THE PENNSYLVANIA STATE UNIVERSITY
SCHREYER HONORS COLLEGE

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

ROAD TRAFFIC CRASH SEVERITY PREDICTION USING MULTI-STATE DATA

THOMAS ENGLAND
SPRING 2021

A thesis
submitted in partial fulfillment
of the requirements
for a baccalaureate degree
in Computer Engineering
with honors in Computer Engineering

Reviewed and approved* by the following:

Kamesh Madduri
Associate Professor of Computer Science and Engineering
Thesis Supervisor

John Sampson
Associate Professor of Computer Science and Engineering
Honors Adviser

*Electronic approvals are on file

Abstract

The socioeconomic burden of road traffic crashes is immense. Safer roads and vehicular mechanisms to reduce distracted driving help reduce collisions. Additionally, computational models can be used to understand the reasons for crashes and devise interventions. We study models predicting the severity of a crash based on the data reported at the crash scene. Many U.S. states have developed traffic safety programs to make the anonymized crash data publicly available. These datasets aid researchers in the creation of predictive models for crashes.

While many states make data from collisions publicly available, each state reports data differently. There is a lack of standardization. As a result, it is difficult for researchers to develop machine learning algorithms to process data from multiple states without adequate preprocessing. Currently, the vast majority of projects in this field of study utilize a dataset of a single city, road, or state. This limits the use of the developed model to a region. This project aims to create a large crash database that will allow researchers to develop algorithms that utilize data from across the country. Additionally, we want to examine if the use of data from multiple states is effective in increasing the accuracy of machine learning models.

In order to achieve these goals, we develop software to find common data categories from state reports and combine them into one large dataset. The data categories were selected based on reports from previous projects that identified variables having a large impact on model accuracy. In order to test the effectiveness of the new multi-state dataset, we used two models (neural network-based and decision tree-based) to predict crash injury severity. We trained and tested these models on datasets from a single state, combined two-state datasets, and a combined multi-state dataset.

The results of this research reveal that there is a drop in accuracy when data from multiple states are combined. This trend is present in both the models tested, with the trend being more pronounced in the decision tree. There are some cases in the neural network model where multi-state data lead to a higher accuracy compared to the single-state experiments. We also observe a downward trend

between neural network accuracy and the distance between the states present in the dataset. This implies that the closer the states are together geographically, the better the accuracy will be using the neural network model. In the decision tree model, there is a positive correlation between overall accuracy and the number of features present in the dataset. This observation means that the more features states have in common, the better the accuracy will be for a decision tree classifier. The software artifacts from this project are open-sourced.

Table of Contents

List of Figures

List of Tables

Acknowledgements

I would like to express my deepest appreciation to Dr. Kamesh Madduri. Dr. Madduri has been an incredible teacher and mentor throughout this entire process. Without his guidance and help I would not be able to have done what I have and I am deeply grateful for all of his help. I owe my ability to do this project at all to Dr. Madduri because he was generous enough to offer me the opportunity to do this research.

I am also thankful to my mom and dad for always encouraging me to be the best I can be. Throughout my academic journey my parents have always been my support system. My mom has always been willing to talk me through any issues I have to make sure I am able to perform the best work that I can. My dad not only wanted to aid me in this endeavor, but actively sought out the opportunity to help however he could in this process. I wouldn't be the person I am today without the support of my family.

Dr. John Sampson has also been a huge help in my college career as well as the thesis process. Dr. Sampson has guided me through college as my advisor and helped me in the review of my thesis as well. I am grateful for all he has done for me over the years.

Chapter 1

Introduction

There has been a steady increase in car crashes since the year 2000. Not only does this increase in collisions result in approximately 1.35 million fatalities globally, but countries spend around 3% of their Global Domestic Product (GDP) every year to account for the costs caused by these collisions. The World Health Organization (WHO) has set a goal to halve the number of collisions in the world by the year 2030 [1]. One method used to combat the number of vehicular collisions in recent years is the use of artificial intelligence and machine learning. Machine learning models have the ability to predict the severity of a car crash given the data reported at the time of the collision. Crash severity prediction would allow researchers to identify factors causing fatal crashes, and such research outcomes can guide policymakers in developing interventions from reducing crash incidence. This project will focus on improvements to severity prediction for use in the United States (US). We chose the US because crash data are available at the level of a state, and federal agencies also provide historical datasets. Additionally, there is no charge to access anonymized road traffic collision data for research purposes.

1.1 Crash Severity Prediction

Crash injury severity is a major area of focus in the research field of road traffic safety. A crash severity prediction program can classify the severity rating of existing crashes and predict the severity outcome of hypothetical crash scenarios. Currently, the vast majority of studies limit the scope of their data to a single city, state, or road over the course of multiple years. As a result, the models produced in prior research are specific to the location of the dataset they were trained on. This project will focus on the creation of a multi-state dataset with the purpose of examining the effectiveness of a broader range of data on the accuracy of a machine learning model. The main

idea behind this research is that an increase in the amount of data given to a machine learning model will result in a higher level of accuracy for severity predictions. In addition, a model that is able to predict crashes in a broader geographic range can provide an insight into the causes of crashes on a larger scale. For instance, instead of needing multiple models to see the reasoning behind a crash in California versus a crash in New Jersey, a single model can be used over the entire United States. This provides uniformity in a field that is very divided when it comes to data reporting.

There are many applications for an accurate crash severity prediction model. One of the major uses for the model is to decrease the number of severe crashes throughout the United States. A machine learning model that classifies crashes based on severity can also detail which factors are most prevalent in severe crashes. Traffic researchers can then use the data to make changes that would theoretically prevent severe crashes in the future. For instance, if the model finds that severe crashes occur most often due to speeding on three-lane roads, a local government could station law enforcement officers on three-lane highways to prevent speeding and therefore prevent crashes. A second application of this research would be for uses in emergency services dispatch. If the model predicts that a crash is severe based on the data reported, then the proper amount of emergency services could be dispatched to the location in an attempt to deal with the situation most efficiently. In general, a model such as this one provides data that can be crucial for the safety of drivers on the road. In addition, the prevention of crashes will result in less spending by governments across the globe. By reducing the 3% of the GDP spent on car crashes per year, countries will be able to spend money on other important issues in their nation. Finally, car crash severity prediction could help the WHO with its goal to halve the number of car crashes by 2030.

Chapter 2

Machine Learning Models

This chapter introduces the machine learning models used in this project and reviews prior work.

2.1 Neural Networks

Neural networks are a series of nodes called neurons that are organized in layers. The network of neurons finds patterns in data that are typically not distinguishable by the human eye. The patterns in the data are found through a system of weights and biases. The neurons connect to each other to form a network. Based on the input, a non-linear activation function, and the bias at each neuron, the node will create a new output that will be pushed forward in the network. By the time the new values reach an output layer, the values are compared to the actual results of the data in a process known as training. The error between the generated output and the correct output is used to update all of the biases and weights in the network through a process called backpropagation. By the end of the training phase, the goal is that the network will be able to correctly predict outputs for data it has never seen before [2].

