Location and severity Prediction of the accidents in the US

Farimasadat Miri
Ontario Tech University
farimasadat.miri@uoit.ca

***Abstract*—Accident prediction and location of the accident prediction has been one of the hottest topics in recent years, as many companies such as insurance companies and emergency services can be informed in advance to help the people in different areas where the chance of accident is actually higher than the other places. In this paper, we tried to predict the severity of the accident in one year and then predicted the number of accidents in different locations through using ARIMA model on a daily timeseries data. Our work showed that univariate ARIMA model works really good in the regions that have large number of data and worked fine in the areas with lower number of accidents.**

***Index Terms*—ARIMA, machine learning algorithms, K-NN, Random Forest, Decision Tree, Logistic Regression, Univariate, Multivariate, Stationary, Non-stationary, Lazy learning, eager learning, classification, regression.**

I. Introduction

One of the challenges in recent years is predicting the accident through different models in a city to estimate how many accidents do we have in a certain date in a certain region. Collecting data through time series manner can help us to analyses our data in different models. Based on the data characteristics, we can apply different models to predict the number of accidents. Aside from data characteristics, one the most important things in data analysis especially time series data is having real and acceptable data that is gathered over the time. There are many advantages that are coming after accident and location prediction in a city which are undoubtedly useful for the police stations, emergency services, hospitals and even insurance companies. By predicting the number of accidents in different regions of a city, police stations would know in what locations they should be present in advance, emergency services would prepare themselves to deal with the number of injured people and even insurance companies can make money based on having different scenarios in different parts of the city. One of the big challenges in analyzing time series data is finding dependencies between different attributes to having a good accuracy in our prediction. The problem which might come arise here is finding the correlation between the number of accidents and each region based on different factors such as weather conditions, temperature, different traffic signage, day time or night time and etc. finding the correlation between each columns to understand which one has more effect on the other can help us to filter our data and just work on correlated data to have better prediction in severity of the accidents as well. This paper first tries to visualize data to understand the correlation between each column and then based on the correlation of columns, prediction of severity is done through using different machine learning algorithms (K-Nearest Neighbors, Decision Tree, Random Forest and Logistic Regression). After prediction of accident severity, we are trying to predict the number of accidents in each proposed region in California through using ARIMA model. Our time series data has been collected in the US from 2016 to 2019 which has 49 columns and has the number of accidents in different locations through displaying latitude and longitude of the location. The rest of this paper is organized as follows. In the next section, we are trying to explain the necessary materials about what is time series data and what is ARIMA model. Then we talk about four different types of machine learning algorithms that we want to use them for severity prediction of the accident. Afterward, we show data visualization and the correlation between each column. Then, we discuss our methodology. In the next section we showed the results and finally we go into conclusion.

II. Background and related work

1) Introduction to timeseries forecasting: Time series is a sequence of time that considers one or several metric over time intervals. Based on our goal and our dataset we can have different frequencies for time series such as : yearly, monthly, weekly, daily or even hourly or minutes (like prediction of prices minute by minute in DowJones). Having timeseries data in each field helps us to forecast the future values of the given data which has a high commercial value for every organizations and companies [1]. One of the pressing aspects that have an important effect on timeseries is lags or time periods meaning it shows the patterns within the timeseries and shows us how much our timeseries correlates with itself [2] [3]. when the periodicity of the data are low or high we have a component which is trendy and we should deal with it to resolve trendy issues [4] [5]. There are some algorithms that can predict through timeseries. However, the main one named Auto-Regressive Integrated Moving Average (ARIMA) [6] [7] It has been used in different domains such as traffic noise time series by Kumar and Jain [8] and fuel demands in Turkey by Ediger and Akar [9]. Likewise, there are two different types of timeseries prediction through ARIMA model (short of Autoregressive Integrated Moving Average) namely: Univariate Time Series Forecasting and Multi Variate Time Series Forecasting. by having past value, we can predict the

future value by ARIMA model through dividing our model to training and test groups.

A. Arima Introduction

The important thing for using Arima model is having a non-seasonal and non-trendy data set. In the seasonal data we are seeing some patterns over some lags which is not random. The other important thing about data is not having upward or downward trend when you want to apply ARIMA model on it. For better understanding, ARIMA has 3 different terms [8]: p shows us the order of autoregressive model, d can be defined as how many times we need to have differencing for having a stationary data, in other words, how many times we need to difference the data for changing data from non stationary to stationary and q which is the order of MA and in some papers it can be defined as size of the MA window. There is a linear regression model in ARIMA which named AR. In order to have prediction through ARIMA, we have a linear regression model that use some lags for doing that. If the predictors are not correlated to each other and have an independent behavior toward each other Linear regression works in a best way. There are different methods to make a time series stationary such as differencing and doing log on our data. We can have several times differencing on our time series. For example if d=1 it means that we did one time differencing on our data or if d=3 it means that we did 3 times differencing on our data, the number of differencing really depends on when we can change data to stationary. If p=3 it means that the number of lags in the data is three in other words 3 lags can be used as predictors. If q=4 it means that our number of lagged forecast errors in our prediction model is equal to 4. The mathematical formula for AR model is this :

$$Y_t = \alpha + \beta_1 Y_{t-1} + \beta_2 Y_{t-2} + .. + \beta_p Y_{t-p} + \epsilon_1$$

