parameter is a hyperparameter that specifies the amount of curvature we want in the decision boundary, the line that separates the hyperparameter is practically linear for small values of gamma and becomes more curved for bigger values. Excessively increasing gamma may allow for overfitting on train data [23]. Polynomial kernels are defined by three parameters: C, gamma, and degree. The number of options is enormous even though there are actually more than three parameters, as there can be a large number of steps (or possible values).Figure (4.14) in Appendix (B) shows how we optimized these parameters using grid search (without degree in polynomial case).

Furthermore, we utilized GridSearchCV to optimize the kernel, gamma, and C parameters; we divided the available data into k subsets (k=10); and the fit () algorithm was running ten times independently to pick the table tuning settings with the greatest precision. The following Table contains all of the optimal solutions for the 10 instances of the fit () function (4.5) Kernel's ideal values are rbf, C = 100, and gamma=0.001; these values are repeated four times with a best score of 0.9031 for accuracy, and hence we chose these values as the final model parameters.

Table (4.5)

Fit () Function Optimal Results

K Number	Best Parameters			Best Score
	Kernel	C	Gamma	
1	rbf	100	0.01	0.8922
2	poly	100	0.001	0.8901
3	rbf	100	0.001	0.8942
4	rbf	10	0.001	0.9021
5	poly	100	0.001	0.8950
6	rbf	100	0.0001	0.9053
7	rbf	100	0.001	0.9031
8	rbf	100	0.01	0.9011
9	rbf	100	0.01	0.9012
10	rbf	100	0.001	0.8968

4.4.3.2 Building SVM Model

Following the tuning parameters selection process, Figure (4.15) in Appendix (B) shows how the SVM classification model is implemented. Among the three tuning parameters, kernel = rbf, C=100, and gamma = 0.001 are the only settings required.

4.5 Optimization Algorithms Performance

The performance of the models was report on the test set below:

- TR, FP, and TN: Table(4.6) These table summarizes the performance of models trained from a training set (including ANN, RF, SVM), the performance for each class is reported. In addition, the weighted average of all class values of the test set, divided by the number of instances in each class, is computed for the classifier.

Table (4.6)

TPR, FPR, And TNP Of The Three classifiers

Model	**ANN**			**RF**			**SVM**		
Metric	TPR	FPR	TNR	TPR	FPR	TNR	TPR	FPR	TNR
0	0.3306	0.0029	0.9970	0.5874	0.0023	0.9976	0.3007	0.0033	0.9966
1	0.9920	0.7941	0.2058	0.9904	0.5050	0.4949	0.9950	0.9007	0.09921
2	0.0991	0.0028	0.9971	0.2969	0.0026	0.9973	0.0634	0.0018	0.9981
3	0.1921	0.0044	0.9955	0.5833	0.0078	0.9921	0.0091	0.0001	0.9998
Weighted Average	0.90	0.706	0.29	0.93	0.449	0.55	0.894	0.80	0.199

- precision, F1 and recall scores: Table (4.7) shows their performance based on the training data, and report the performance for the classes, and Performance of the classifier based on the weighted average.

Table (4.7)

Results Of The Three Classifiers In Terms Of Precision, Recall, And F1

Model	**ANN**			**RF**			**SVM**		
Metric	***Prec.***	***Rec.***	***F1***	***Prec.***	***Rec.***	***F1***	***Prec.***	***Rec.***	***F1***
0	0.73	0.33	0.45	0.85	0.59	0.69	0.66	0.30	0.41
1	0.91	0.99	0.95	0.94	0.99	0.97	0.90	1.00	0.95
2	0.64	0.10	0.17	0.85	0.30	0.44	0.63	0.06	0.12
3	0.64	0.19	0.30	0.74	0.58	0.65	0.65	0.01	0.02
Weighted Average	0.88	0.90	0.87	0.93	0.93	0.92	0.87	0.90	0.86

- overall accuracy and Confusion matrix: Table (4.8), Table (4.9), and Table (4.10) In appendix (A) shows the confusion matrices of the models trained by training set. The number of correctly classified instances provides an overall measure of accuracy. A percentage and number of incorrectly classified instances are also reported.

Despite having the same meaning, TPR and recall are reported together because FPR is usually presented alongside TPR, whereas precision is frequently presented in conjunction with recall.

The TPR, FPR, and TNR of ANN, RF, and SVM algorithms are listed in Table (4.6). As expected, Table (4.6) demonstrates that the algorithms perform satisfactorily on a weighted average basis for TPR. Indeed, the TPR varies from 89 to 93 percent (SVM) (RF). However, by examining the performance of the classifiers for each class, we can determine that they are ineffective at predicting the two classes. Indeed, TPR spans from 6.3 to 29 percent (SVM to RF), while FPR ranges from 0.18 to 0.28 percent (SVM to RF) (ANN). Assessing a classifier's performance on a per-class basis appears to be the most appropriate and trustworthy way.

Using the training set, Table (4.7) reports the precision recall, and F1 score for the models trained on it. Table (4.7) includes an additional set of performance measurements that appear to corroborate the findings in Table (4.6) Indeed, we can expect good classifiers based on the weighted average precision, recall, and F1 score.

The confusion matrices and overall accuracy for ANN, RF, and SVM are presented in Tables (4.8), (4.9), and (4.10) above. Each confusion matrix has a bolded diagonal.

Because the confusion matrix compares the number of observed cases classified correctly versus wrongly for each class. It appears to be a reasonable performance metric for the purpose of introducing a classifier. By examining the three tables (4.8), (4.9), and (4.10), According to RF, 467 of 795 cases of the negative class can be correctly predicted (69%).

SVM ranked highest for predicting 1 class (32651 out of 32695 correctly identified instances), while RF ranked highest for predicting 2 class (503 out of 1785 correctly identified instances), and the best classifier for predicting 3 class is RF also (805 out of 1447 correctly detected). We evaluated our three-model using 10-Fold the result is showed in Table 11-13.

