JOURNAL OF
EMERGING INVESTIGATORS

Article

Using data science along with machine learning to determine the ARIMA model’s ability to adjust to irregularities in the dataset

Avi Choudhary[1], Pranav Singh[1], Dhanvi Ganti[1], and Suresh Subramaniam[2]
[1]BASIS Independent Silicon Valley
[2]Department of Computer Science, Aspiring Scholars Directed Research Program

SUMMARY

Auto-Regressive Integrated Moving Average (ARIMA) models are known for their influence and application on time series data. This statistical analysis model uses time series data to depict future trends or values: a key contributor to crime mapping algorithms. With crime being a concerning topic in many urban areas like Chicago and Oakland, crime mapping algorithms have become a topic of discussion. However, the models may not function to their true potential when analyzing data with many different patterns. In order to determine the potential of ARIMA models, our research will test the model on irregularities in the data. Our team hypothesizes that the ARIMA model will be able to adapt to the different irregularities in the data that do not correspond to a certain trend or pattern. Using crime theft data and an ARIMA model, we determined the results of the ARIMA model’s forecast and how the accuracy differed on different days with irregularities in crime. For comparison purposes, we compared the model’s error when implemented on stationary data. Our results show that the model is accurate with a margin of error at least 25 cases per day when there were 250 cases per day on average. These findings will help law enforcement systems who are focused on crime suppression and help future researchers that are interested in utilizing ARIMA models to reach its true potential.

INTRODUCTION

While countries have their differences in geographical areas and culture, there is one thing that they all have in common: crime. Whether in technologically savvy areas, such as Seattle, or developing countries, such as Bangladesh, crime has inflicted its influence and violence upon its communities. Furthermore, recent statistics show that committed crimes are reported less frequently, which makes it more difficult for law enforcement to do their jobs and restore justice (1). Crime has become a prevalent issue in our society as thefts, robberies, and murders have corrupted cities. Therefore, we must utilize technologies of the modern era to confront crime. ARIMA models are implemented on time series data and use past values to forecast future values. The model is utilized for forecasting in many different applications, including COVID-19 data and livestock products. The model relies heavily on accurate and consistent time series data. However, crime data is very difficult to track, resulting in inconsistencies (2).

Furthermore, the intense corruption in law enforcement in both federal and state have also prompted a plight in receiving accurate and consistent data (3). Thus, there are inconsistences and null values in crime data, making the data harder to analyze and pre-process. As the data may not be fully pre-processed to clean the raw data of any null values or irregularities, the data may not be equipped for ARIMA model use as the model operates on smooth/consistent data, which limits its full potential on different crime time series datasets (4). However, our team hypothesizes that the ARIMA model will adapt to the different outliers in the data that do not correspond to a correlation or trend with the same accuracy. In this project, our team used, cleaned, and pre-processed Chicago Crime Time Series Data containing around 700,000 rows of raw data. We then used this to train the ARIMA model with the p, d, and q parameters. The p parameter represents the AR (auto-regressive terms) part of the model and determines the output by representing the lagged data points. The d parameter is for the differencing the values between the current time period and the previous time period and is used to determine the number of times the lagged data points were subtracted to make to make the data stationary. Because the d parameter plays a key role in transforming a non-stationary time series dataset to a stationary dataset, we assigned this parameter a low number as our purpose was to implement the ARIMA model and see how it adjusts to discrepancies. Lastly, the q parameter is used to represent the size of the moving average window; we made this value zero as we did not want to use moving average to help the model.

We created multiple graphs and visuals to determine the vulnerabilities in the models. Finally, we graphed the vulnerabilities to determine if the errors were connected to the unique inconsistencies in the data. We compared the error of the model with two different types of data implementations (non-stationary vs. stationery) to illustrate the importance of data science methods. Raw data is made stationary to ensure that the architecture or model does not fall for the false trends that can be illustrated by the raw data. Our data is deemed stationary if our rolling mean is relatively flat and does not correspond to the raw data’s trends. We used a Dicky Fuller