A Recurrent Neural Network (RNN) is a type of neural network that includes feedback connections. The feedback connections allow for the network to have a memory of past computations. Due to this memory, a RNN is computationally more powerful than a normal neural network classifier. However, RNNs are known to have an issue called the long-term dependency problem. This problem is described as being the trade-off that must occur between efficient learning and holding onto information for long periods of time [3]. The specific layer used in the RNN in this research is called long short-term memory (LSTM) which was originally proposed by Hochreiter and Schmidhuber [4]. Unlike normal RNNs, LSTM networks are able to remember information from cases that were done much earlier in the training period. LSTMs are considered to be the

solution to the long-term dependency problem. This solution is a good fit for this project because the data will span long time periods and therefore might suffer from the long-term dependency issue.

The neural network in this project consists of an embedding Layer, an LSTM recurrent layer, two hidden layers of 32 nodes, and an output layer. A Recurrent Neural Network (RNN) was used because it has the highest level of accuracy of all neural networks in previous works [5]. In order to use an RNN, all location data (latitude and longitude) are fed into the network as their absolute values. An early stopping callback was used to minimize the loss of the validation data and to prevent over-fitting. The callback looked to minimize the parameter "val_loss" and had a patience level of 5. This means that if the loss value does not decrease for 5 epochs, the training would terminate and the best iteration would be chosen for the final model. The RNN is run for 100 epochs or until the EarlyStopping callback is called. Since most crashes are typically Property Damage Only (PDO), early versions of the RNN predicted only PDO as an outcome. To combat this behavior, the classes were given weights which forced the model to view all severity outcomes as equal possibilities. Unfortunately, this change biased the results too far in the opposite direction. Models with balanced weights predicted a fatal severity almost always. The final version of the RNN limited the added weights so they could only have a value of at most 20. The final model was able to predict classes in the most balanced way so it was chosen as the model to use going forward.

2.2 Decision Trees

A decision tree is a series of branching binary decisions based on the data fed into it. The decision tree decides which data feature to split first based on which feature will cause the greatest gain in information. The information gain is calculated by checking to see which data feature results in the largest amount of correct classifications when it is split. The decision tree typically uses entropy when deciding which features to split. Entropy, in general, is the measure of disorder. It is the goal of the decision tree to lower the entropy as much as possible in order to classify as many

cases correctly as it can [6].

A decision tree classifier was created using the sklearn DecisionTreeClassifier package in Python. In order to combat a mismatch in the amount of data in each category, the weights of the classes were balanced. Unlike the neural network, the decision tree did not have an issue with fully balanced weights. Due to the complexity of the data, there was no limit on the number of levels the decision tree could have. By default, the tree used entropy to calculate when and how to split its nodes.

2.3 Models in Crash Severity Prediction

The hope of this research is that a broader dataset will allow for a model to accurately predict crashes from a larger geographic region. In the field of traffic analysis, models are trained on datasets from very specific locations such as states [7, 8], cities [9, 10], or roadways [11, 5]. By training on specific datasets, the model can only be used within the small region it was trained on, as it has no data regarding locations outside of that region. It is hypothesized in this study that through the process of combining multiple state datasets into one, a model will be able to effectively predict crash severity over a larger region. Past studies show that neural networks are effective at classifying crash severity [12, 5]. In addition, decision trees are also a popular choice for use in severity prediction [11, 9, 13]. Therefore, a neural network and a decision tree were used in this project to test the wide variety of datasets. Unlike past studies, the focus of this project is on the dataset itself rather than the inner workings of the machine learning models.

When a decision tree is used as a model for severity prediction, the overall accuracy of the output ranged between 34.06% - 91.70% with an average overall accuracy of 66.92% [11, 9, 13]. On the other hand, neural networks had an average overall accuracy of 65.18% [5, 9]. Recurrent neural networks were found to have the highest overall accuracy when compared to other types of machine learning models, such as Multi-Layer Perceptrons and Bayesian Logistic Regression [5]. It was also found that location [5], weather condition [10, 8, 7], type of collision [9], and road condition [13, 10] were all major factors in the outcome of a vehicle collision. However, both types

of models appear to have a more difficult time predicting severe crashes. Specifically, fatal crashes are often never predicted correctly by the models [8, 10]. In general, it seems that there is a good baseline in collision severity prediction but there is still room for improvements to be made.

Chapter 3

Data Creation Methodology

3.1 Creation of Dataset From Multiple States

A Python script was created to find common data features from a list of data (csv) files that hold crash records from different states. The script was able to find the data features that were present in all of the state data files provided, and combine them into a single data file that contained crash records from all of the states provided. The script accomplished this by using hard-coded lists of data feature titles. The hard-coded lists were a way to show that a single data feature could be represented by many different names depending on the state that reported it. The software and data created in this project will be available through Penn State's ScholarSphere service at DOI:10.26207/52t2-tz38.

The script searched through the first non-empty row of the first state data file provided. The first non-empty row would be the row containing the names of the data features. The script then checked to see if any of the data features present in the hard-coded lists (described above) were present in the file. If a feature was found to be present, then the column containing the feature and the index of that column would be saved as the key and value in a Python dictionary. In addition, a second dictionary known as "available categories" was set up to count the number of files that contained a given feature. This dictionary was originally set so all features have a value of 0, meaning the feature was found in zero of the files given. The value in the "available features" dictionary was incremented by 1 for a given feature if it was found in a state data file. The subsequent files were then searched in a similar manner to the first file, with the dictionaries being updated accordingly. When all of the state data files had been searched, only the features that had a value in the "available features" dictionary equal to the total number of files given were included in the final dataset. This process ensures that only data present in all of the states given would be included in the final dataset.

The data features were then converted to their corresponding numerical values. The numerical forms of the data were based on the meaning of the data found. As an example, the numerical forms of the weather data are shown in Table 1.

Table 1: An example converting categorical data to numerical data.

Categorical Data Value	Numerical Value
Clear	1
Cloudy	2
Rain	3
Snow/Sleet	4
Fog	5

This numerical normalization was done because, similar to the data feature titles, the states may report data in different words that mean the same thing. If left untouched, the data would confuse the machine learning model. In the end, the script had a count of how many files had a specific feature, and the correct index of where to find that feature in each file. A random order was generated to ensure that all data points were in the correct order when they were written back to a training file. The data was split into 80% training and 20% testing from a random set of 125,000 cases. Finally, the data was written to the corresponding training and test files, thus creating a single consolidated dataset containing data from multiple states.