Table (4.11)

ANN model results using K-fold cross validation

Fold	Precision	Recall	*F1*-Score	Accuracy
1	0.88	0.90	0.87	90.16
2	0.88	0.90	0.87	90.17
3	0.89	0.91	0.88	90.47
4	0.88	0.90	0.87	90.15
5	0.88	0.90	0.87	89.90
6	0.88	0.90	0.87	90.25
7	0.89	0.91	0.88	90.49
8	0.89	0.91	0.88	90.54
9	0.88	0.90	0.87	90.30
10	0.88	0.90	0.87	90.41
Average	**0.883**	**0.903**	**0.873**	**90.284**

Table (4.12)

RF model results using K-fold cross validation

old	Precision	Recall	*F1*-Score	Accuracy
1	0.92	0.93	0.91	92.78
2	0.92	0.92	0.91	92.48
3	0.92	0.93	0.91	92.82
4	0.92	0.93	0.91	92.69
5	0.93	0.93	0.92	92.90
6	0.92	0.93	0.91	92.68
7	0.92	0.93	0.91	92.73
8	0.92	0.93	0.91	92.76
9	0.92	0.92	0.91	92.42
10	0.92	0.92	0.91	92.45
Average	**0.921**	**0.927**	**0.911**	**92.671**

Table (4.13)

SVM model results using K-fold cross validation

Fold	Precision	Recall	*F1*-Score	Accuracy
1	0.79	0.89	0.84	89.14
2	0.79	0.89	0.84	89.15
3	0.80	0.89	0.84	89.18
4	0.80	0.89	0.84	89.23
5	0.79	0.89	0.84	88.97
6	0.79	0.89	0.84	89.14
7	0.79	0.89	0.84	89.03
8	0.79	0.89	0.84	88.81
9	0.79	0.89	0.84	89.15
10	0.79	0.89	0.84	89.12
Average	**0.792**	**0.89**	**0.84**	**89.092**

Considering the overall average accuracy in the three tables above, RF seems the best classifier (92.671%), followed by ANN (90.284%), and SVM (89.74%).

According to our previous discussion, the prediction accuracy was 90.18 for the testing set, 90.02 for the validation set and approximately 90.4 for the training set based on Figure (4.16) in appendix (B), which shows the accuracy of the model over the first 800 epochs using the Neural Network method.

As shown in Figure (4.17) in appendix (B) for the first 800 epochs, the model loss is 0.317 for the training set and 0.327 for the validation set.

Chapter Five
Conclusion

By employing machine learning algorithms, the thesis attempted to predict traffic accident injury severity.

At first, it was found that a significant number of recent important papers relying on machine learning algorithms in order to predict crash severity were published in the literature. These papers used a variety of datasets from various countries in order to apply machine learning algorithms to crash severity predictions.

Machine learning algorithms have demonstrated a high degree of success in extracting knowledge from provided data. Unfortunately, their success is directly related to the data quality with which they work. If the data is insufficient or contains irrelevant and unnecessary information, machine learning algorithms may fail to uncover anything useful or may generate results that are less accurate and intelligible. Thus, data pre-processing is a critical phase in the machine learning process; in our thesis, the pre-processing step includes dealing with missing values, dealing with categorical features, and scaling features.

The available dataset was split randomly into a training and test set. Three algorithms trained on the training set (ANN, RF, and SVM), before building MLs model we choose to tuning parameter using grid search, we compared three algorithms using a similar dataset size, which made it possible to compare their results.

For each algorithm, a wide range of performance metrics were computed, including the confusion matrix, TPR, FPR, TPN, recall, precision, and overall accuracy. As far as describing the predictive capacities of a classifier is concerned, the confusion matrix appears to be the best and most suitable representation.

Considering the overall accuracy, RF classifier was outperformed with (93.45%), followed by ANN (90.18%), and SVM (89.74%).

Future Work:

1. Collection more crash dataset in Palestine.

2. Use similar algorithms for classification real crash data in Palestine.

3. Implementation more advanced machine learning in the same dataset.

References

[1] who, "Road traffic injuries," 2021. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries.

[2] wafa.ps, "2020 ",الحـــــــــــــوادث المروريـــــــــــــة. [Online]. Available: (http://info.wafa.ps/ar_page.aspx?id=8209).

[3] Y. Sarraj, "Developing Road Accidents Recording System in Palestine," 2016.

[4] "Road Traffic Accidents in Palestine by Governorate and Month," 2020. [Online]. Available: https://www.pcbs.gov.ps/statisticsIndicatorsTables.aspx?lang=en&table_id=788.

[5] "US-Accidents: A Countrywide Traffic Accident Dataset," [Online]. Available: https://smoosavi.org/datasets/us_accidents.

[6] S. A. &. S. T. Madhar Taamneh, "Data-mining techniques for traffic accident modeling and prediction in the United Arab Emirates," *Journal of Transportation Safety & Security,* 2016.

[7] X. Z. T. Y. J. T. a. R.-. c. M. Zhuoning Yuan, "Predicting Traffic Accidents Through Heterogeneous Urban Data: A Case Study," 2017.

[8] Q. H. J. G. M. N. Zhenhua Zhanga, "A deep learning approach for detecting traffic accidents from social media data," *elsevier,* 2018.

[9] D. K. a. P. E. P. S. B. Kotsiantis, "Data Preprocessing for Supervised Leaning," *INTERNATIONAL JOURNAL OF COMPUTER SCIENCE,* vol. 1, 2006.

[10] W. S. Noble, "What is a support vector machine?," *NATURE BIOTECHNOLOGY,* vol. 24, 2006.