In formula that showed above Yt-1 is first lag of our time series and 1-p shows how much this lag is coefficient. is the slice term which can be tested through ARIMA model in different phases. Moreover, the mathematical formula for MA has been shown in equation 2 which Yt is only dependent on the lagged forecast errors and t is defined as error terms. Overall, the equation of ARIMA model has been shown in

$$Y_t = \alpha + \epsilon_t + \phi_1 \epsilon_{t-1} + \phi_2 \epsilon_{t-2} + .. + \phi_q \epsilon_{t-q}$$

equation 3. As it is clear, we have put together AR and MA equations and having at least one time differencing to make the data stationary is necessary in ARIMA equation as well.

$$Y_t = \alpha + \beta_1 Y_{t-1} + \beta_2 Y_{t-2} + .. + \beta_p Y_{t-p} \epsilon_t + \phi_1 \epsilon_{t-1} + \phi_2 \epsilon_{t-2} + .. + \phi_q \epsilon_{t-q}$$

The question might come arise here is what is the best way for identifying the values of p,d and q? For getting the value of d we should make sure that we have the minimum differencing to get our data stationary. Its important to note that the time series data set needs more differencing if for large number of lags, we have positive autocorrelation. Through using ADF which is short form of Augmented Dickey Fuller we have a function named adfuller that is already exist in the package of statsmodel in python and allows us to evaluate our time series stationary or not. Our assumption at first is that our time series is not stationary. Through this hypothesis, if p ¡ 0.05 then our data is stationary and the first assumption is going to be declined. However if p¿0.05 we define data as non-stationary and try to find the best value for d to having stationary data. its worthy to mention that sometimes through some plots you can see how many times is enough for differencing in our data. For instance, in autocorrelation plot if the first plot goes into a negative field, we need to decrease the number of differencing that we did in our time series. Another plot is ACF plot to help us delete autocorrelation in our time series data which is considered non stationary through using MAs . Its important to know about handling a timeseries data which is under or over differenced. For under differenced time series, adding one or more additional Auto Regressive (AR) terms helps the model to come up. And for the slightly over-differenced time series, adding one additional Moving Average (MA) term can fix the problem. For finding the optimal ARIMA model. There is a need to have training and testing data in two parts in the portion of 70:30 or 75:25 for predicting our time series into the future. In order to understand how much your accuracy is good enough or not you need to evaluate your model through some metrics. The commonly used accuracy metrics are (MPE wich is short form of Mean percentage Error, RMSE which is short form Root Mean Square Error) and the other one is ME which is a short form of Mean Error)For comparing the prediction between two different timeseries data, MAPE and Min-Max are used as they are between 0-1. Through this percentage error you can evaluate how much your prediction has accuracy. Its important to note that all of the above metric are not good for comparing accuracy between two different timeseries as they are quantities which means that RMSE of 100 for a series with 3000 mean is better than RMSE of 20 for series with 40. There are some attempts in the scope of traffic prediction on small scale dataset despite the fact that the given data is stationary but the ARIMA model that was used was computationally intensive and did not provide an acceptable result [10] [11].In [12] the authors showed a technique named carlo model for prediction of traffic into timeseries data, it has some problems as well such as applying the model on just peak hours and rush hours during the day to have a stationary data rather than non-stationary. There are many researches in the scope of traffic prediction in large scale dataset and small-scale dataset but there are a few numbers of papers that have worked on location prediction of the accident on timeseries data which didn't show the obvious and acceptable results.

1) K-nearest neighbors: K nearest neighbors is one of the simple algorithms for classifying the data. For example, if we have different types of cells such as fat cells, or stem cells or cancer cells, through using this algorithm we can identify that

