Time Series Forecasting and Predictive Modeling of Automobile Accident Severity in Virginia

Saarthak Gupta
School of Engineering and Applied Science
University of Virginia
Charlottesville, VA 22903
uzn2up@virginia.edu

Joshua Seiden
School of Engineering and Applied Science
University of Virginia
Charlottesville, VA 22903
vuc7cu@virginia.edu

Agi Luong
College of Arts and Sciences
University of Virginia
Charlottesville, VA 22903
xwq5ja@virginia.edu

December 4, 2023

Abstract

In this paper, we model a classification problem on car crash data from the Virginia Department of Transportation (VDOT), which contains over 1 million crash records from 2015 to 2023. With the rising number of tragic losses from car accidents, analysis of this data can help save lives. We predict the severity level of accidents based on environmental, social, and geographic factors and discover major characteristics that contribute to lethal crashes. For this purpose, we explore machine learning algorithms like random forests, logistic regression, and Artificial Neural Networks (ANNs). We describe data encoding and cleaning methods for this mostly categorical dataset and employ oversampling techniques to handle highly imbalanced classes in the data. The goal is to provide actionable insight for drivers and the Virginia Department of Transportation to increase road safety. State-of-the-art methods for time series forecasting are used to identify trends in the number of accidents per day at the state and county levels. Using a Recurrent Neural Network-based approach, the trends in the daily crashes are predicted for up to a year into the future. Sequence generation methods for this approach are also described.

1 Introduction

With the rise of machine learning and predictive analysis, there exists an unprecedented opportunity to proactively address challenges associated with car accidents. This paper delves into the intricate landscape of road safety, seeking to leverage the power of machine learning and big data analytics to predict and mitigate the risk and severity of car crashes. Against the backdrop of a significant increase in car crash fatalities in Virginia in 2022 and the continuing growth of retail car sales, our project aims to contribute to the ongoing discourse on road safety.

The two main frontiers we investigate are the classification of car accidents based on fatalities using deep neural networks on synthetic data and LSTM-based time series forecasting to glean patterns and predict the number of accidents per day at the state and county level. We also focus on related data pre-processing and data manipulation techniques. As we navigate the intricacies of car crash prediction, our ultimate goal is to enhance road safety measures and usher in a new outlook on accident prevention through the lens of machine learning.

1.1 Related work

Related work on Virginia car crash data was conducted by University of Virginia Data Science Institute graduate students. These researchers utilized mixed linear modeling techniques to merge Virginia Department of Motor Vehicles data with demographic data to build a predictive model for car crash fatalities. They determined a correlation between the geographic crash location at the county level and the outcome, whether or not there was a fatality. Their model produced better predictions than a logistic regression model that was the threshold at the time [2]. We intend to follow up on the related work by having more nuance in outcome injury level than the binary fatality or non-fatality.

Furthermore, multiple factors can affect the severity of accidents. Related work was carried out in a study in Iran in 2018 [3]. Researchers used a technique called Analytical Network Process (ANP), which includes modeling of factors affecting crash severity and assessing traffic experts' estimation of the relative weight of factors. The results are four main factors: safety, health, environment, and other factors, and all have their sub-factors along with weights. This information will be important to us in the data engineering process. For example, the safety factor holds the highest weight, with "speed over the upper limit" weighted the most among its sub-factors. As a result, we can adjust bias for "speed over the upper limit" when preparing our data for modeling.

2 Motivation

Retail car sales in the U.S. are projected to continue growing well into 2024, and more cars on the road lead to a higher likelihood of adverse events like car crashes. Car crash fatalities increased to a 15-year high in Virginia in 2022, and historical crash data is a significant predictor of future events and deaths (see appendix Fig. 1). Our project aims to address the issue of car accidents by predicting the risk and severity of car crashes on different roads in the Commonwealth of Virginia. This is an application of machine learning that has the potential to improve road safety and help drivers make informed decisions about their routes and associated risks.

3 Method

Here, we describe the setup and various experiments we performed on the data.

3.1 Data

The dataset from the Virginia Department of Transportation may be found at:

`https://www.virginiaroads.org/datasets/crashdata-basic-1/about`

The dataset has 1,048,311 samples with 69 features incorporating factors like road conditions, crash type, location, vehicles involved, people involved, etc. These features can be used to predict crash risk at various locations in Virginia. The dataset mostly has categorical features and a few numerical features.

3.2 Preprocessing for Classification

To prepare for our preliminary experiments, we conducted exploratory data analysis using geospatial plots and histograms to understand the data distribution better (see Figs. 1-3 in the appendix). As the first data cleaning step, we filled in null values for the work zone location and work zone type features. Then, we dropped 18 of the 69 features. Most of the dropped features provided redundant location information for GIS systems or contained very few samples with non-null values. Most importantly, we dropped features that conveyed the exact number of people killed or injured in a crash, as keeping these would make predicting our target variable (crash severity) trivial. In the real world, we cannot have information about injuries or death toll before the crash occurs. A few examples with invalid location values were also dropped. Each crash has a string that specifies the date and time, and this was encoded as a numerical feature.

The dataset divides crash severity into five categories (see appendix Fig. 1 for details), and our goal is to predict this class based on road, vehicle, and human factors. We prepared two versions of our data for binary and multiclass classification. In the binary version, the target column was reduced to Fatal (K or A) or Not Fatal (B, C, or O) in place of the five categories: K, A, B, C, and O. Stratified sampling was used to split the data into train and test sets based on the crash severity distribution discovered during the EDA phase (see appendix Fig. 3). Categorical features were one-hot encoded, numerical features were standardized, and the target class was encoded numerically for neural networks. After the processing step, we ended up with 283 features. The cleaned data was fed into learning algorithms, including random forest, logistic regression, and neural networks, as described in the experiments section.