[11] L. Breiman, "RANDOM FORESTS--RANDOM FEATURES," 1999.

[12] F. CHOLLET, Deep Learning with Python.

[13] A. A. &. Paprzycki, "Traffic Accident Analysis Using Machine Learning Paradigms.," *ResearchGate,* 2005.

[14] "National Center for Statistics and Analysis http://www-nrd.nhtsa.dot.gov/departments/nrd- 30/ncsa/NASS.html," *Cornell Univerdity Library.*

[15] A. S. Y. M. G. J. P. Mehdi Hosseinpour, "Application of Adaptive Neuro-fuzzy Inference System for road accident prediction," *ResearchGate,* 2013.

[16] Q. W. A. W. S. Lei Lin, "A novel variable selection method based on frequent pattern tree for real-time traffic accident risk prediction," *elsevier,* 2015.

[17] *. M. A.-A. Q. S. J. P. Ling Wanga, "Real-time Crash Prediction for Expressway Weaving SegmentsR," *ResearchGate,* 2015.

[18] S. A. a. S. T. Madhar Taamneha, "Data-mining techniques for traffic accident modeling and prediction in the United Arab Emirates," *JOURNAL OF TRANSPORTATION SAFETY & SECURITY,* 2016.

[19] a. Y. C. Miaomiao Liu, "Predicting Real-Time Crash Risk for Urban Expressways in China," *Hindawi,* 2017.

[20] Z. Zhang, Q. He, J. Gao and M. Ni, "A deep learning approach for detecting traffic accidents from social media data," *ELSEVIER,* 30 January 2017.

[21] M. I. Sameen, "Severity Prediction of Traffic Accidents with Recurrent Neural Networks," *MDPI,* 8 June 2017.

[22] Z. Yuan, X. Zhou and T. Yang, "Predicting Tra c Accidents Through Heterogeneous Urban Data: A Case Study," *doi,* august 2017.

[23] H. Ren, Y. Song, J. Wang, Y. Hu and J. Lei, "A Deep Learning Approach to the Citywide Traffic Accident Risk Prediction," *Cornell University Library,* 15 April

2018.

[24] Z. L. ,. Z. P. A. C. X. JIAN ZHANG, "Comparing Prediction Performance for Crash Injury Severity Among Various Machine Learning and Statistical Methods," *IEEE,* 2020.

[25] B. G. a. M. A. Natalia Casado-Sanz, "Analysis of the Risk Factors Affecting the Severity of Traffic Accidents on Spanish Crosstown Roads: The Driver's Perspective," *MDPI,* 2020.

[26] H. H. a. S. A.-H. Ali J. Ghandour, "Analyzing Factors Associated with Fatal Road Crashes: A Machine Learning Approach," *MPDI,* 2020.

[27] N. F. a. M. Losa, "Handling Imbalanced Data in Road Crash Severity Prediction by Machine Learning Algorithms," *MPDI,* 2020.

[28] A. Géron, Hands-On Machine Learning with Scikit-Learn and TensorFlow, 2017.

[29] G. E. B. a. M.-C. Monard, "An Analysis of Four Missing Data Treatment Methods for Supervised Learning.," *ResearchGate,* 2003.

[30] D. Hendrickson, "Missing Data Imputation: Predicting Missing Values," 2018.

[31] http://shop.oreilly.com/product/0636920052289.do.

[32] (Kotsiantis, supervised Machine Learning : A Review of Classification Techniques, 2007.

[33] A. K. a. J. Mao, "Artificial Neural Networks: A Tutorial," 1996.

[34] D. STATHAKIS, "How many hidden layers and nodes?," *International Journal of Remote Sensing,* 2008.

[35] L. A. a. D. W.Aha, "Survey, Simplifying Decision Trees: A".

[36] □. L. J. B. F. B. R. P. Franco Bassoa, "Real-time crash prediction in an urban

expressway using disaggregated data," *elsevier,* 2018.

[37] J. H. F. R. A. O. C. J. S. Leo Breiman, Classification And Regression Trees, 1984.

[38] A. G. f. D. Scientists, Introduction to Machine Learning with Python, 2017.

[39] E. Z. R. M. a. A. K. Petre Lameski1 ⋆, "SVM Parameter Tuning with Grid Search and its Impact on Reduction of Model Over-fitting," *Academia.*

[40] [Online]. Available: (https://www.who.int/news-room/detail/07-12-2018-new-who-report-highlights-insufficient-progress-to-tackle-lack-of-safety-on-the-world's-roads). [Accessed 6 2019].

[41] [Online]. Available: (http://info.wafa.ps/ar_page.aspx?id=8209). [Accessed 3 2019].

[42] [Online]. Available: https://smoosavi.org/datasets/us_accidents.

[43] A. Géron, Hands-On Machine Learning with Scikit-Learn & TensorFlow.

[44] N. Shukla, Machine Learning with TensorFlow.

[45] A. Abraham and P. Paprzycki, "Traffic Accident Analysis Using Machine Learning Paradigms," *ReserchерGate,* 2005.

[46] M. Hosseinpour, S. M. Ghadiri, A. S. Yahaya and J. Prasetijo, "Application of Adaptive Neuro-fuzzy Inference System for road accident prediction," *ResearchGate,* November 2013.

[47] L. Lin, Q. Wang and A. W. Sadek, "A novel variable selection method based on Frequent Pattern tree for real- time traffic accident risk prediction," *ResearchGate,* March 2015.

[48] L. Wang, M. Abdel-Aty, Q. Shi and J. Park, "Real-time Crash Prediction for Expressway Weaving Segments," *ResearchGate,* December 2015.

[49] M. Taamneh, S. Alkheder and S. Taamneh, "Data-mining techniques for traffic accident modeling and prediction in the United Arab Emirates," *Transportaion Safty&Security,* 27 Apr 2016.