which one of these a person is. If we want to groups our action into different steps we should firstly start categorizing our data based on the name of the unknown categories. For example, regarding to our different cells we have 3 three different groups or clusters. In the next step, as we don't know what categories our new data goes into we have an unknown category. And finally, we are going to group our new cluster by looking at the nearest neighbors. There are some options here, for example if the number of neighbors that are close to our unknown cluster is equal to one, we put our new cluster into that cluster. In other words, we describe that unknown cluster into a known cluster which is nearest to itself. However, if our unknown cells are placed between 4 or 5 different cells, we need to take into consideration which of them has more numbers. For instance, imagine we have 5 different clusters and each of them has different number of members. In this case our unknown cells go into a cell that has higher members. The question that comes arise here is: what the value of k can be? Or what is the best value for k? through using different values for k, we can figure it out which one is the best one for using. The other way is we can set some parts of our training data as unknown data and then test the best value for k. But, the important thing is some values such as k equals to one or two in most cases are not good enough because of having outliers. On the other hand, having large values of k cause even a cell with a few members be selected by other cells or groups. For more clarification, in figure two as you see, we have a two-dimensional dataset and having two classes which are blue and red classes. All these two classes are our training examples, and green box is also my testing examples. As our green box is really near to red ones and far from blue ones we categorize the green circle into red group. But now imagine our green circle is put into the same distance between red and blue ones. In this case, we cannot only rely on the Ellucian distance between green circles and blue and red groups but we should take that which region in space is closer to our green circle into consideration and in this case as shown in figure 2, the red triangles are closer in terms of region to our green circle. The decision boundary tells us classifier learns this line that separate one class from the other class and in the K-NN case, it can be a broken line through following Voronoi cells in space where all the cells in one side of the decision boundary are dominated by blue class and the other side are dominated by the red class. Compared to the other methods such as Naïve based, this method is very flexible and can be overfitted very well with complicated boundary. But as we talked before, this algorithm is sensitive to outliers and just one single mislabeled example can change the boundary in a complete way. The other disadvantage of K-NN is being insensitive to class prior. For resolving this issue, we can have more than one nearest neighbor for making decision that makes our classifier a little bit more stable and smoother which is not being infected by just a little change in a data point. We have two terminologies named lazy learning and eager learning in which K-NN can be defined as lazy learning as it distributes data into the testing phase which is completely the opposite of eager learning which distributes the data into training phase.

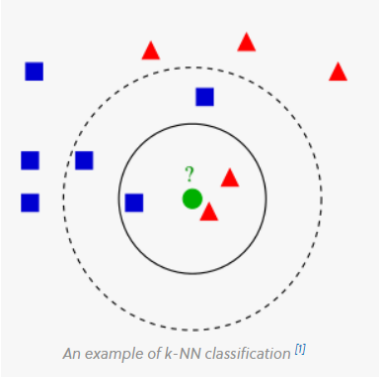

Fig. 1. An example of k-NN classification

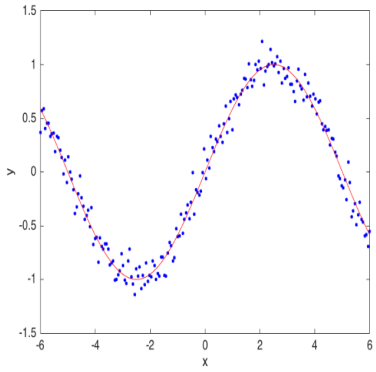

Fig. 2. An example of data points from a sine wave

B. Decision Tree

Decision tree is one of the supervised machine learning algorithms which is being used in classification and regression problems. We can define a decision tree as a flowchart where each node shows a test on a feature and each branch shows the outcomes of a test and each leaf hold a class label. And the top most node in a tree is named root node. There are some advantages regarding to this model such as we can visualize it which is really better for understanding the dataset through it. Moreover, decision tree can manage categorial and numerical data. Let's consider one example for more clarification, suppose we have a dataset which wants to predict a person named Zoe to figure out in what day she is going to play tennis. And imagine our dataset is composed of different attributes such as the days, type of weather (which is sunny, rainy, overcast and etc.), type of humidity (which is normal or high) and type of the wind (weak or strong) and finally whether the person plays or not in different types of features in different days. For example, sometimes, she doesn't want to play with strong winds and high humidity and sometimes in different days she prefers to play tennis with the same situation that she experienced before. What decision tree does is taking a look at one of these attributes and categorizing or spliting them

into subset that can be named as divide and conquer algorithm which evaluates that are they pure or not? And if they are pure it stops the analysis and if not, it repeats that over and over again to see which subset our new data falls into. For instance, imagine there are some subsets that under the humidity and high wind condition the player goes into playing tennis one day and the other day she is not going to playing tennis. For more clarification you need to again split this subset again to have a pure answer like yes or no. So as it is obvious, you have taken out your training set and you have sorted out it into pure subsets based on some attribute values. All in all, decision tree is like a logical formula that tells you in what cases a player plays a tennis and in what cases she does not play. In figure 3 we are seeing the simple example about the students who want to play Crickets in different conditions or not. As it is clear, we have data splitting into different subsets to get the pure response such as yes or no. in figure 4 we see pruning, meaning the size reduction of tree through converting some branch nodes into leaf nodes.