3.2 Problems with Multi-State Data Reporting

3.2.1 Data Feature Consistency and Naming Conventions

The major issue when attempting to utilize data provided by different states was the lack of consistency in the data features provided. In addition, the naming standard for data features varied greatly from state to state. As a result, reading a file and finding common data features between states became a more difficult task. In addition, many states reported their levels of severity differently. The standard of severity levels in vehicle collisions is described in Table 2. Some states, such as

Massachusetts, only report severity levels in a binary fashion. This means that all crashes provided by the state of Massachusetts are classified as either "Non-Fatal Injury" or "Property Damage Only". As a result, the binary nature of the classifications could skew predictions towards those two specific severity levels, since there is more data pointing towards them.

Table 2: Collision Severity Classifications

Severity Description	Severity Level Code
Property Damage Only (PDO)	0
Minor Injury Suspected	1
Moderate Injury Suspected	2
Major Injury Suspected	3
Fatality	4

The models were still able to predict with some level of accuracy when trained on a dataset consisting of states that have different severity levels, but the consistency varied greatly with each iteration. Specifically, this inconsistency occurred mostly in the neural network. The decision tree was typically constant with its predictions. It was very common for models trained on states with mismatched severity levels to predict that all crashes are PDO. The problem then became how to change the model so that it predicts more accurately. Unfortunately, in an instance where all PDO was predicted, the overall accuracy would be fairly high since the large majority of crashes are actually PDO. However, a model that predicts only non-severe crashes is not of any use to researchers. Therefore, the model in this project attempted to optimize the loss of the validation data instead of the accuracy. This way, the model has a better chance of predicting all collisions correctly instead of consistently predicting PDO to raise overall accuracy.

3.2.2 Common Data Features

The number of common data features that can be utilized in the creation of a new dataset decreases drastically with the addition of more states. As a result, the more data added to the dataset yields lower quality data per crash due to fewer features. Therefore, the datasets in this project

examined how the number of features for each crash affected the accuracy of each model. There were a total of 9 data features that were searched for when the data was being processed. These features were chosen due to their frequency of appearance in the states selected for this project. These data features provided an ample pool of features that could be used when combining states.

Table 3: Data Feature Description

Data Feature	Abbrv.	Short Description of What Feature Reports
Latitude	LAT	The latitude at which the crash occurred
Longitude	LONG	The longitude at which the crash occurred
Road Condition	RC	The state of the road at the time of the crash
Light Condition	LC	The amount of light present at time of crash
Weather Condition	WC	The weather at the time of the crash
Collision Description	COL	The manual description of the crash
Day of the Week	DAY	The day of the week the crash occurred
Number of Vehicles	NUMV	The number of vehicles involved
Number of Lanes	NUML	The number of lanes on the road

Table 4: Data Feature Possible Outputs

Data Feature	Possible Outputs
Latitude	Numerical Value
Longitude	Numerical Value
Road Condition	Dry, Wet, Ice, Mud/Dirt/Gravel, Snow/Slush
Light Condition	Dark - Unlit Roadway, Dark - Lit Roadway, Daylight, Dawn/Dusk
Weather Condition	Clear, Cloudy, Rain, Snow/Sleet, Fog
Collision Description	Single Vehicle Crash, Rear-End, Angle, Sideswipe
Day of the Week	Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday
Number of Vehicles	Numerical Value
Number of Lanes	Numerical Value

The most important feature that must be reported by each state is the crash severity. This feature is essential in order for supervised learning to occur. The standard for crash severity prediction is to classify all collisions into one of 5 different features. The features increase in severity as their

number increases. However, each state reports crashes in a different way. Therefore, states often have a different number of severity levels. The classifications of each severity level remain the same, but the number reported may differ. For instance, some states may not include a classification of possible injury (severity level #1) and instead report it as something else. Table 5 lays out the features present in each state and the severity levels that get reported.

Table 5: Data Feature and Severity Classes By State

State	Relevant Features	Severity Levels
Massachusetts	RC, LC, LAT, LONG, COL, WC, NUML, NUMV	0, 2
Maryland	RC, LC, LAT, LONG, COL	0, 2
Pennsylvania	RC, LC, LAT, LONG, DAY	0, 1, 2, 3, 4
New York	DAY, NUMV, RC, WC, COL	0, 2, 4
Illinois	LAT, LONG, DAY, NUML, LC, NUMV, COL, RC	0, 2, 4
Ohio	LAT, LONG, COL, WC, LC	0, 1, 2, 4
Iowa	LAT, LONG, NUMV, LC, WC, DAY	0, 1, 2, 3, 4

3.3 Datasets Used for Evaluation

There are a number of factors that are in consideration when choosing states to include in the combined dataset. The considerations are as follows:

- The number of data features the state has in common with others in the dataset
- The location of the states when compared to other states in the dataset
- The levels of severity reported by each state
- The number of crashes the state reports compared to others in the dataset

The goals when selecting states to include in the datasets are to test the impact of as many of the above features as possible. In order to do this, states were divided based on the amount of data reported and the levels of severity reported. States that report a large amount of data were paired to

states with small amounts of data to see if the larger state had more of an impact on the accuracy. The locations of the states were also taken into account. States that were close together to each other were tested in combination to see how location has an impact on accuracy. In addition, states that were far apart were tested together in order to tell if distance increases or decreases accuracy. The number of severity levels reported was taken into account, but it seemed that this had little to no effect on overall accuracy.

Chapter 4

Evaluation of Predictive Models

The results in this chapter were generated based on the outcomes of the machine learning models run on a personal computer. Both models were created using the Python library and were run locally on a Python IDE. The state datasets were also stored locally on the personal computer as comma separated value (csv) files. The data files were normalized as described in Chapter 3, randomly sampled, and combined into a single array. The single array was divided into the training and test datasets. The training and test samples were written to different csv files. The training file was then fed into both the RNN and the decision tree in order to get a fit for the data. The test data file was then read and the confusion matrices were printed for both models. The accuracies seen in this chapter are based on the confusion matrices that were generated in the testing process.

4.1 Neural Network Results

4.1.1 Single State Dataset

Table 6: Single State Dataset Neural Network Accuracy

State	# Features	Class 0	Class 1	Class 2	Class 3	Class 4	Overall
Massachusetts	9	83.06%	N/A	23.60%	N/A	N/A	70.51%
Maryland	5	56.76%	N/A	56.34%	N/A	0.00%	56.35%
Pennsylvania	5	44.18%	18.75%	18.64%	0.00%	55.11%	37.48%
New York	5	50.43%	N/A	74.42%	N/A	0.00%	55.44%
Illinois	8	99.74%	N/A	0.56%	N/A	0.00%	83.29%
Ohio	5	40.25%	32.02%	45.96%	N/A	0.00%	39.67%
Iowa	6	91.89%	9.43%	100.00%	0.00%	0.00%	78.55%

The results seen in Table 6 are fairly consistent with those seen in Sameen and Pradhan [5]

which had an average accuracy rate of 65.48% for its RNN. The interesting results here are in the cases of Ohio and Pennsylvania, which have accuracies that are lower than average. Ohio is particularly interesting because all other states near it (Illinois and Iowa) have significantly higher accuracy rates. It also appears that all states that have 5 features to utilize are lower than those that have more features to work with.