[50] M. Liu and Y. Chen, "Predicting Real-Time Crash Risk for Urban Expressways in China," *Hindawi,* 30 November 2016.

[51] www.manning.com, www.manning.com: Manning Publications Co..

[52] https://www.datasciencecentral.com/m/blogpost?id=6448529:BlogPost:410294, 12.

[53] [Online]. Available: "New WHO report highlights insufficient progress to tackle lack of safety on the world's roads." https://www.who.int/news/item/07-12-2018-new-who-report-highlights-insufficient-progress-to-tackle-lack-of-safety-on-the-world%27s-roads (accessed Oct. 13, 20.

[54] Y. S. ,. J. W. ,. Y. H. ,. a. J. L. Honglei Ren*, "A Deep Learning Approach to the Citywide Traffic Accident Risk Prediction," 2018.

[55] H. H. a. S. A.-H. Ali J. Ghandour, "Analyzing Factors Associated with Fatal Road Crashes: A Machine Learning Approach," *MDPI,* 2020.

Appendices

Appendix (A) Table

Table (1.1)

Road Traffic Accidents In Palestine By Governorate And Month,2020 [5].

Governorate	***January***	***February***	***March***	***April***	***May***	***June***	***July***	***August***	***September***	***October***	***November***	***December***	***Total***
Jenin	100	137	87	68	123	156	120	146	137	157	131	111	1473
Tubas	25	22	18	12	28	32	19	33	25	19	28	19	280
Tulkarm	55	53	52	33	37	45	48	73	58	59	58	48	619
Nablus	170	235	189	153	188	194	186	211	242	283	217	148	2416
Qalqiliya	19	28	20	12	17	26	16	23	28	31	16	20	256
Salfit	36	30	30	11	30	25	36	51	35	50	35	39	408
Ramallah	252	252	198	101	189	232	151	231	217	231	217	221	2492
Jericho	32	37	20	16	24	46	33	31	41	56	36	32	404
Jerusalem	43	55	32	18	42	49	48	42	41	52	40	44	506
Bethlehem	73	75	25	37	43	63	45	63	67	70	57	52	670
Hebron	112	79	70	64	141	119	102	157	159	190	133	127	1453
Total	917	1003	741	525	862	987	804	1061	1050	1198	968	861	10977

Table (3.2)

A Countrywide Traffic Accident Dataset Of United States[6]

#	Attribute	Data Type	Description
1	ID.Number	Numeric	A unique number for the accident log
2	Num	Numeric	Displays the street number in address field.
3	S.Time	Numeric	When the accident started
4	S.Latitude	Numeric	The latitude is shown in the GPS coordinates of the starting point
5	S.Lngitude	Numeric	The longitude is shown in the GPS coordinates of the starting point
6	E.Time	Numeric	End time of the accident. End time refers to when the impact of accident on traffic flow was dismissed.
7	End.Lat	Numeric	The latitude is shown in the GPS coordinates of the end point
8	Accident. Severity	Numeric	a number which has four values ranging between 1 and 4, this number shows the severity of the accident, where 1 means the least impact on traffic (short delay after of the accident) and 4 means a significant impact on traffic (long delay).
9	End.Lng	Numeric	The longitude is shown in the GPS coordinates of the end point
10	Description	Nominal	Displays description of the accident.
11	Side	Nominal	Displays the relative side of the street (Right/Left)
12	County_ Name	Nominal	Displays the county name
13	Time.zone	Nominal	Based on where the accident occurred (central, eastern, etc.).
14	Distance	Numeric	Displays the section of the road that affected by the accident.
15	Weather.Tim estamp	Numeric	Displays what time the weather station saw something and records.
16	Street_Name	Nominal	Displays the street name
17	Humidity	Numeric	Displays Humidity.
18	City_Name	Nominal	Displays the city name
19	Precipitation	Numeric	Displays the amount of precipitation, if any.
20	State.Name	Nominal	Displays the state name
21	Zipcode	Numeric	Displays the zipcode
22	Visibility	Numeric	Vision appears
23	Airport.Code	Nominal	This shows the closest weather station to where

#	Attribute	Data Type	Description
			the accident happened
24	Railway	Nominal	Annotation POI indicating a railway nearby.
25	Temp (F)	Numeric	Displays thetemperature.
26	Wind.Chill (F)	Numeric	Displays coolness of the wind.
27	Turning.Loop	Nominal	Annotation POI indicating a turning loop nearby.
28	Pressure	Numeric	Displays air pressure.
29	Give.Way	Nominal	Annotation POI indicating a give way nearby.
30	Wind. Direction	Numeric	Displays the direction of the wind.
31	Wind.Speed (mph)	Numeric	Displays the speed of the wind.
32	Bump	Nominal	Annotation POI indicating a bump or hump nearby.
33	Weather.Condition	Nominal	Displays the weather condition (snow, rain, fog, thunderstorm, etc.)
34	Amenity	Nominal	Annotation POI indicating the presence of amenities nearby
35	Traffic.Signal	Nominal	Annotation POI which indicating a Traffic_Signal nearby.
36	Crossing	Nominal	Annotation POI indicating a crossing nearby.
37	Sunrise. Sunset	Nominal	Displays the period of day (i.e., day or night) based on sunrise and sunset.
38	Junction	Nominal	Annotation POI indicating a junction nearby.
39	No.Exit	Nominal	Annotation POI indicating a no exit nearby.
40	Civil. Twilight	Nominal	Displays the period of day (i.e., day or night) depending on civil twilight.
41	Roundabout	Nominal	Annotation indicating a roundabout nearby.
42	Station	Nominal	Annotation POI indicating a station nearby.
43	Stop	Nominal	Annotation POI indicating a stop nearby.
44	Traffic. Calming	Nominal	Annotation POI indicating a trafficcalming nearby.
45	Nautical. Twilight	Nominal	Displays the period of day (i.e., day or night) depending on nautical twilight.
46	Astronomical .Twilight	Nominal	Displays the period of day (i.e., day or night) depending on astronomical twilight.