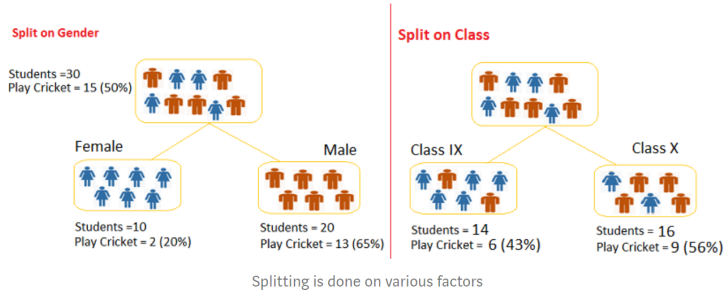

Fig. 3. splitting in decision Tree

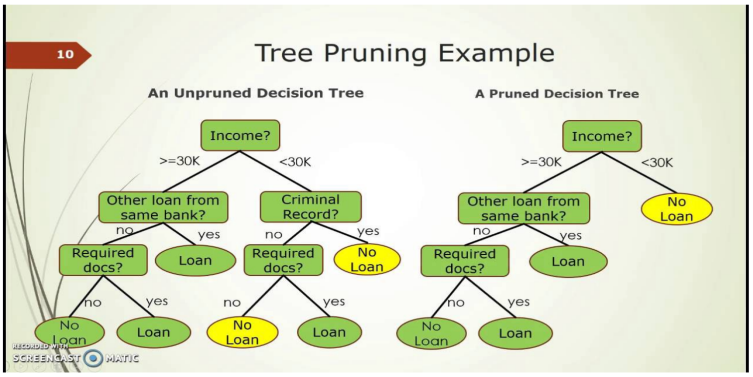

Fig. 4. Pruning in decision Tree

C. Logistic Regression

This algorithm can dissolve a classification issue. Linear regression and logistic regression both follow the same idea. Let's take an example for more clarification about what is classification in real world.For instance, we can classify the people as thin or fat or we can classify the houses as old or new. Classifying the job as high paid jobs or low paid jobs can be another sample of classification. As it is clear all of the examples are categorial types as we have only two class classification problems. For dealing with outliers logistic regression uses the function named Sigmoid that only uses the values between 0-1. In figure 5, plot of this function has been

$$Y_t = \alpha + \beta_1 Y_{t-1} + \beta_2 Y_{t-2} + .. + \beta_p Y_{t-p} + \epsilon_1$$

shown and in order to resolve the outlier problems as shown in figure 6 the y axis moves right or left. Changing the y axis completely depends on the concentration of outliers. It is defined in equation 4 as follows :

When we plot this function, the graph is like a S curve.

$$p(x) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x)}}$$

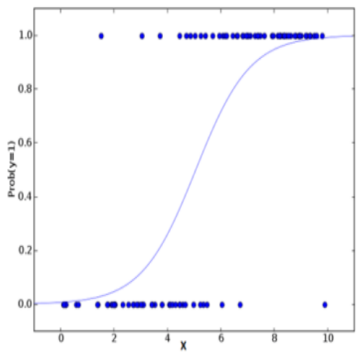

Fig. 5. Sigmoid function plot

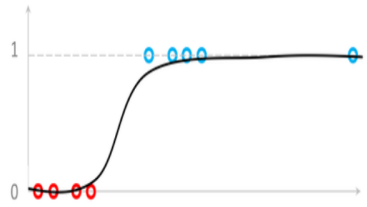

Fig. 6. Dealing with outliers

III. Methodology

In this section we try to show that what we have done in our data. First thing we need to know about our data is finding the correlation between columns and missing values. After visualizing the data, we are trying to clean the data and then applying four different machine learning algorithms to forecast the severity of the accidents.

A. Data Visualization and Experimental Setup

Our data set consists of 49 columns which has the accident information from 2016-2019 in the US. Figure 7 shows 49 columns.

```
Index(['ID', 'Source', 'TMC', 'Severity', 'Start_Time', 'End_Time',
       'Start_Lat', 'Start_Lng', 'End_Lat', 'End_Lng', 'Distance(mi)',
       'Description', 'Number', 'Street', 'Side', 'City', 'County', 'State',
       'Zipcode', 'Country', 'Timezone', 'Airport_Code', 'Weather_Timestamp',
       'Temperature(F)', 'Wind_Chill(F)', 'Humidity(%)', 'Pressure(in)',
       'Visibility(mi)', 'Wind_Direction', 'Wind_Speed(mph)',
       'Precipitation(in)', 'Weather_Condition', 'Amenity', 'Bump', 'Crossing',
       'Give_Way', 'Junction', 'No_Exit', 'Railway', 'Roundabout', 'Station',
       'Stop', 'Traffic_Calming', 'Traffic_Signal', 'Turning_Loop',
       'Sunrise_Sunset', 'Civil_Twilight', 'Nautical_Twilight',
       'Astronomical_Twilight'],
      dtype='object')
```

Fig. 7. Name of the 49 columns in our data set

After, we can start looking for relations between the data. For example, let's take a look at the amount of accidents per state in Figure 8. As its clear CA has the highest number of accident compared to other states in the US. After finding out