4.1.2 Two State Dataset

Table 7: Two State Dataset Neural Network Accuracy

States	# Features	Class 0	Class 1	Class 2	Class 3	Class 4	Overall Accuracy
MD, PA	4	90.42%	0.00%	3.11%	14.53%	40.64%	63.20%
PA, NY	2	88.28%	0.00%	11.21%	78.13%	0.00%	66.60%
NY, MA	4	41.88%	N/A	79.17%	N/A	0.00%	50.15%
MA, MD	5	94.63%	N/A	3.45%	N/A	N/A	67.37%
IL, OH	4	20.27%	95.92%	54.35%	N/A	0.00%	30.60%
IL, IA	5	3.63%	99.90%	97.58%	0.00%	0.00%	19.16%
IA, OH	4	25.56%	31.95%	14.32%	67.72%	0.00%	26.13%
IA, MA	5	0.00%	99.37%	99.14%	0.00%	0.00%	19.98%

The results in Table 7 indicate a mixed improvement in accuracy. In the Northeast region, the average overall accuracy increased from 54.945% in the single state section to 61.83%, which hints at the possibility that adding states could in fact be an effective way to increase performance. In fact, there are a few cases of a combination of states performing better than both single states that make it up. However, the opposite effect is seen in the Midwest region. The Midwest saw a decrease in overall accuracy from an average of 67.17% to 25.29%. The combination of Iowa and Illinois is particularly perplexing because both states performed very well in the single state dataset, but dropped dramatically when combined. In addition, the combination of states that was the farthest away, Iowa and Massachusetts, performed poorly when compared to the states that are within it.

4.1.3 Multi-State Dataset

Table 8: Multi-State Dataset Neural Network Accuracy

States	# Features	Class 0	Class 1	Class 2	Class 3	Class 4	Overall
MD, PA, MA	4	94.02%	0.96%	65.57%	0.00%	19.85%	65.54%
MD, NY, MA	2	41.34%	N/A	77.08%	N/A	0.00%	52.48%
IL, OH, IA	3	25.34%	31.33%	38.94%	73.46%	0.00%	28.29%
NY, MA, OH	2	62.13%	63.00%	0.00%	N/A	0.00%	51.23%
IL, PA, MD	4	56.74%	41.91%	94.74%	0.00%	3.13%	52.12%

The results seen in Table 8 again show that each region performs oppositely of the other. In this case, the Northeast decreased in accuracy from its two-state datasets. The Northeast actually performed the worst overall in this section, dropping to a low average accuracy of 59.01%. The Midwest, when combined, had an overall accuracy of 28.29%. The Midwest increased in accuracy from its two-state datasets, but still well performed below the average in the single state section. It also seems that all cross-region combinations seem to average around 50.00% in accuracy.

4.2 Decision Tree Results

4.2.1 Single State Dataset

Table 9: Single State Dataset Decision Tree Accuracy

State	# Features	Class 0	Class 1	Class 2	Class 3	Class 4	Overall
Massachusetts	9	77.42%	N/A	28.33%	N/A	N/A	66.84%
Maryland	5	70.42%	N/A	35.52%	N/A	2.50%	59.25%
Pennsylvania	5	63.15%	0.00%	1.70%	17.20%	26.14%	47.09%
New York	5	47.48%	N/A	50.38%	N/A	37.50%	48.09%
Illinois	8	84.05%	N/A	24.09%	N/A	0.00%	74.06%
Ohio	5	76.23%	17.32%	14.53%	N/A	2.44%	61.83%
Iowa	6	82.63%	19.02%	24.58%	11.76%	3.28%	72.16%

The results in Table 9, when compared to the results in Chang and Wang [9], show that the overall accuracy is improved, but the predictions seem to be skewed more towards a Property Damage Only classification. Chang and Wang found an overall accuracy of 34.06% while the average accuracy in this project is 61.33%. However, Chang and Wang found that the accuracies for each class were relatively even, where as in this project the accuracies are heavily skewed to less fatal. Overall, the accuracies are relatively high. Unlike the neural network, Ohio is able to achieve an accuracy on par with its region. Again, the Northeast has lower accuracies than the Midwest when it comes to single state average. The average for the Northeast region is 55.32% while the Midwest has an average overall accuracy of 69.35%.

4.2.2 Two State Dataset

Table 10: Two State Dataset Decision Tree Accuracy

States	# Features	Class 0	Class 1	Class 2	Class 3	Class 4	Overall Accuracy
MD, PA	4	68.10%	0.00%	33.34%	21.37%	10.70%	56.90%
PA, NY	2	13.53%	50.00%	79.27%	36.46%	9.52%	31.63%
NY, MA	4	34.92%	N/A	64.00%	N/A	30.00%	41.39%
MA, MD	5	71.14%	N/A	34.15%	N/A	0.00%	59.99%
IL, OH	4	77.53%	18.05%	21.67%	N/A	0.00%	64.68%
IL, IA	5	81.60%	18.28%	26.87%	10.00%	0.00%	70.15%
IA, OH	4	69.35%	20.36%	13.76%	5.27%	0.00%	54.51%
IA, MA	5	77.39%	23.15%	31.82%	11.11%	1.96%	67.14%

Table 10 indicates that the accuracy in both regions has dropped. In the Northeast, the average accuracy dropped significantly to 47.48% whereas in the Midwest the drop was less significant to 63.11%. The inter-region pairing performed better than expected since its overall accuracy was in between the two states that made it up. However, it is starting to seem that a decision tree is not a good model to use for state combination.

4.2.3 Multi-State Dataset

Table 11: Multi-State Dataset Decision Tree Accuracy

States	# Features	Class 0	Class 1	Class 2	Class 3	Class 4	Overall
MD, PA, MA	4	69.31%	0.00%	32.67%	19.67%	8.40%	57.84%
MD, NY, MA	2	33.29%	N/A	67.55%	N/A	50.00%	38.20%
IL, OH, IA	3	71.11%	21.51%	16.39%	5.18%	2.46%	57.03%
NY, MA, OH	2	32.42%	31.50%	48.55%	N/A	62.50%	35.23%
IL, PA, MD	4	70.10%	0.00%	33.24%	25.26%	10.00%	58.55%

The results in Table 11 show that the Northeast improved with the addition of another state with an average of 51.02%, but it still does not perform better than the single state average. Similarly, the Midwest also performed worse than its single state average. The Midwest had an accuracy of 57.03%. The cross-region combinations have a similar result with a lower accuracy than single states. Overall, it appears that the decision tree model is not compatible with state combination used for this purpose.