Table (4.3)

Average Accuracy For Test Data

Neurons	Average accuracy
5	0.90024
10	0.90344
20	0.90091
30	0.9018
40	0.90188
50	0.9029
60	0.90323

Table (4.8)

Confusion Matric Of The Classifier ANN

		ANN			
True Label	0	284	564	9	2
	1	86	32436	68	105
	2	4	1555	177	49
	3	17	1130	22	278
		0	1	2	3

Predicted Label

Correctly Classified Instances:33175 (90.18%)
Incorrectly Classified Instances: 3611 (9.81%)

Table (4.9)

Confusion Matric Of The Classifier RF

		RF			
True Label	**0**	**467**	319	8	1
	1	55	**32603**	64	195
	2	20	1088	**503**	83
	3	8	547	20	**805**
		0	**1**	**2**	**3**

Predicted Label

Correctly Classified Instances:34378 (93.45%)
Incorrectly Classified Instances: 2408 (6.55%)

Table (4.10)

Confusion Matric Of The Classifier SVM

		SVM			
True Label	0	240	554	4	0
	1	105	32651	54	5
	2	3	1633	111	2
	3	13	1390	8	13
		0	1	2	3
		Predicted Label			
		Correctly Classified Instances:33015 (89.74%)			
		Incorrectly Classified Instances: 3611 (10.25%)			

Appendix (B) Figure

Figure (3.1)

Day of week

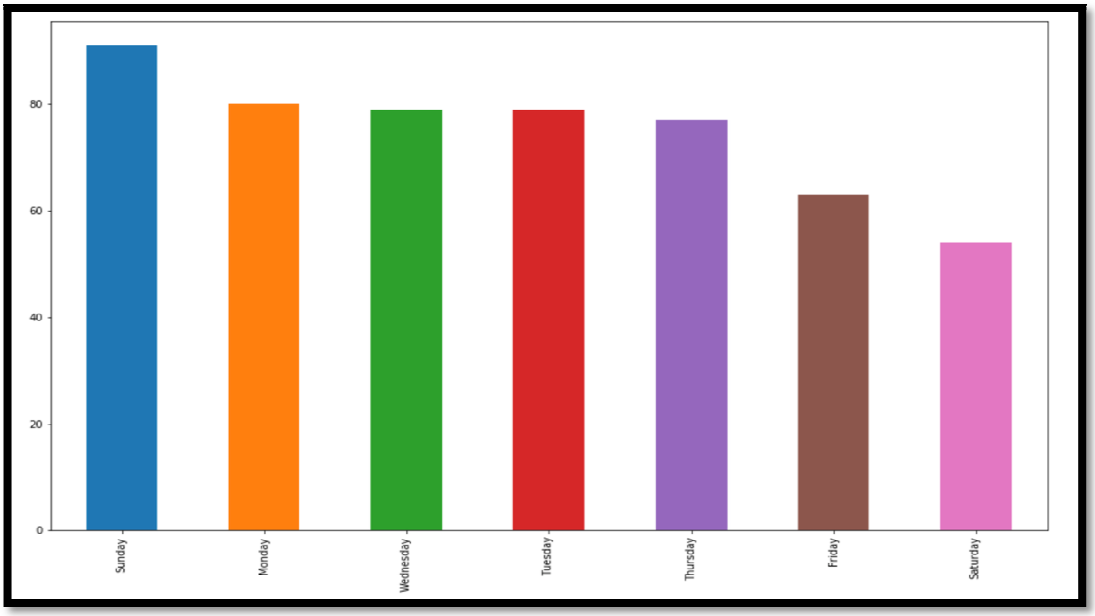


Figure (3.2)

Month of year

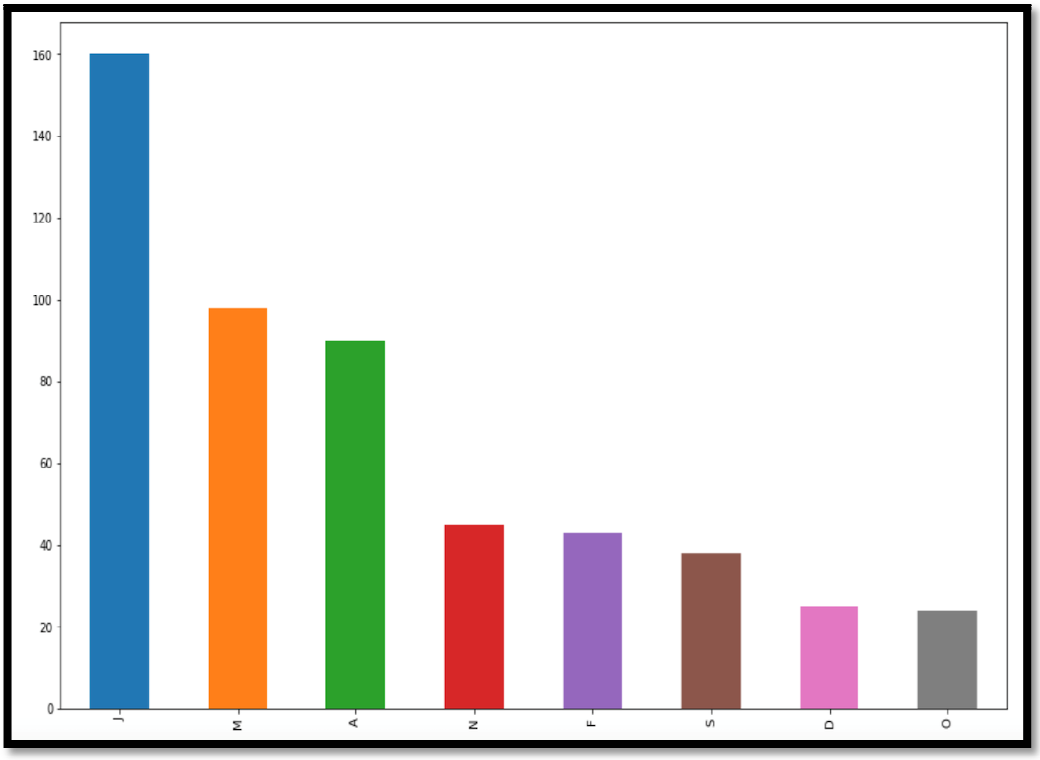


Figure (3.3)