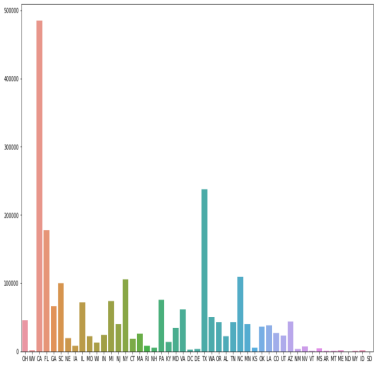

Fig. 8. Number of accidents per states

the number of accidents per state we are going to compare the severity levels in each state as it is shown in figure 9, there are 4 different severity levels which one of them is not really clear in the plot. Then we are going to explore whether the temperature is a key factor that increase the severity of the accident or not ? as it is shown in the figure 10, the difference are really small but, there is still a few correlation between temperature and severity. In the figure 11, we can observe that the most accidents happened when the weather was clear. It might have a reason which is maybe the people have more tendency for driving in a clear weather and the other reason is they might be careless in driving in clear weather compared to the time when the weather is rainy or snowy that they try to be more careful while driving.

In the figure 12 and 13 we see that Most of the accidents happened during the day not night especially in rush hours both in the mornings and afternoons of work days from 7 till

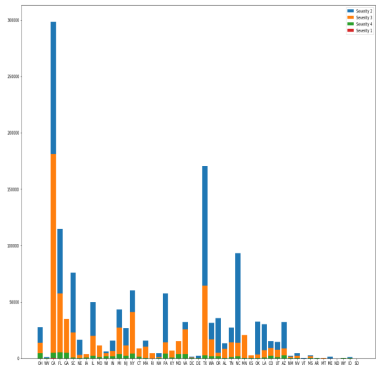

Fig. 9. Different severity levels comparison

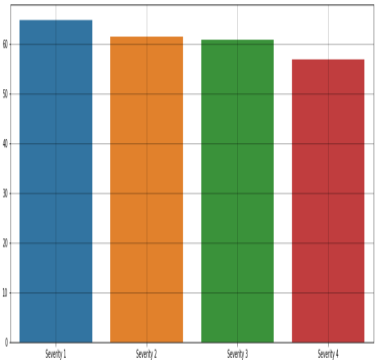

Fig. 10. temperature severity of the accident

9 AM. At weekends there are a few accidents and most of these accidents are occurred from 12:00 AM till 14:00 PM.

B. severity prediction

In this section we are using 4 different machine learning algorithms to forecast severity of the accidents in CA. in order to do that we are following these step to clean the data and deal with missing values and categorial data and then apply them on our data one by one to see the results. Step 1. Extract year, month, day, hour, weekday, and time to clear accidents Step

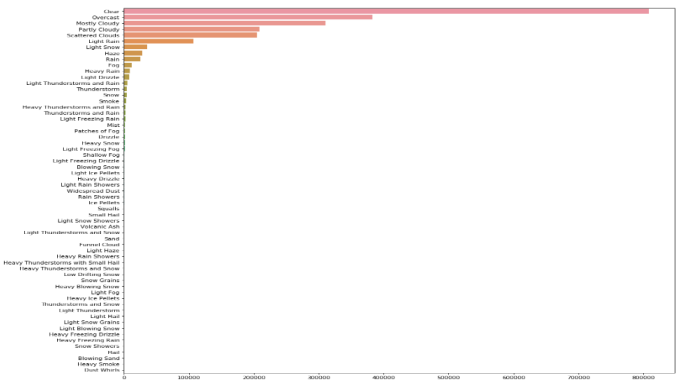

Fig. 11. weather comparison in terms of number of accidents

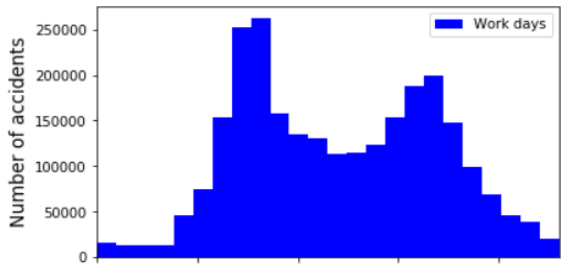

Fig. 12. Number of accidents in work days

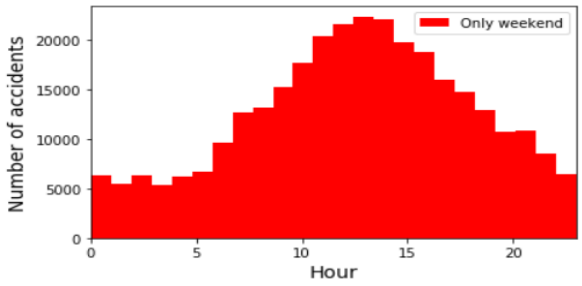

Fig. 13. Number of accidents in weekends

2. Deal with outliers (drop rows with negative timeduration and Fill outliers with median values) Step 3. Select a list of features for machine learning algorithms(only select relevant columns without putting more intensive computation on the computer). Step 4. Drop rows with missing values Step 5. Select the state of interest: CA Step 6. Deal with categorical data: pd.getdummies() Step 7. Predict the accident severity with various supervised machine learning algorithms