3.3 Preprocessing for Time Series Forecasting

Road safety is a critical concern, and predicting the frequency of car accidents can aid in implementing proactive measures. The first step is to extract the data to construct the time series from the larger dataset. The data utilized for time series forecasting comprises daily counts of car accidents organized by date and geographical location (county or state-wide). The dataset is preprocessed to extract relevant information, including the date and number of accidents, by counting up the accidents on each day for the nine-year period. For state-level predictions, the entire dataset is considered, while for county-level predictions, data is filtered based on the specified county.

4 Experiments

The initial models trained were random forest and logistic regression multiclass classifiers. Two deep neural networks were also trained, one for multiclass classification and one for binary classification. For tuning hyperparameters, five-fold cross-validation was used. The evaluation metrics utilized were precision, recall, F1 score, and accuracy.

4.1 Random Forest for Multiclass Classification

Random forest was chosen as it is an ensemble model that averages multiple decision trees and is less sensitive to noise and outliers. We used grid search to train a random forest model for multiclass crash severity classification to boost performance and provide regularization to mitigate over-fitting. The hyperparameters we tuned and their optimal values were 200 estimators, a max tree depth of 30, and entropy as the decision criterion.

4.2 Logistic Regression for Multiclass Classification

The next model we trained for multiclass crash severity classification was a logistic regression classifier. We trained a logistic regression model because using the one-vs-rest (OvR) scheme makes it a powerful and easy-to-train multiclass model. Hyperparameters were tuned using grid search to determine appropriate regularization and solver functions. The optimal value of C was 0.5, and the SAGA solver performed the best.

4.3 ANN for Multiclass Classification

Neural networks can detect subtle patterns in data with many features, so we decided to train them for this problem. We trained a 2-layer deep neural network with 300 and 200-neuron hidden layers and a 5-neuron output layer for multiclass classification.

4.4 ANN for Binary Classification with SMOTE

The results for the above models (see Table I) led us to believe that since our data was imbalanced, we did not have enough training examples for fatal accidents, leading to poor accuracy. To overcome this, we used oversampling techniques and framed our problem as a binary classification task.

4.4.1 Oversampling and Undersampling

The Synthetic Minority Oversampling Technique (SMOTE) [1] is a statistical technique for increasing the number of fatal cases in our dataset and making it more balanced. In our original dataset, we have about 5% fatal examples, making model training difficult. Our implementation of SMOTE uses a nearest-neighbors approach to create synthetic data for the minority class. We also used Edited Nearest Neighbour (ENN) [4] for undersampling the majority class and reducing the noise in the generated data. We tried two ratios for SMOTE-ENN: 0.6 and 0.9, resulting in 665,705 and 428,177 synthetic fatal examples being generated, respectively.

4.4.2 Model Training

Once we had generated the synthetic training examples to balance the data, we trained a deep neural network with a 283-neuron input layer, a 40-neuron hidden layer, and a single neuron output layer with sigmoid activation on each of the datasets described above. For our task, false negatives are very dangerous. False negatives arise when our model fails to predict a fatal crash as fatal, and a false positive is when our model predicts a non-fatal crash as fatal. To maximize safety, it is highly important not to miss fatal accidents, even if this means more misclassification of non-fatal accidents. When determining plans to improve safety, we want to focus on areas that contribute the most to fatalities. Please see Table I in the results section for a summary of the results.

4.5 LSTM-based Approach for Time Series Forecasting

We decided to use a particular type of Recurrent Neural Network (RNN) called Long Short-Term Memory (LSTM) to model the number of crashes per day as LSTM units are capable of capturing both long-term trends, such as a yearly pattern and short-term trends, such as weekly patterns.

4.5.1 Training the LSTM

The state-level LSTM model is trained using a sequential process. The dataset is normalized using MinMaxScaler, and an 80% to 20% training-test split is performed. The LSTM model is configured with 50 units, ReLU activation, and a dense output layer. The mean squared error is employed as the loss function, and the model is optimized using the Adam optimizer. The model is trained for 50 epochs with a batch size of 32.

The county-level LSTM models follow a similar procedure but are customized for each specific county. The data is filtered to be from a particular county only. The training data is again normalized, and sequences are created with a length of 10 days. The LSTM model is trained and validated using the same parameters as the state-level model. These models can be run for any county in the state, and in this paper, we present data for the five counties with most crashes in Virginia.

4.5.2 Sequence Generation

For both state and county levels, training sequences are generated using a sliding window approach. Sequences of 10 consecutive days are used as input features, and the subsequent day's accident count is used as the target variable. This process continues throughout the training data. Sequences to make predictions for future dates are created similarly to the training sequences. The main difference is that the sequences now extend into the future. For example, to predict accidents for the next year, sequences of 10 days are created, starting from the most recent data point in the dataset.

5 Results

The code for our study can be accessed at:

`https://github.com/saarthak2002/ML4VA`

5.1 Classification Results

The Random Forest's performance on the testing data was 0.6377 precision, 0.6896 recall, and 0.6046 F1 score. The Logistic Regression's performance on the testing data was 0.6265 precision, 0.6852 recall, and 0.5939 F1 score. The multiclass Artificial Neural Network's performance on the testing data was 0.70 precision, 0.66 recall, 0.65 accuracy. These scores are low due to the underlying imbalanced distribution, making it challenging to learn patterns necessary for multi-class classification.

To achieve higher performance, we performed binary classification and dealt with the data imbalance using SMOTE to create synthetic fatal crashes so the model has more instances to learn from. We got better performance for the 0.6 SMOTE ratio data with 84.11% accuracy, our highest yet (see appendix Fig. 6 for confusion matrix). Even though the dataset with a 0.9 SMOTE ratio had lower accuracy at 79.51%, the model trained on it was able to classify a lot more fatal examples correctly (see appendix Fig. 5 and compare to Fig. 6). For this problem, false negatives are more dangerous than false positives, so the second model may still be a good option.

Our models identified the most critical features in determining the fatality risk of a crash as location, date, time, and number of vehicles involved (see appendix Fig. 4). We can infer that the location reflects the danger of specific roads, the date relates to seasonal road conditions, the time relates to driver visibility, sharpness, and traffic level, the year relates to safety technology advancements, and the vehicle count depicts increased danger in multi-car accidents.