4.3 Consistency of Results

The results shown in the tables presented in this chapter are the maximum accuracy in a series of three runs of the models. The maximum result was taken in order to show the best possibility when running the datasets. However, despite the maximum being used, it is worth noting that each iteration in the three-run series typically did not vary from each other by more than 5% in overall accuracy. The decision tree was the more consistent of the two models as its overall accuracy tended to stay within $\pm$ 1% for each iteration on the same dataset. Overall, the results seen in this project are reproducible due to the consistent nature of the outcomes produced.

Chapter 5

Discussion

5.1 Reasoning for Choice of State Combinations

The two state combinations of states were chosen in a way to test a wide variety of cases that factor in details such as the amount of data reported, location of the state, accuracy of single state dataset, and number of severity classes used. The multi-state datasets were created mostly to discover patterns based on distances between states. Specifically, the datasets were made to answer whether states from a certain region would perform better together than states outside of that region. For example, Massachusetts, New York, Pennsylvania, and Maryland represented the Northeastern region of the United States while Illinois, Iowa, and Ohio represented the Midwest. The idea was to test to see how combinations from the same region would perform compared to combinations of separate regions.

5.2 Comparison of Results Between Datasets

5.2.1 Neural Network

There appears to be a downward trend in accuracy when more states are used in a dataset for a RNN. However, there are some instances, such as Pennsylvania and New York, where a combination of states yields better results than all of its single state parts. In the case of Pennsylvania and New York, the individual accuracies are 37.48% and 55.44% respectively, but the accuracy becomes 66.60% when combined. However, there are also instances where a combination of states does significantly worse than the single state results of its parts. For instance, Iowa and Illinois have high accuracies individually at 78.55% and 83.29% respectively, but when combined the accuracy falls

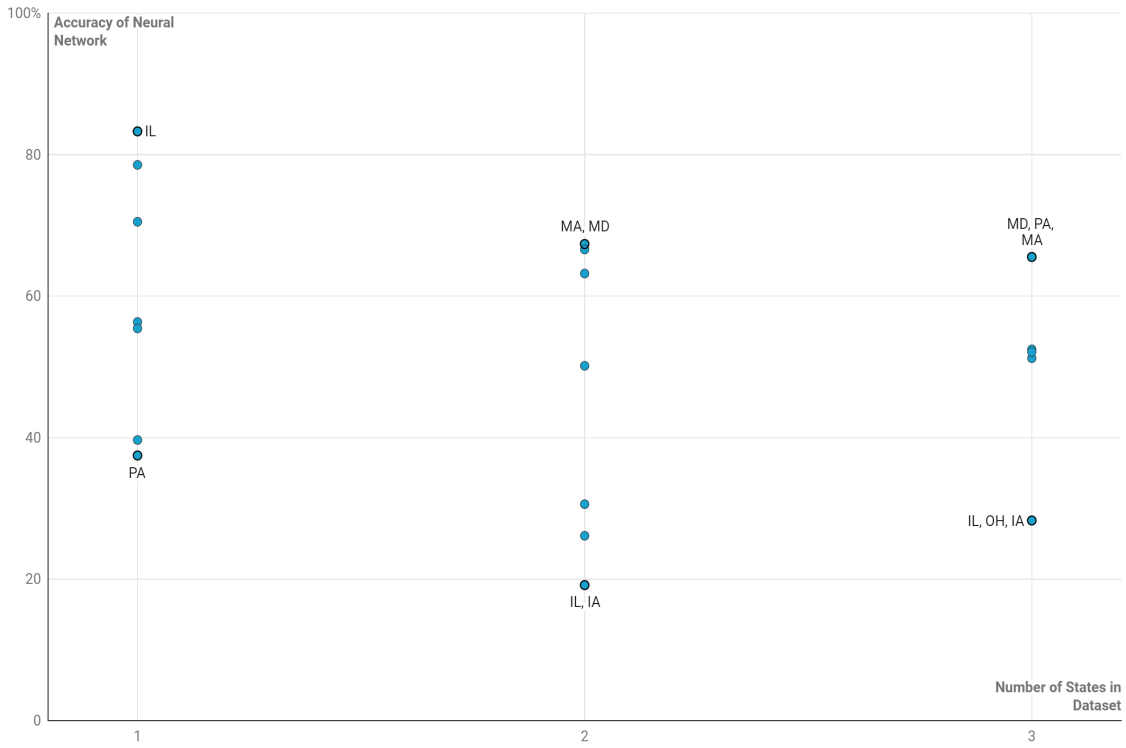

Figure 1: Accuracy of Neural Network Compared to Number of States in Dataset

to 19.16%. These examples demonstrate a pattern of unpredictability with the outcomes. While the percentages themselves are very consistent between different runs of the same dataset, there is a clear randomness when it comes to the outcomes of different combinations.

Since there is no apparent reason for why some state combinations provide a better result than others, more data was collected based on the results found in this project. Two possible contributors to the accuracy of a state combination were the number of features that the combination had in common with each other, and the distance between states in each dataset. These two contributors were analyzed further to see if there were any patterns to be discovered.

The number of data points that each dataset has in common does not seem to affect the overall accuracy of the neural network. Therefore, it is safe to assume that there is no correlation between the number of data features included in the dataset and the accuracy of the model for this RNN. This is an interesting result because it is commonly thought that the more data features the model has to use, the more patterns it can find and thus the more accurate. There might be what can be