C. Using ARIMA to predict the future accidents in each region per day

In this section, we want to know about number of accidents in a specified region in a certain time. In order to forecast the number of accidents in one location in a certain time, we need to grid the map of California. As we can have small regions in our map and each one has its own accidents. In the real world, in order to explore the number of accidents in one region, it's important to know about this accident happens in what region? Suppose that a police officer wants to predict about the number of accidents in one specific location in the map on Monday. Firstly, he divided the map into smaller regions and then based on the previous accidents that have happened in that region, he estimates the number of accidents in coming Monday. We can divide our map into 10*10 or 100*100 or maybe more. It depends on our goal. But, its important to know that if we grid our map to 10*10 the dependency between the regions are too weak or none. There is major distinction between the accident prediction and traffic prediction in time series data. In traffic prediction, usually one street is highly correlated to the other streets, if one highway is congested, there is a high probability that the other neighboring regions might be congested. However, in accidents we cannot correlate the number of accidents in one big region to the other neighboring regions. But, if we grid our map into smaller regions these dependencies go stronger. Suppose that you divide a highway into 100 smaller regions, if one accident happened into one small area there is a high chance that the other accidents happen before that location as the cars don't know about accident event in that location. In our work we grid the map of California into 100 regions and collect the number of accident in each region and then apply ARIMA model on our timeseries data which is daily. Our training dataset is 70 percent from 2016-2018 and our testing data includes 30 percent of data which is the data of 2019. Figure 14 shows the number of accident in each region of California.

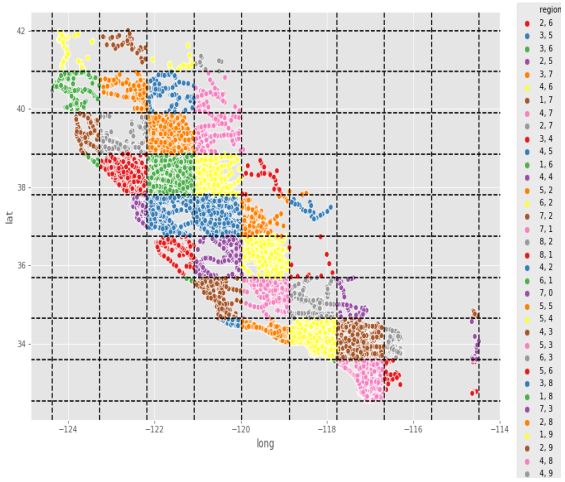

Fig. 14. Gridding California 10*10

In order to explore whether our data is stationary or not we plot our data to finds out it has seasonality or trend or not. As you see in the figure 15, we have both stationary and trend in our timeseries and through using two function named kpsstest() and ADFtest() we test our data to make sure whether its stationary or not.

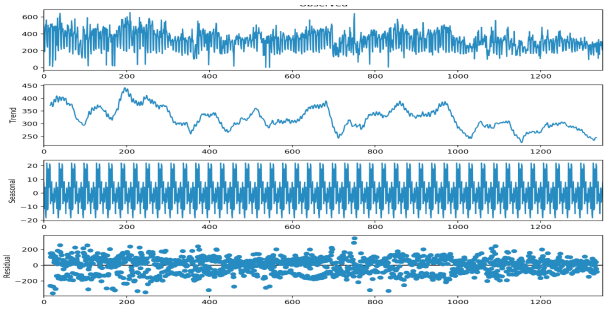

Fig. 15. data plot

As its obvious in both tables1 and 2, one of them clearly shows that our data is not stationary which is table1 and the other one contradicts the first one. For making sure that our

TABLE I
RESULT OF KPSS FUNCTION IN ONE TIMESERIES OUT OF 100 OF CA

Test Statistic	0.137693
p-value	0.100000
Lags Used	23.000000
Critical Value (10%)	0.347000
Critical Value (5%)	0.463000
Critical Value (2.5%)	0.574000
Critical Value (1%)	0.739000

TABLE II
RESULT OF ADFTEST() FUNCTION IN ONE TIMESERIES OUT OF 100 OF CA

Is the data stationary?	Yes
Test statistic :	-7.128
P-value :	0.000
Critical values:	
1%:	-3.435739110194116 - The data is stationary with 99% confidence
5%:	-2.863919777127088 - The data is stationary with 95% confidence
10%:	-2.5680370312770515 - The data is stationary with 90% confidence

data is stationary we do differencing on our data through using diff() function and again test our data. After finding out our data is not stationary through those functions again, we apply our ARIMA model on our data for each region. We run ARIMA 100 times for each region and plot the data to see the result of our work. Afterward, we reverse the data which has been differenced to real data to evaluate the approximate number of accidents in each region.