Table I: Comparison of Classification Models on VDOT Data

Model	Parameters/Structure	Testing		
Random Forest	Optimal Hyperparameters	Precision	Recall	F1 Score

	n_estimators = 200,
	max_depth = 30,
	criterion = ’entropy’
		0.6377	0.6896	0.6046

Logistic Regression	Optimal Hyperparameters	Precision	Recall	F1 Score

	C = 0.5,
	solver = SAGA
		0.6265	0.6852	0.5939

Multiclass ANN Classifier	Network Structure	Precision	Recall	Accuracy
	Input Layer / 283 Neurons / ReLU, Dense Layer / 300 Neurons / ReLU, Dense Layer / 200 Neurons / ReLU, Output Layer / 5 Neuron / Softmax, Sparse Categorical Crossentropy, Adam Optimizer	0.7084	0.6648	0.6566
Binary ANN Classifier	Network Structure and SMOTE Minority/Majority Ratio	Precision	Recall	Accuracy
	SMOTE Ratio = 0.9, Input Layer / 283 Neurons / ReLU, Dense Layer / 40 Neurons / ReLU, Output Layer / 1 Neuron / Sigmoid, Binary Crossentropy Loss, Adam Optimizer	0.9732	0.8051	0.7951
Binary ANN Classifier	Network Structure and SMOTE Minority/Majority Ratio	Precision	Recall	Accuracy
	SMOTE Ratio = 0.6, Input Layer / 283 Neurons / ReLU, Dense Layer / 40 Neurons / ReLU, Output Layer / 1 Neuron / Sigmoid, Binary Crossentropy Loss, Adam Optimizer	0.9712	0.8571	0.8411

5.2 Time Series Forecasting Results

For the state of Virginia as a whole and at the county level, the LSTM model was able to learn the trends of the crashes over time very well with test set Mean Squared Errors (MSE) ranging from 0.5%-1% (Table II). While the model could learn the trends well, the peaks and valleys on specific days were not captured and generalized in testing (Figs. 7, 9, 11, 13, 15, and 17). The general trends are helpful to plan around, but we cannot isolate the specific days where it is safest or riskiest to travel. This is reflected in the future prediction graphs for the state and the counties, with the shape of the graph following the pattern of the prior data but with smaller magnitudes (Figs. 8, 10, 12, 14, 16, and 18). Based on the observed trends, the prediction results show seasonality trends, with holiday and break travel times having spikes in the number of crashes and then dipping back down during periods of mainly commuter traffic.

Table II: Time Series Forecasting with LSTM

Location Scope	Training Days	Testing Days	Mean Squared Error
Virginia State	2482	621	0.0057144
Fairfax County	2482	621	0.0103103
City of Virginia Beach	2482	621	0.0104179
Henrico County	2482	621	0.0057555
City of Richmond	2482	621	0.0083323
Chesterfield County	2482	621	0.0054106

6 Conclusion

By analyzing a dataset from the Virginia Department of Transportation, our study aims to predict fatalities in car crashes based on environmental, social, and geographic factors. In the realm of classification, our research employs machine learning algorithms such as random forests, logistic regression, and Artificial Neural Networks (ANNs) to predict accident severity. We used techniques for addressing imbalanced classes in the dataset and implemented oversampling techniques, such as the Synthetic Minority Oversampling Technique (SMOTE), to enhance the performance of our predictive models.

The findings of this research offer concrete suggestions and solutions for both drivers and the Virginia Department of Transportation (VDOT) to enhance road safety and mitigate the risk and severity of car accidents. VDOT can address the risk factors we identified through targeted infrastructure improvements like adding lights to increase driver visibility and preparing roads to deal with adverse conditions like snow or rain. Other solutions include dynamic traffic management based on real-time risk assessment, public safety and communication initiatives, and data-driven strategies for deploying resources effectively. For individual drivers, the factors we identified can help create safety metrics for route planning and real-time warning systems.

The other major component of our project was using LSTM-based time series forecasting for identifying trends in the number of crashes per day and extrapolating crash numbers for future dates. Sequence generation techniques for training the model and predicting future values employed a sliding window approach where 10-day sequences are used as input features with the subsequent day being the target. The LSTM approach shows the potential of such techniques to capture both short-term and long-term trends at the state or county level.

Based on the forecast trends, drivers can plan their routes around when the crash trends dip down. If traveling during peak crash times is unavoidable, drivers can take extra caution and emergency services can increase staffing levels to heighten response preparedness and save lives when every minute counts. In the long term, infrastructure projects can be planned around the locations that need the most attention to help reduce crash severity.

The primary shortcomings of our experiments are related to the amount of data we have. To make multiclass classification feasible in future work, more data is needed in the categories besides "Property Damage Only (No Apparent Injury)" for the model to have more examples of severe and fatal crashes and distinguish between severity levels. Also, more data would allow for greater geographic precision to map out the travel risk for individual roads. Lastly, additional data would also allow for more granular time series analysis for predicting crashes in a zipcode and even at a particular intersection or section of highway.

References

[1] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321–357, jun 2002.

[2] Qi Tang Lulu Ge, T. Hutcherson and Quanquan Gu. Mixed linear modeling techniques for predicting fatalities in vehicle crashes. *Systems and Information Engineering Design Symposium (SIEDS), Charlottesville, VA, 2017*, pages 353–358, 2017.

[3] Safari M, Alizadeh SS, Sadeghi Bazargani H, Aliashrafi A, Shakerkhatibi M, and Moshashaei P. The priority setting of factors affecting a crash severity using the analytic network process. *J Inj Violence Res. 2020 Jan;12(1):11-19*, oct 2019.

[4] Dennis L. Wilson. Asymptotic properties of nearest neighbor rules using edited data. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-2(3):408–421, 1972.