Gender of driver

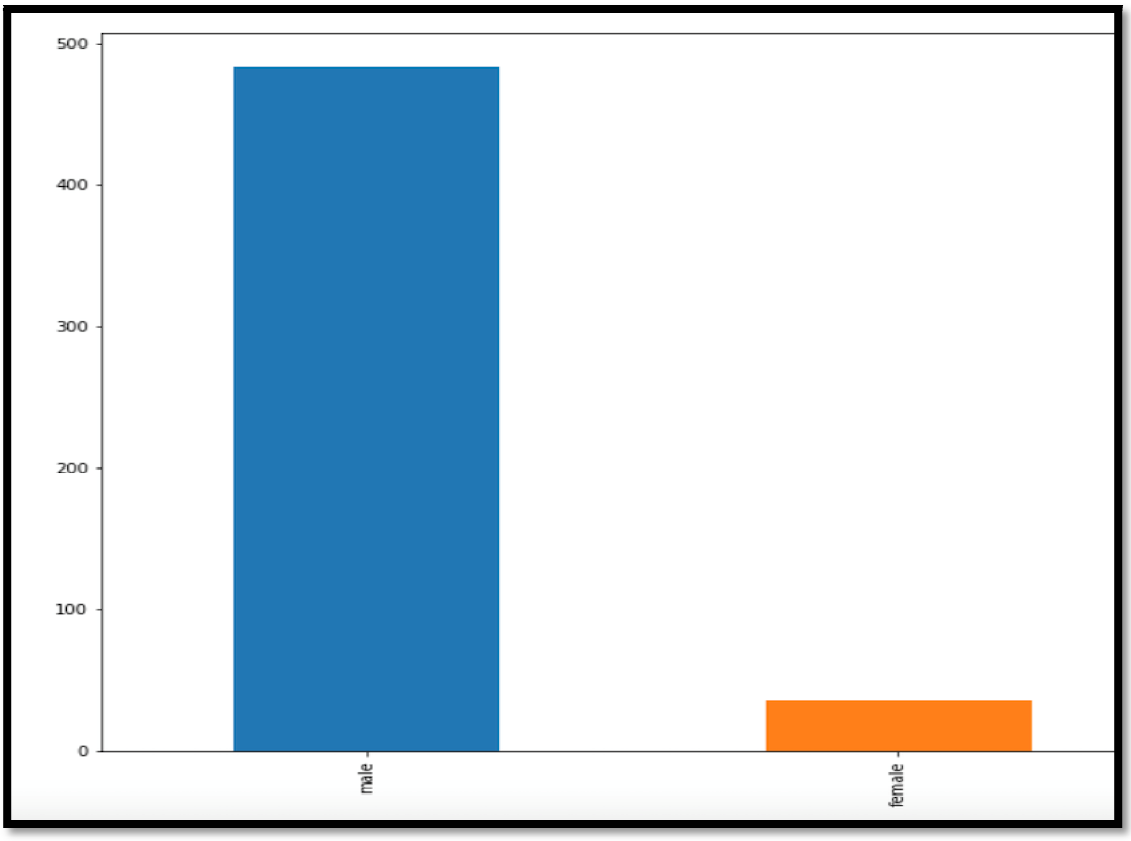


Figure (3.6)

A Feed-forward Artificial Neural Network, which only allows signals to travel from input to output[33].

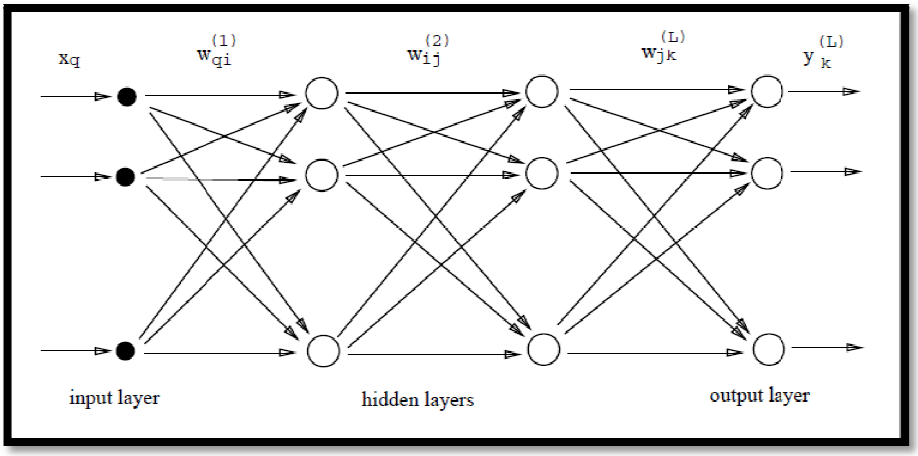


Figure (4.7)