IV. EVALUATION AND EXPERIMENTAL RESULTS

A. Severity prediction results

After applying 4 different machine learning algorithms namely: K-NN, Random Forest, Decision Tree and logistic regression, as its shown in figure 16, we can conclude that random forest has highest accuracy score among all of them and K nearest neighbor has lowest accuracy score between them. This result is just for California.

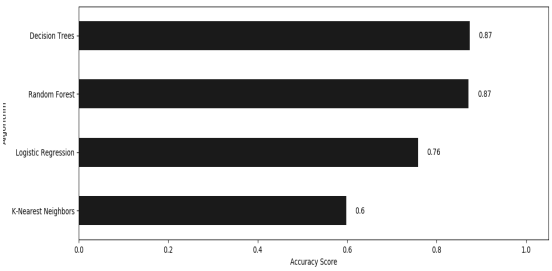

Fig. 16. comparison between 4 different algorithms in severity prediction for California

1) Arima Model Results: For understanding about our results first we used some metric on our data to see how much it has accuracy. Table 3 shows different metrics that have been calculated just for one region out of 100 region and the related plot about that has been shown in figure 17.

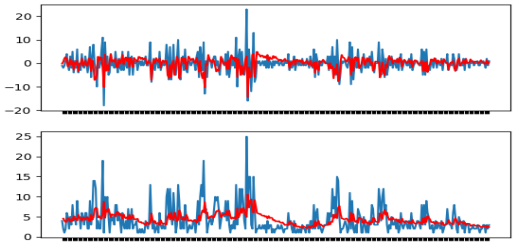

Fig. 17. One region result after applying ARIMA

TABLE III
DIFFERENT METRICS FOR TESTING OUR ACCURACY IN ONE REGION

Test MSE	10.787
Test MAE	2.361
Test RMSE	3.284

There are 100 different plots for 100 different regions which we can put just 10 of them in this paper. All of the figures in each region have 2 plot the first one shows the data prediction which had been differenced before and the other one shows the real data prediction on expected data. As its clear, all of the metric that have been used show us univariate ARIMA model, predicted the number of accidents in each region in a good way. Apart from that, the plots show us how expected and predicted data cover each other. The only minor thing here is, if the number of accidents in each region would not be too large [13], the predicted model is not good enough [14] [15]. In other words, predicted and expected data have not been covered each other, mainly due to the model does not have enough data to be trained more. we did this experiment in 10*10 and 100*100 grids and we had better coverage in expected and predicted in 10*10 in which each region has more number of accidents in timeseries.