7 Appendix

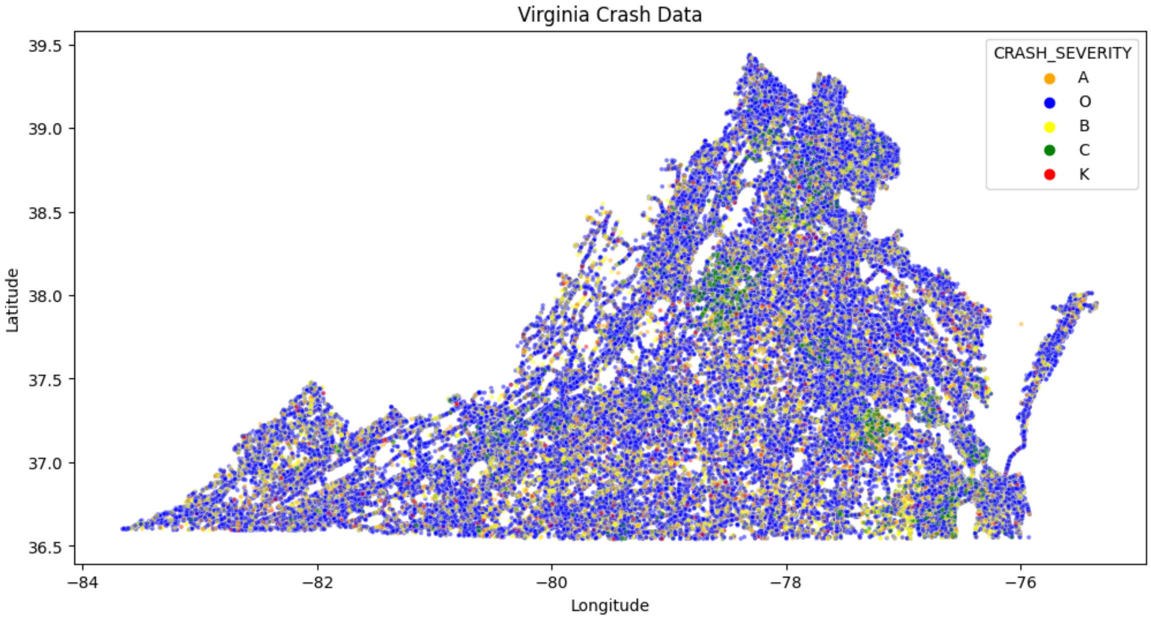

Figure 1: Distribution by Crash Type in Virginia 2015-2023

K	Fatality
A	Suspected Serious Injury
B	Suspected Minor Injury
C	Possible Injury
O	Property Damage Only (No Apparent Injury)