Confusion Matrix

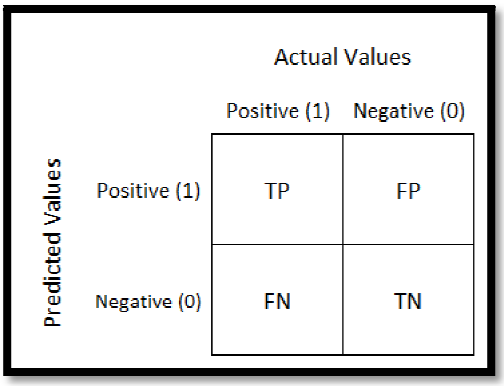


Figure (4.8)

Summary Of The ANN Model

```
model_dense_5.summary()
```

```
Model: "sequential_16"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_31 (Dense)             (None, 10)                890
_________________________________________________________________
dense_32 (Dense)             (None, 4)                 44
=================================================================
Total params: 934
Trainable params: 934
Non-trainable params: 0
_________________________________________________________________
```

Figure (4.9)

Grid Search To Optimize ANN Parameters

```
from keras.wrappers.scikit_learn import KerasClassifier
from sklearn.model_selection import GridSearchCV
KC = KerasClassifier(build_fn=dense_model_5)
parameters = {'batch_size' : [8,16,32,64],
              'optimizer':['SGD','Adam','Nadam']}
grid_search = GridSearchCV(estimator=KC ,
param_grid=parameters,cv=5)
grid_search.fit(X_train,y_train,epochs=100)
```

Figure (4.10)

Artificial Neural Network Model

```
def dense_model_5():
    opt=Adam(lr=0.01)

    model = Sequential()

    model.add(Dense(10,activation='relu',kernel_initializer='random_uniform',
                    kernel_regularizer=regularizers.l2(0.0001),input_dim=88) )


    model.add(Dense(4,activation='softmax'))

    model.compile(loss='categorical_crossentropy', optimizer=opt, metrics=['accuracy'])
    return model
```

Figure (4.11)

Grid Search To Optimize RF Parameters

```
from sklearn.model_selection import GridSearchCV
trees= RandomForestClassifier()
parameters = {'max_depth' : [30,40,55,60,70],
              'n_estimators' : [70,100,120,200,400,500]}
grid_search = GridSearchCV(estimator=trees ,
param_grid=parameters,scoring='accuracy',cv=10)
grid_search.fit(X_train,y_train)
```

Figure (4.12)

Random Forest Model

```
trees= RandomForestClassifier(max_depth=55, n_estimators=500, random_state=0)
trees.fit(X_train, y_train)
```

Figure (4.14)

Grid Search To Optimize SVM Parameters

```
# Set the parameters by cross-validation
tuned_parameters = {'C': [0.1,1, 10, 100],
                    'gamma': [1,0.1,0.01,0.001],
                    'kernel': ['rbf', 'poly']}
clf = GridSearchCV(SVC(),tuned_parameters,scoring='accuracy',cv=10)
clf.fit(X_train, y_train)
```

Figure (4.15)

SVM Model

```
from sklearn.svm import SVC
clf = SVC(kernel='rbf', gamma=1e-3,C=100)
clf.fit(X_train, y_train)
clf.fit(X_train,y_train)
```

Figure (4.16)

ANN Model Accuracy

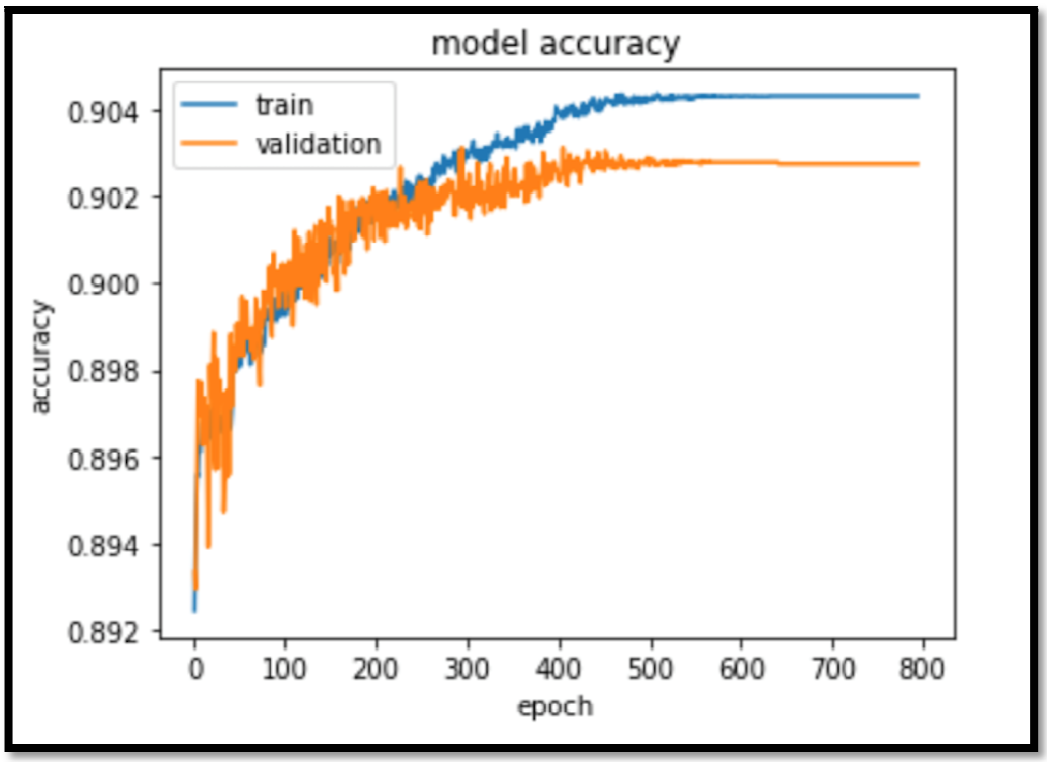
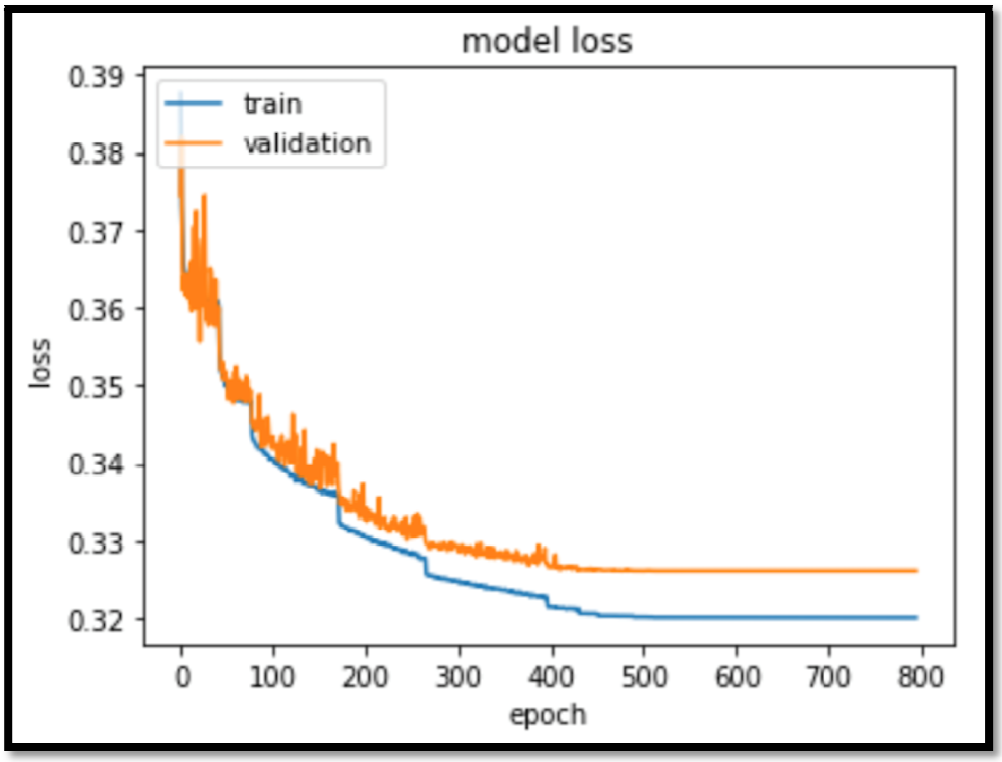