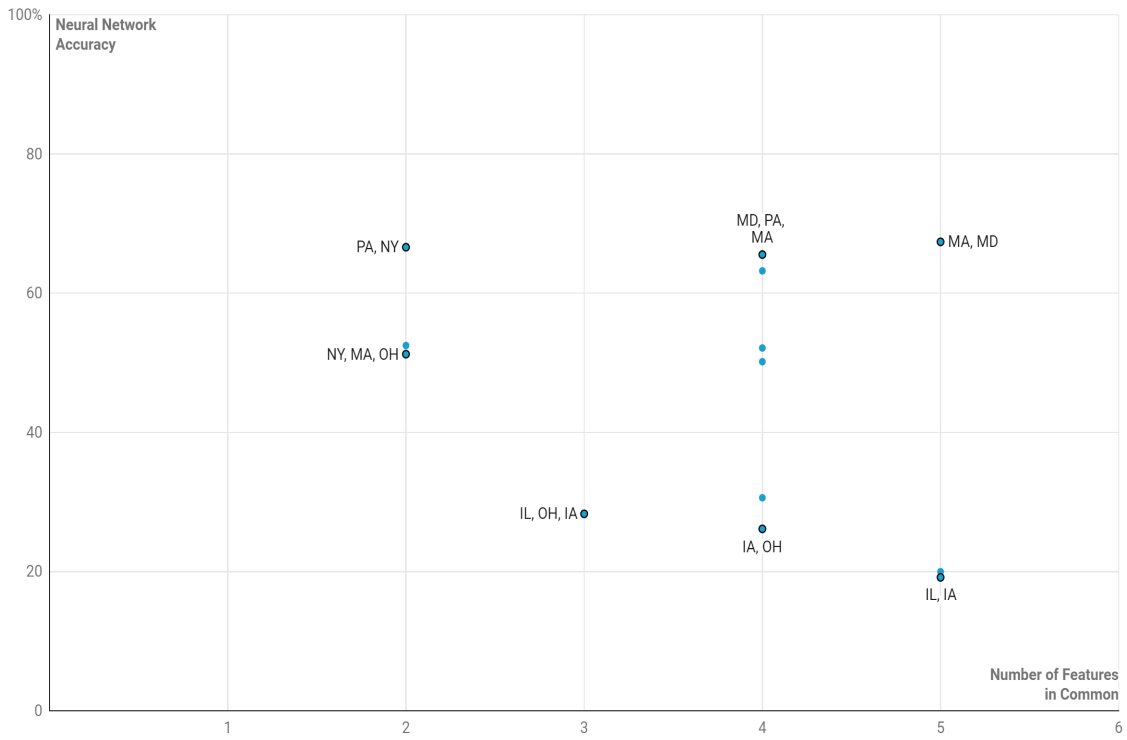

Figure 2: Accuracy of Neural Network Compared to Common Features

considered a small decrease in accuracy as more data features are included but this seems to most likely due to the stochastic nature of the results in general.

While the data in this plot has a lot of noise, there is a clear downward trend when it comes to increasing distance between states. In this plot, the distance between the centers of the states was taken (in multi-state datasets the average distance was used). It is possible to conclude that the neural network does better when the states it is using in its dataset are close together geographically. Perhaps this is due to the possibility that a shorter distance between states could mean that drivers are similar in their habits and tendencies. It is also possible that states that are geographically linked develop in similar ways and thus have similar roadways. It is clear that the neural network in this project performs better when data from states close together are combined.

The cases where a combination of states produces a higher overall accuracy than its parts are from the datasets of Pennsylvania / Maryland and Pennsylvania / New York. These two datasets are also the two lowest distances between states. Therefore, it is a possibility that two states that

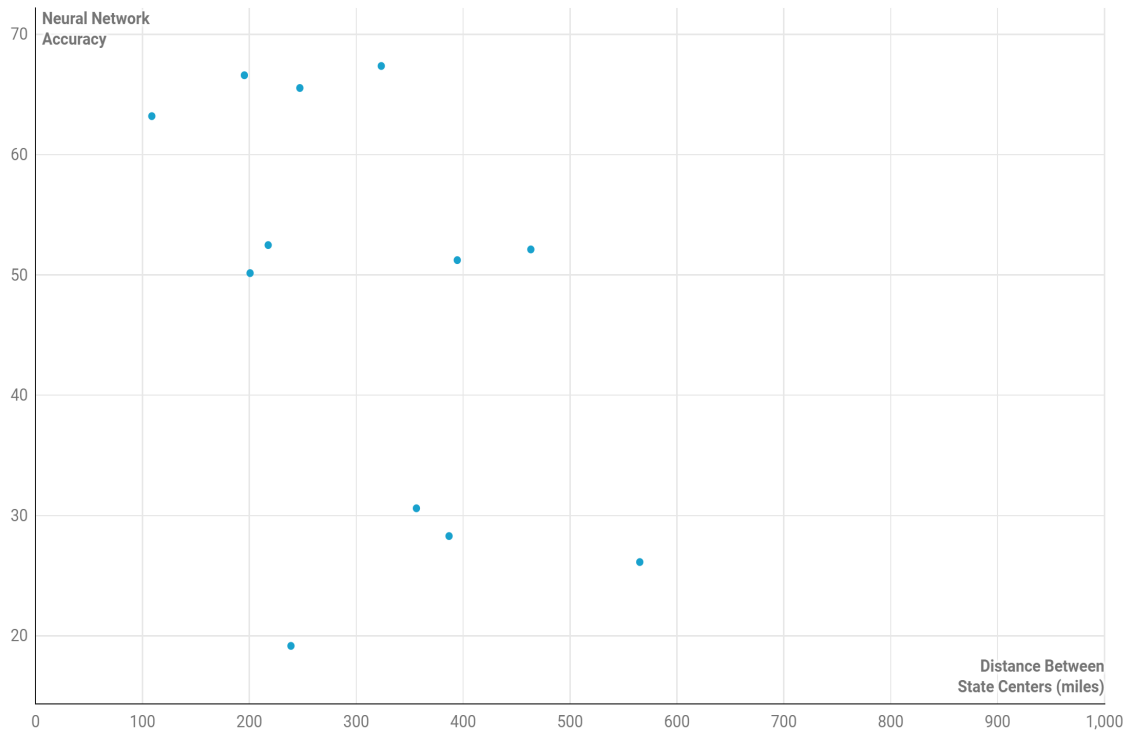

Figure 3: Accuracy of Neural Network Compared to Distance

are geographically close could produce a better result when combined than when used separately. This finding could be important for future work in the field since this could become an easy way to improve accuracy while simultaneously increasing the geographic range the model can predict.

5.2.2 Decision Tree

The decision tree results consistently get less accurate with the addition of more states. Unlike the neural network, the decision tree does not have a single case where a combination of states does better than all of its single state components. Therefore, it seems unlikely that a decision tree is a model that can be utilized when combining states for traffic collision severity prediction. However, it is still worth looking into what factors affect the results of the decision tree model.