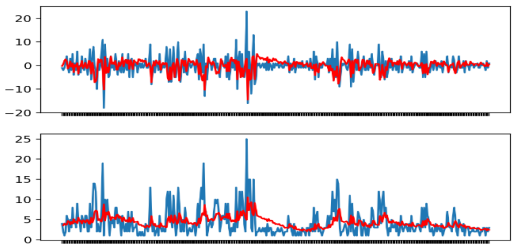

Fig. 18. region 1 result after applying ARIMA

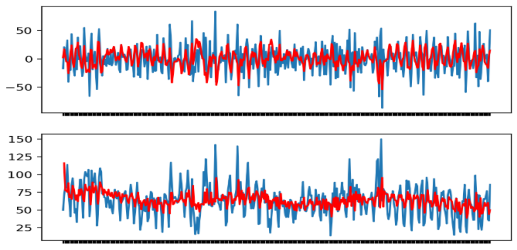

Fig. 19. region 2 result after applying ARIMA

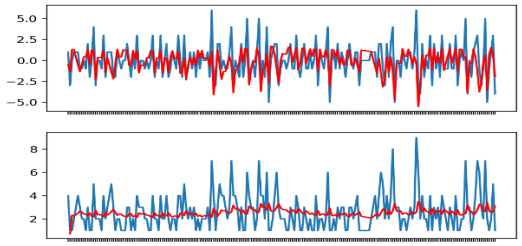

Fig. 20. region 3 result after applying ARIMA

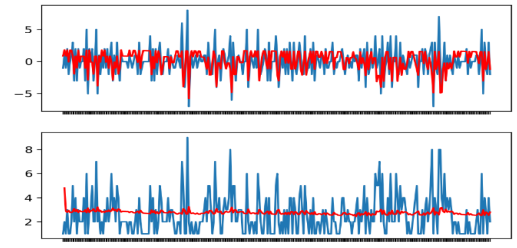

Fig. 21. region 4 result after applying ARIMA

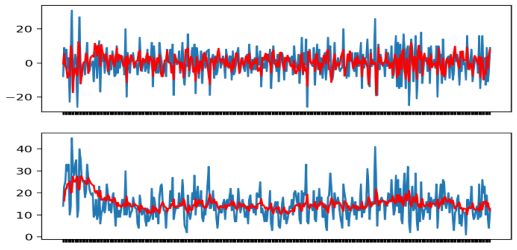

Fig. 22. region 5 result after applying ARIMA

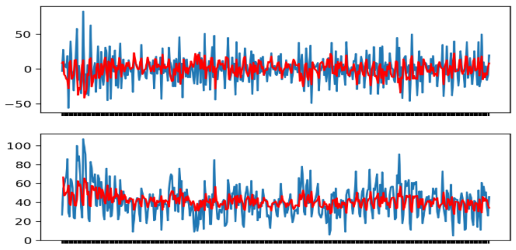

Fig. 23. region 6 result after applying ARIMA

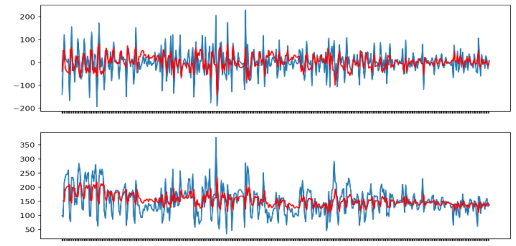

Fig. 24. region 7 result after applying ARIMA

V. Conclusion

In this paper we tried to explain some basic ideas about 4 different machine learning algorithms and why we are using them in severity prediction in accidents. Then we show that K-NN has the highest accuracy among others. After, we used ARIMA model for 100 regions in the map of California to predict how many accidents do we have in 2019 in timeseries data which is daily. Our plots show that Univariate ARIMA model could predict the number of accidents in each region with good accuracy. Our next work is forecasting the number of accidents and severity of the accidents in each region through a daily timeseries.

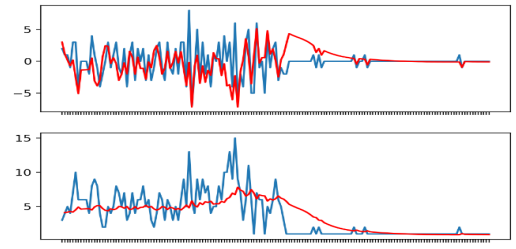

Fig. 25. region 8 result after applying ARIMA

REFERENCES

[1] J. G. De Gooijer and R. J. Hyndman, "25 years of time series forecasting," *International journal of forecasting*, vol. 22, no. 3, pp. 443–473, 2006.

[2] C. W. Ostrom, *Time series analysis: Regression techniques*. Sage, 1990, no. 9.

[3] D. McDowall, R. McCleary, and B. J. Bartos, *Interrupted time series analysis*. Oxford University Press, 2019.

[4] L. Li, X. Su, Y. Zhang, Y. Lin, and Z. Li, "Trend modeling for traffic time series analysis: An integrated study," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 6, pp. 3430–3439, 2015.

[5] M. Mudelsee, "Trend analysis of climate time series: A review of methods," *Earth-science reviews*, vol. 190, pp. 310–322, 2019.

[6] C. Chen, J. Hu, Q. Meng, and Y. Zhang, "Short-time traffic flow prediction with arima-garch model," in *2011 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2011, pp. 607–612.

[7] H. Liu, H.-q. Tian, and Y.-f. Li, "Comparison of two new arima-ann and arima-kalman hybrid methods for wind speed prediction," *Applied Energy*, vol. 98, pp. 415–424, 2012.

[8] N. Garg, K. Soni, T. Saxena, and S. Maji, "Applications of autoregressive integrated moving average (arima) approach in time-series prediction of traffic noise pollution," *Noise Control Engineering Journal*, vol. 63, no. 2, pp. 182–194, 2015.

[9] V. Ş. Ediger and S. Akar, "Arima forecasting of primary energy demand by fuel in turkey," *Energy policy*, vol. 35, no. 3, pp. 1701–1708, 2007.

[10] M. R. Nieto, R. B. Carmona-BenÃtez *et al.*, "Arima+ garch+ bootstrap forecasting method applied to the airline industry," *Journal of Air Transport Management*, vol. 71, no. C, pp. 1–8, 2018.

[11] J. Xin, J. Zhou, S. X. Yang, X. Li, and Y. Wang, "Bridge structure deformation prediction based on gnss data using kalman-arima-garch model," *Sensors*, vol. 18, no. 1, p. 298, 2018.

[12] S. K. Kadhem, P. Hewson, and I. Kaimi, "Using hidden markov models to model spatial dependence in a network," *Australian & New Zealand Journal of Statistics*, vol. 60, no. 4, pp. 423–446, 2018.

[13] T.-c. Fu, "A review on time series data mining," *Engineering Applications of Artificial Intelligence*, vol. 24, no. 1, pp. 164–181, 2011.

[14] H. Dong, L. Jia, X. Sun, C. Li, and Y. Qin, "Road traffic flow prediction with a time-oriented arima model," in *2009 Fifth International Joint Conference on INC, IMS and IDC*. IEEE, 2009, pp. 1649–1652.

[15] K. Duangnate and J. W. Mjelde, "Comparison of data-rich and small-scale data time series models generating probabilistic forecasts: An application to us natural gas gross withdrawals," *Energy Economics*, vol. 65, pp. 411–423, 2017.