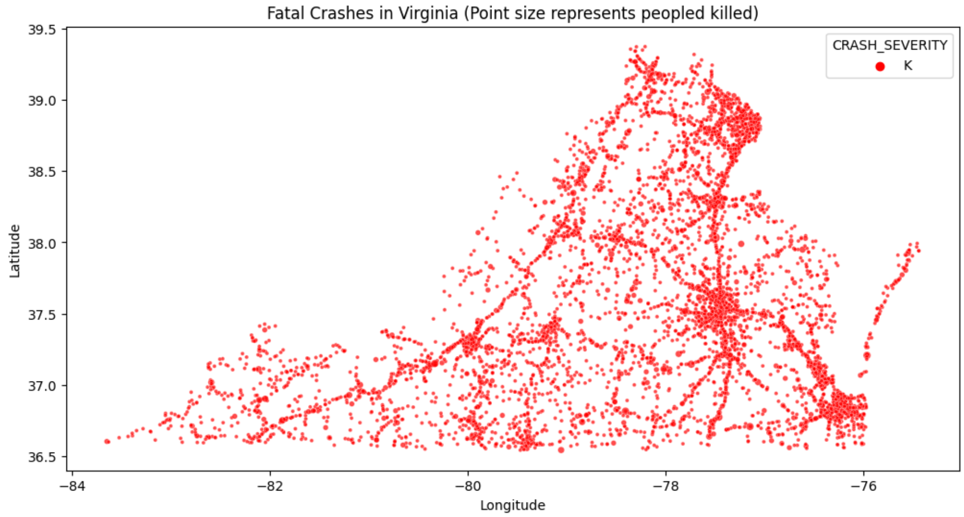

Figure 2: Fatal Crashes in Virginia

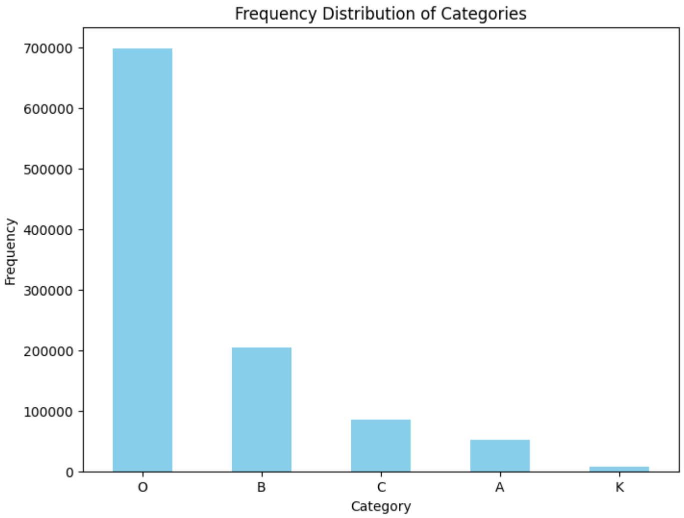

Figure 3: Distribution of crash severity classes in VDOT data

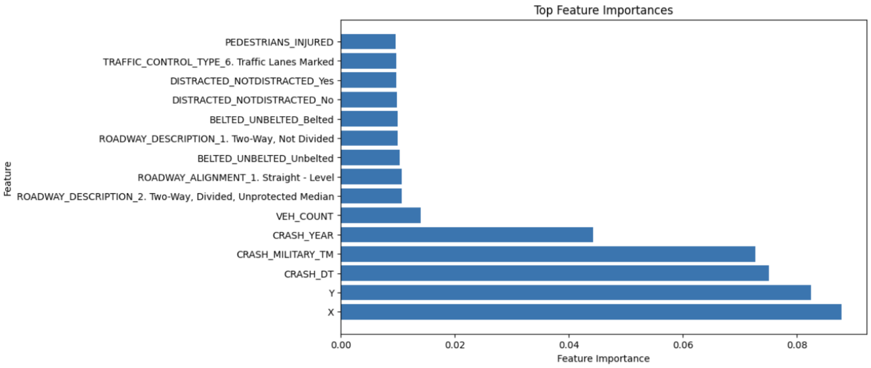

Figure 4: Importance of various features in predicting crash severity

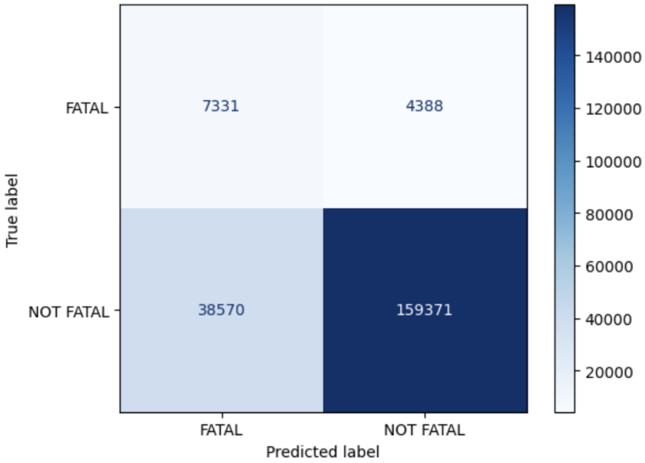

Figure 5: Confusion Matrix for binary ANN classification with SMOTE ratio set to 0.9

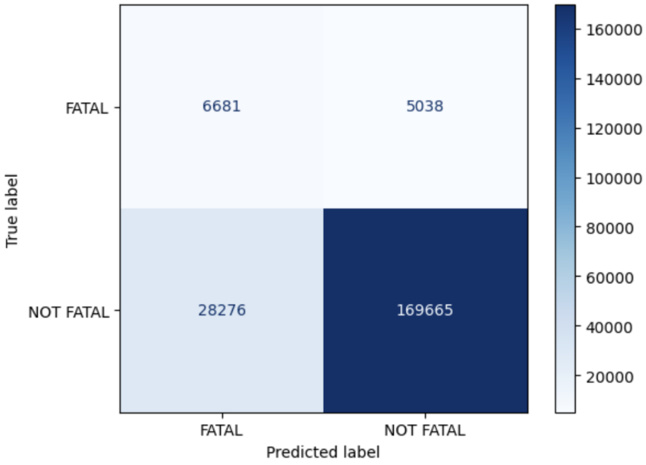

Figure 6: Confusion Matrix for binary ANN classification with SMOTE ratio set to 0.6

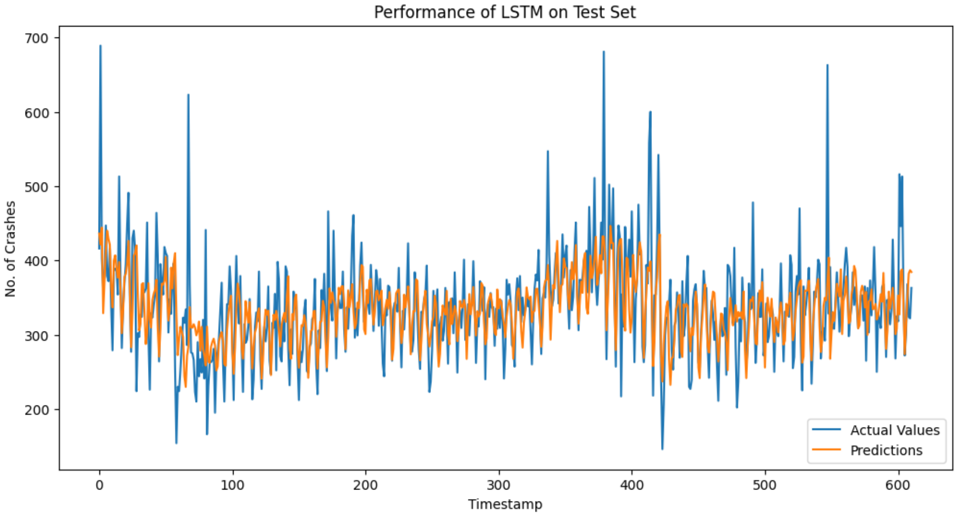

Figure 7: Performance on Test Set of LSTM Regression Model on Virginia Statewide Number of Crashes Per Day