Unlike the neural network, the decision tree shows a clear upward trend in accuracy with an increase in the number of common features. This result makes sense because more data features allow for more possible branches to use in the decision tree. The relationship between the number

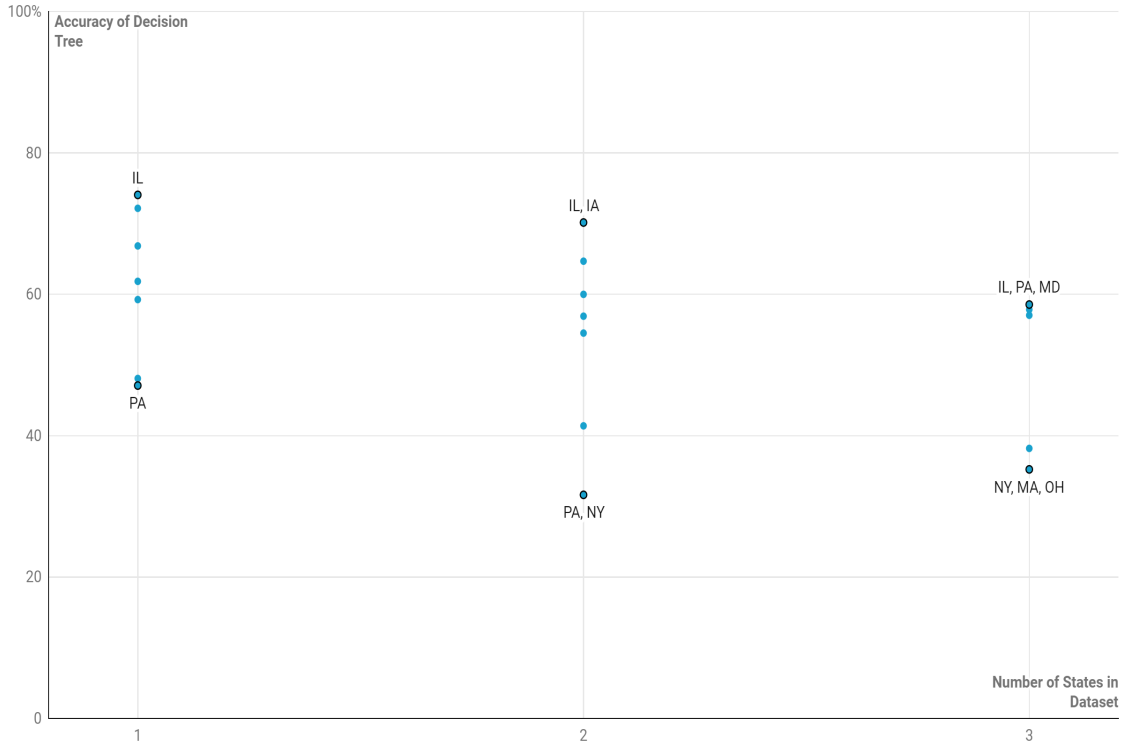

Figure 4: Accuracy of Decision Tree Compared to Number of States in Dataset

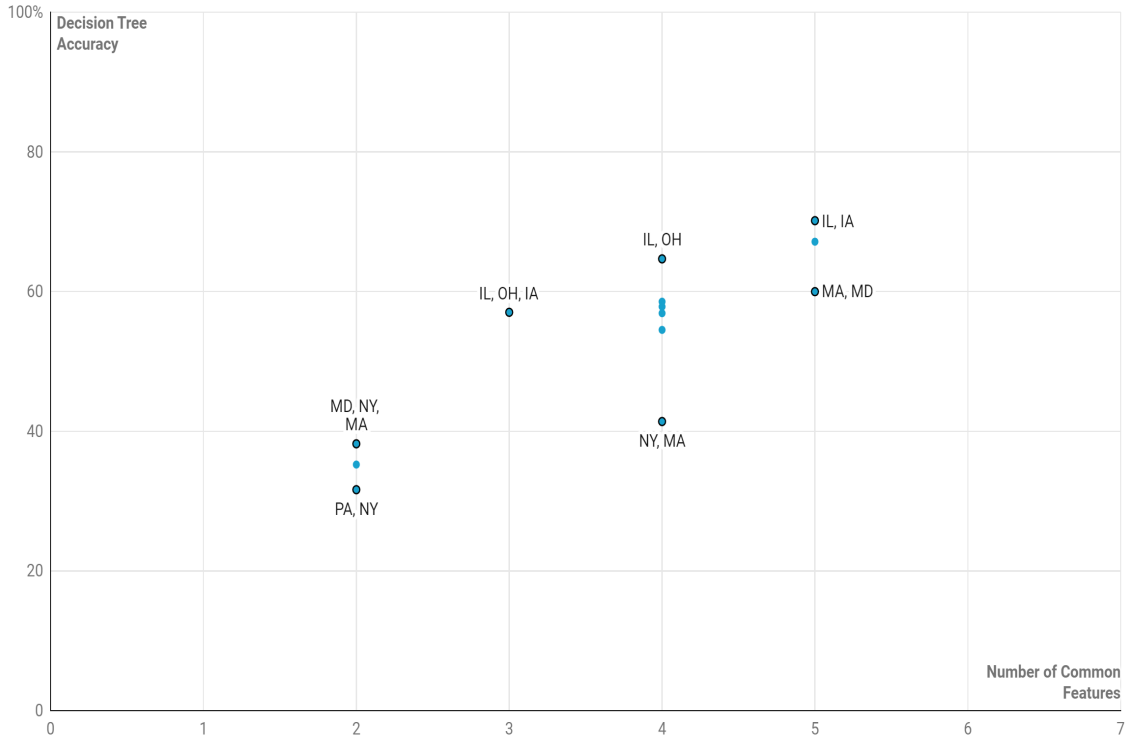

Figure 5: Accuracy of Decision Tree Compared to Common Features

of common features and overall accuracy of the decision tree is very clear and based on this information, the conclusion can be drawn that the number of features is one of the major factors that affect the accuracy of the decision tree in this project.