Figure (4.17)

ANN Model Loss



جامعة النَّجاح الوطنيَّة
كليَّة الدّراسات العليا

التنبؤ بمخاطر الحوادث المرورية باستخدام التعلم الآلي

إعداد
أماني محمد حكواتي

إشراف
د. عدنان سلمان

قدمت هذه الرسالة استكمالاً لمتطلبات الحصول على درجة الماجستير في الحوسبة المتقدمة بكلية الدراسات العليا في جامعة النجاح الوطنية في نابلس، فلسطين.
2022م

التنبؤ بمخاطر الحوادث المرورية باستخدام التعلم الآلي

إعداد

أماني محمد حكواتي

إشراف

د. عدنان سلمان

الملخص

المقدمة: تشير حوادث المرور إلى الازدحام والتأخير والخسائر الاقتصادية والاعاقة واحيانا فقدان الحياة البشرية. هناك العديد من العوامل التي تؤثر على احتمالية وقوع الحادث وشدته. تتضمن العوامل المتعلقة بالسائق، والتضاريس والعوامل المتعلقة بالطرق، والعوامل المتعلقة بالطقس، وغيرها من العوامل المتعلقة بالحوادث. إن التنبؤ بخطورة حوادث الطرق وفهم العوامل التي تسببها أهداف بحثية مثيرة للاهتمام في مجال السلامة المرورية.

تحلل هذه الرسالة العديد من حوادث المرور بعمق وتحدد شدة الحوادث باستخدام تقنيات التعلم الالي. وتبين أيضا العوامل المهمة التي لها تأثير مباشر على خطورة حادث مروري. يمكن أن تساعد هذه المعرفة المدربين على تثقيف السائقين الجدد بشكل أفضل لتجنب الحوادث المرورية ويمكن أن تساعد صانعو القرار في فرض قوانين جديدة تساعد في تقليل شدة خطورة هذه الحوادث.

المنهجية: تم إجراء التحليل باستخدام آلية المتجهات الداعمة (SVM)، والشبكات العصبية الاصطناعية (ANN)، والغابات العشوائية (RF). قبل تطبيق تقنيات التعلم الالي، تمت معالجة بيانات حوادث المرور مسبقا من خلال ثلاث مراحل وهي : معالجة البيانات المفقودة، والتعامل من النص والمتغيرات الفئوية، وميزة التحجيم. تم استخدام التحقق من المقاطع (cross-validation) مع 10 مقاطع لتقييم أداء تقنيات التعلم الالي الثلاثة.

النتائج: في هذه الاطروحة، قمنا بتصنيف شدة الحادث إلى أربع فئات بناءً على التأخير الزمني بعد وقوع الحادث، أيضا تشير النتائج الى ان اهم العوامل التي لها تأثير مباشر على خطورة الحادث هي المدة الزمنية ونهاية خط الطول ونهاية خط العرض وبداية خط الطول وبداية خط العرض والمسافة (ميل).

الاستنتاج: بالنظر الى الدقة الكلية، تفَوَّق منهج الغابات العشوائية بنسبة (93.45٪) يليه الشبكات العصبية الاصطناعية بنسبة (90.18٪)، ثم المتجهات الداعمة بنسبة (89.74٪).

الكلمات المفتاحية: حوادث الطرق؛ الية المتجهات الداعمة؛ الشبكات العصبية الاصطناعية؛ الغابات العشوائية.