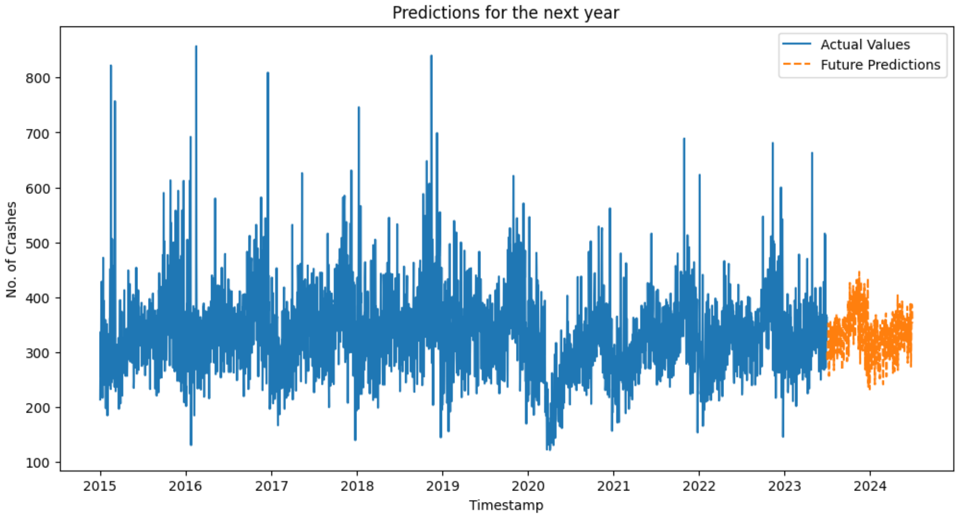

Figure 8: Prediction of Statewide Crashes Per Day for the Next Year

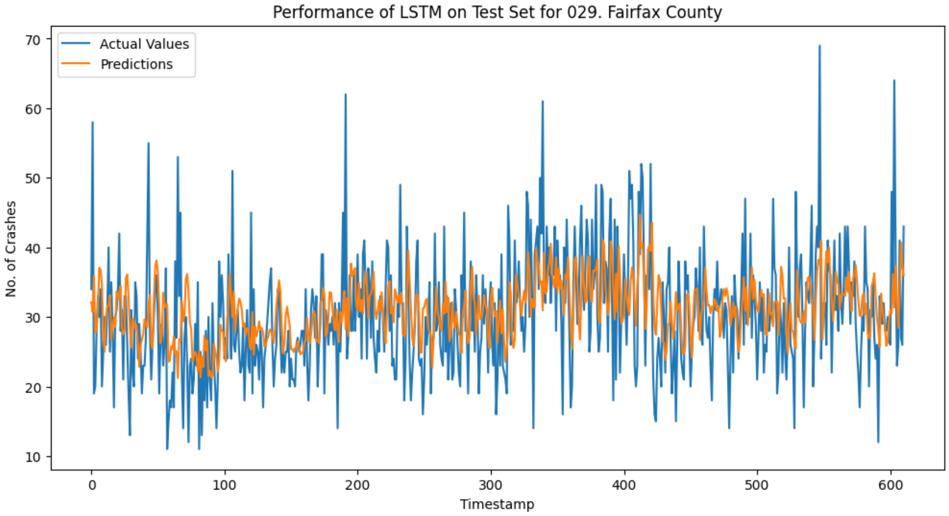

Figure 9: Performance on Test Set of LSTM Regression Model on Fairfax County Number of Crashes Per Day

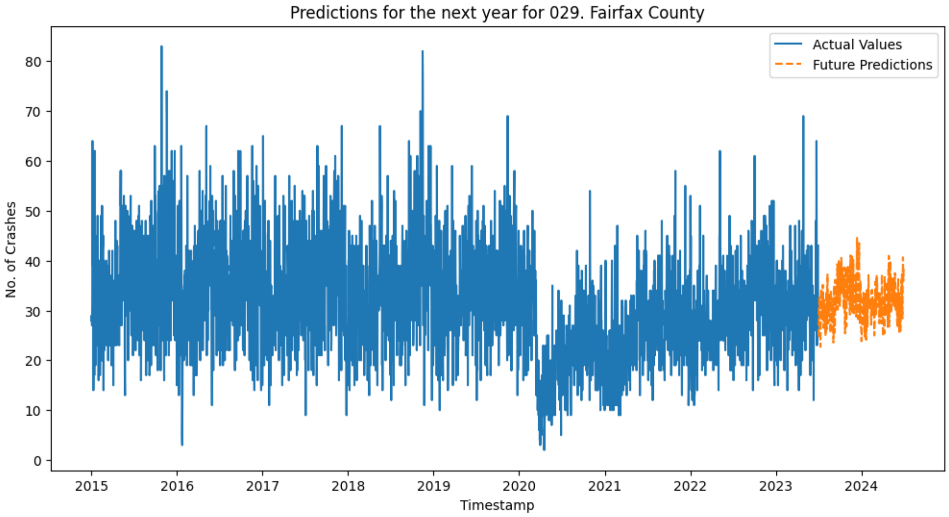

Figure 10: Prediction of Fairfax County Crashes Per Day for the Next Year

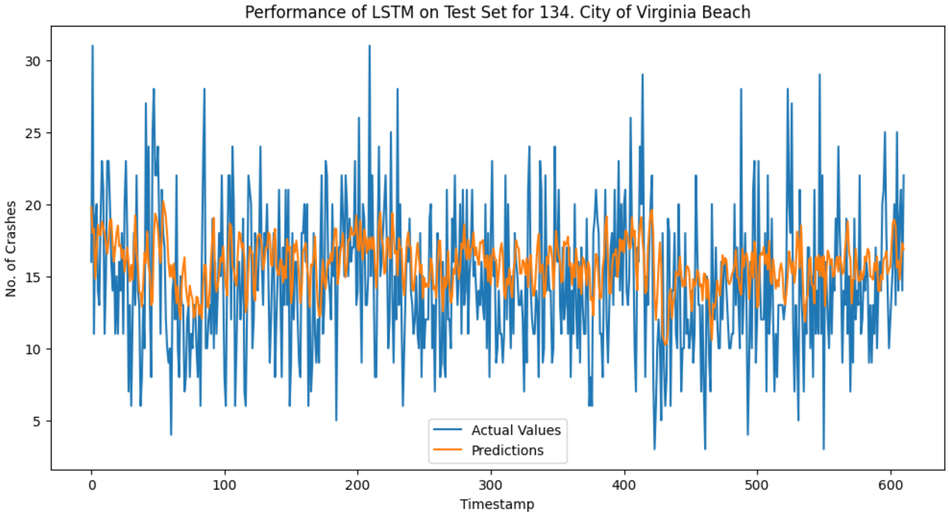

Figure 11: Performance on Test Set of LSTM Regression Model on City of Virginia Beach Number of Crashes Per Day