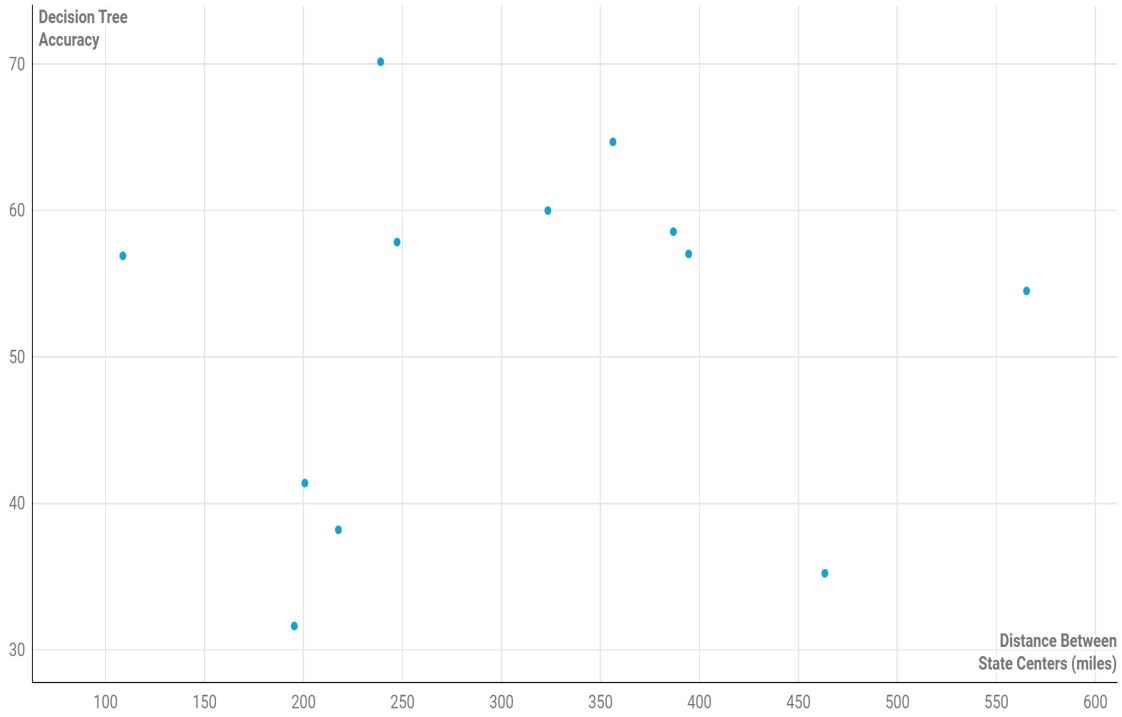

Figure 6: Accuracy of Decision Tree Compared to Distances

There is no apparent correlation between the distances between the states and the overall accuracy of the model. Since the nature of a decision tree is to create a series of branches based on the values, it is possible that the location data of each crash is divided. That is to say, a decision tree could find that all crashes within a certain range of latitude and longitude should be treated differently than crashes from a different area. The decision tree could be sending data from one state down one side of the tree and the data from another state down the other. This would mean that the crashes in the total state could be divided based on the state they came from and treated separately based on that information. Therefore, it would make sense that there is no correlation between state distances and accuracy because the decision tree doesn't take into account the distance between states as a factor in its decision.

Chapter 6

Conclusion and Future Work

Based on the results in this project, the neural network model appears to be the better model to use in multi-state data prediction. While there may not always be an advantage to combine records from multiple states into a unified dataset, there are interesting observations that are worth a closer look. The dependence on the distance between states in a dataset could possibly be used to improve the accuracy of poor performing models. The cases where a combination results in higher accuracy than the individual states that make it up occur, in this project, exclusively in datasets are closer together. Therefore, future work can include the investigation of other states to see if this pattern holds and can be confirmed. Crashes in rural locations can also be possibly combined in future work. In addition, the neural network results showed that there is seemingly no dependence on the number of common features in a dataset. This could mean that a model can predict well with less complex data which would possibly increase the speed of the calculations. In general, based on these findings, a RNN seems to be a promising model to use when trying to combine multiple states into a single dataset for crash severity prediction.

The decision tree is seemingly the opposite of the neural network in terms of dependence. Instead of being correlated to the distance between states, the decision tree accuracy is heavily correlated to the number of common features in the dataset. This is most likely due to the fact that a decision tree could be treating the states' data as separate entities. Overall, there do not seem to be many benefits to combining records from multiple states when using a decision tree.

Additionally, the data processing software that was developed for this project was able to successfully find similarities in data features reported by multiple states. The software developed in this project has the potential to help the future of multi-state crash severity prediction. This software could help researchers bypass the differences in data reporting from different states and allow for one large dataset to be utilized. This software can be improved, however, by removing

hard-coded features. Currently, all categories reported must be hard coded in order for the software to understand which categories correspond to one another. It is possible that in the future a machine learning algorithm could be applied to find which categories are similar to each other so that the user does not have to manually enter the categories they want to include from each state. The software developed in this project shows great potential for future use in the field of crash severity prediction.

Despite all of the findings on the RNN and decision tree, there is still a clear downgrade in accuracy when there is an addition of states to a dataset. This trend is present in both of the models tested. However, there are multiple other models that could have an increase in overall accuracy when data from different locations is included. Future work could include focusing on other models that might prove to be more effective in increasing accuracy. In addition, there is more to focus on in the RNN model shown in this project. Specifically, future work could investigate more into what factors cause the random increases in accuracy that occur in some datasets in the RNN model. In general, the observations in this project could hopefully help in the future of crash severity prediction and allow for more models to predict outcomes on data that comes from a larger span of locations.

Bibliography

[1] World Health Organization. Road traffic injuries. https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries, accessed April 2021.

[2] Warren S. McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133, 1943.

[3] Y. Bengio, P. Simard, and P. Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, 5(2):157–166, 1994.

[4] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[5] Maher Ibrahim Sameen and Biswajeet Pradhan. Severity prediction of traffic accidents with recurrent neural networks. *Applied Sciences*, 7(6), 2017.

[6] Leo Breiman, Jerome Friedman, Charles J Stone, and Richard A Olshen. *Classification and regression trees*. CRC Press, 1984.

[7] Qiang Zeng and Helai Huang. A stable and optimized neural network model for crash injury severity prediction. *Accident Analysis & Prevention*, 73:351–358, 2014.

[8] Heejin Jeong, Youngchan Jang, Patrick J. Bowman, and Neda Masoud. Classification of motor vehicle crash injury severity: A hybrid approach for imbalanced data. *Accident Analysis & Prevention*, 120:250–261, 2018.

[9] Li-Yen Chang and Hsiu-Wen Wang. Analysis of traffic injury severity: An application of non-parametric classification tree techniques. *Accident Analysis & Prevention*, 38(5):1019–1027, 2006.

[10] Sharaf Alkheder, Madhar Taamneh, and Salah Taamneh. Severity prediction of traffic accident using an artificial neural network. *Journal of Forecasting*, 36(1):100–108, 2017.

[11] Ali Tavakoli Kashani and Afshin Shariat Mohaymany. Analysis of the traffic injury severity on two-lane, two-way rural roads based on classification tree models. *Safety Science*, 49(10):1314–

1320, 2011.

[12] D. Chimba and T. Sando. The prediction of highway traffic accident injury severity with neuromorphic techniques. *Advances in Transportation Studies*, pages 17–26, 2009.

[13] Mahdi Rezapour, Amirarsalan Mehrara Molan, and Khaled Ksaibati. Analyzing injury severity of motorcycle at-fault crashes using machine learning techniques, decision tree and logistic regression models. *International Journal of Transportation Science and Technology*, 9(2):89–99, 2020.

Academic Vita

Thomas M. England

Born: May 14, 1999—New Brunswick, New Jersey

Nationality: American

Current position

Software Engineer, Lockheed Martin, King Of Prussia

Areas of specialisation

Machine Learning

Positions held

2020 - Whirlpool Corporation, Benton Harbor, MI

2019 - Lockheed Martin, Moorestown, NJ

Education

2017 - 2021 - BS in Computer Engineering, The Pennsylvania State University

Grants, honours & awards

2018 - President's Freshman Award

2019 - President's Sparks Award

2019 - 1st Prize at Brandywine Campus Undergraduate Research Symposium

2019 - 2nd Prize at Regional Undergraduate Research Symposium