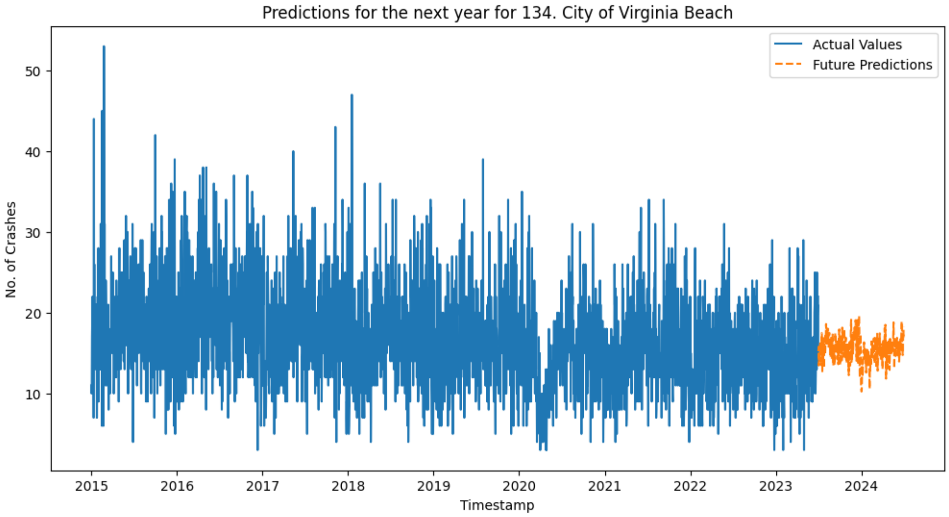

Figure 12: Prediction of City of Virginia Beach Crashes Per Day for the Next Year

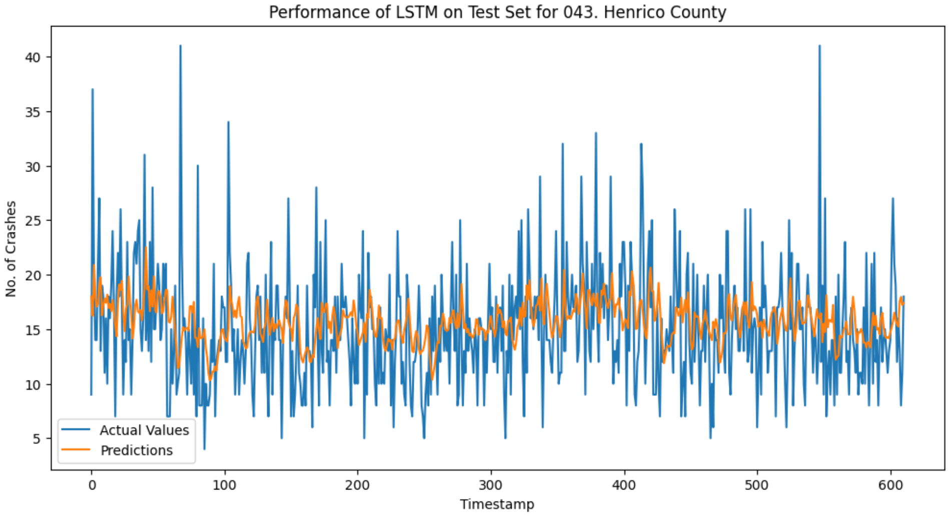

Figure 13: Performance on Test Set of LSTM Regression Model on Henrico County Number of Crashes Per Day

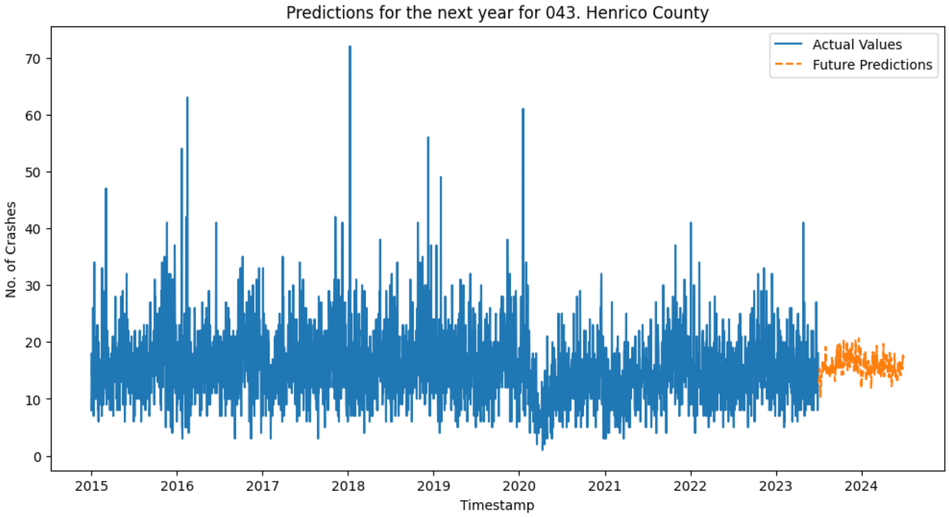

Figure 14: Prediction of Henrico County Crashes Per Day for the Next Year

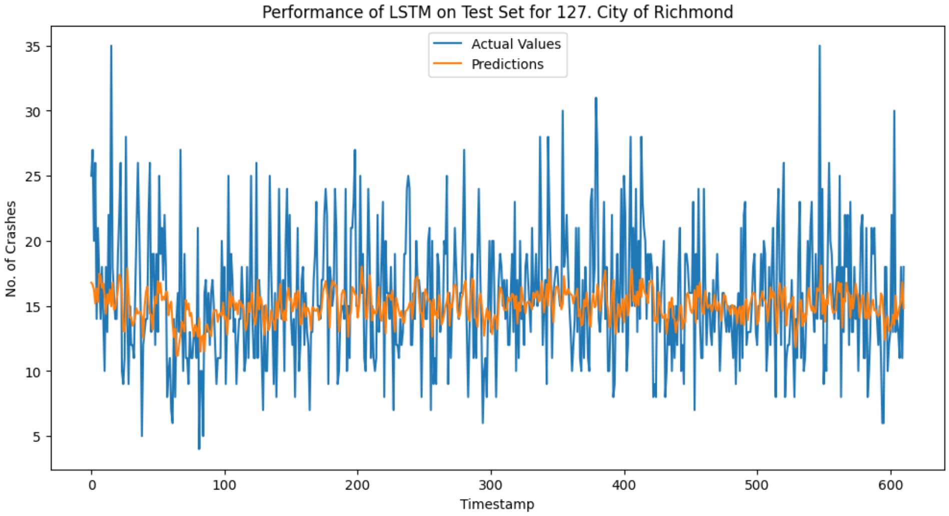

Figure 15: Performance on Test Set of LSTM Regression Model on City of Richmond Number of Crashes Per Day

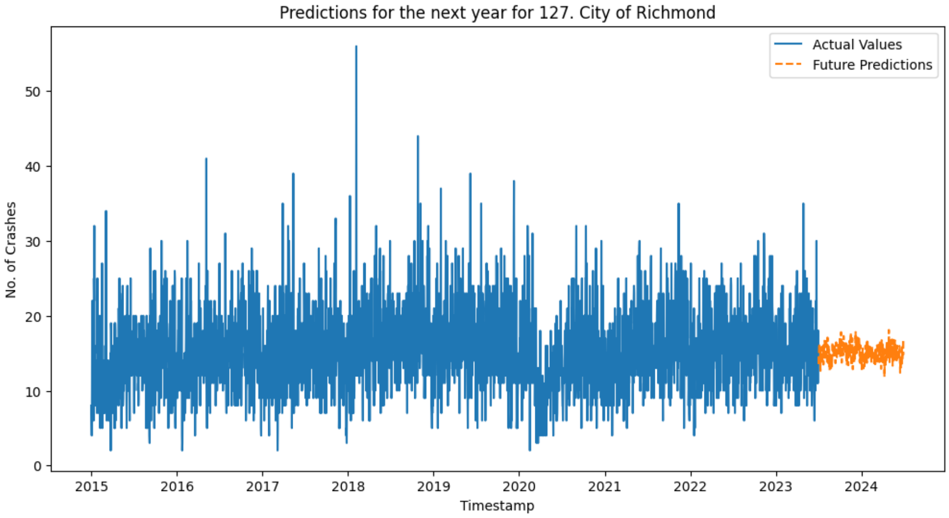

Figure 16: Prediction of City of Richmond Crashes Per Day for the Next Year

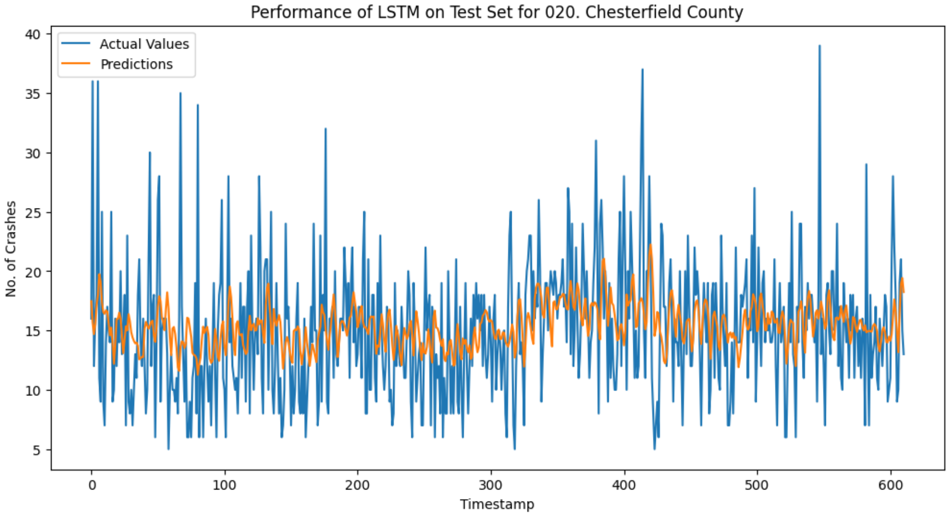

Figure 17: Performance on Test Set of LSTM Regression Model on Chesterfield County Number of Crashes Per Day

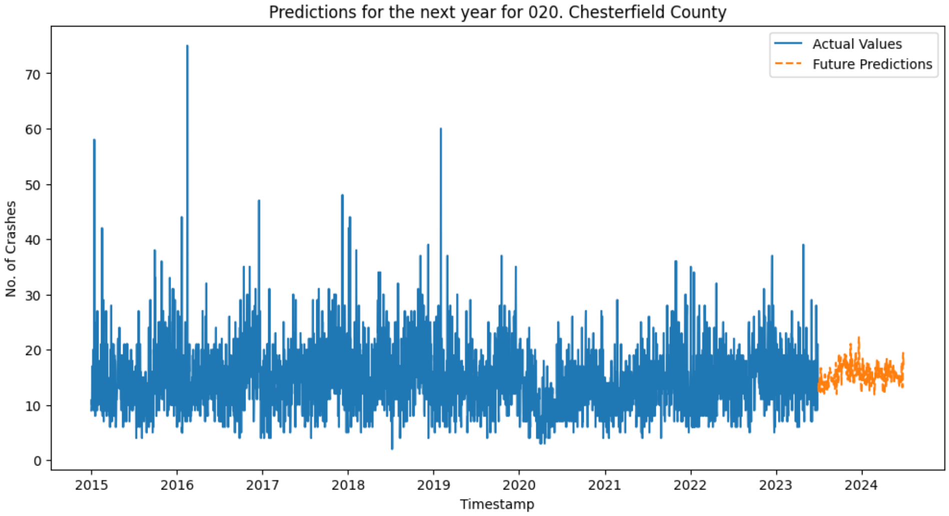

Figure 18: Prediction of Chesterfield County Crashes Per Day for the Next Year