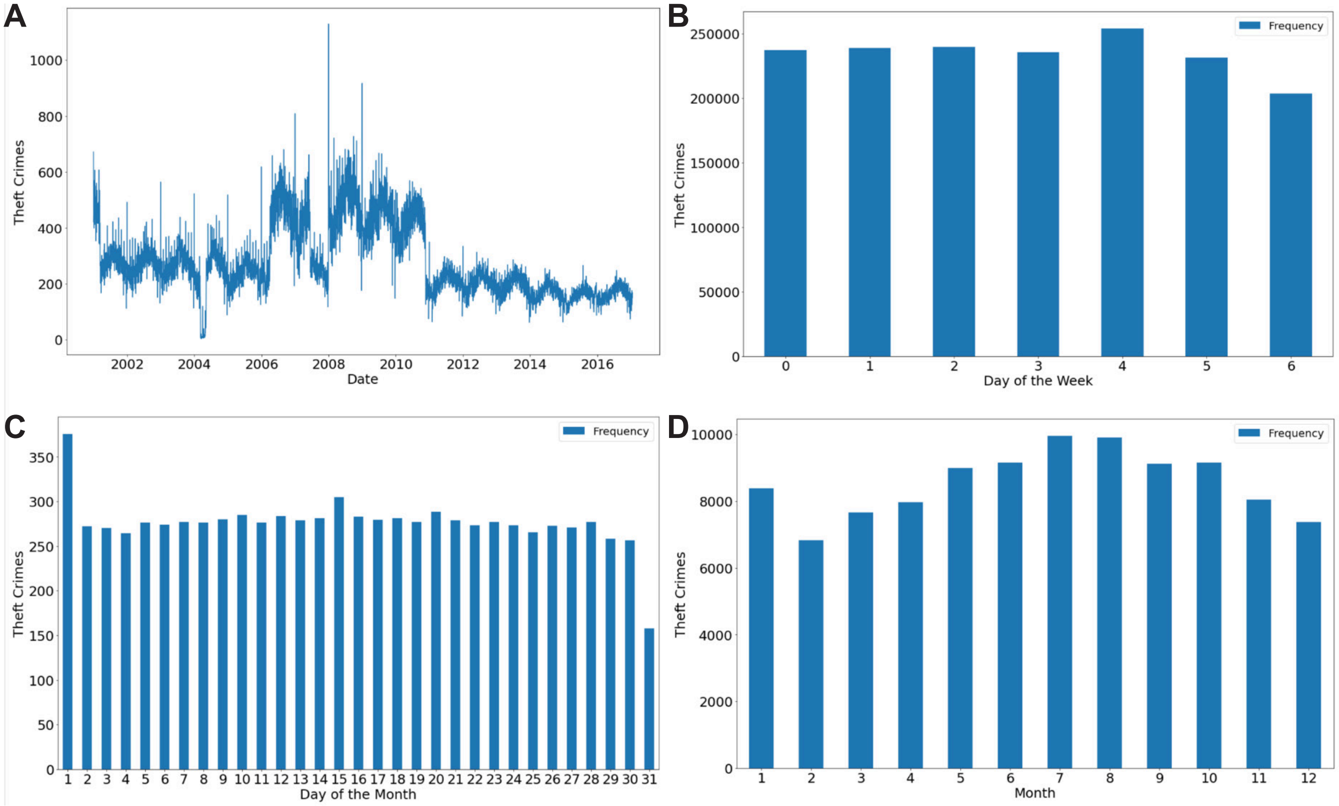

Figure 1: ARIMA dataset. (a) Raw and unorganized depiction of the dataset. The data in all the graphs include data from the years 2001 to 2017. (b) The number of thefts committed for each day in the week. The numbers 0-6 on the x-axis represent the days Monday-Sunday, respectively. (c) The number of crimes committed each month: from January to December. (d) The number of crimes committed each day of the month: from the 1st to the 31st. Note: the 31st day has the lowest days as not all months have 31 days.

Test, a test that returns the p-value used to determine the level of stationarity, on the data along with the actual rolling mean, which is used to determine the stationarity of the data(5). The comparison between the two different types of data implementations teaches us the importance of data science algorithms in handling and perfecting the data for the model to be applied. The results show that the ARIMA model does not provide accurate results for areas of data that have inconsistencies as well as they do for areas of data that are stationary and smooth. We present findings that inform the public about the importance of maintaining proper data and show how machine learning and data science cannot only rely on ARIMA models but must include other deep learning neural networks such as Recurrent Neural Networks.

RESULTS

To test the ARIMA model on the data, we had to initially remove unnecessary columns and information from the dataset. The data initially contained around 700,000 rows with 23 columns. For the project, we required the date and the number of crimes committed on that day. Furthermore, we converted the grouped rows from time periods to days. Lastly, we cut the dataset to include theft crimes as the pattern could be different based on the type of crime. We then plotted the graph to picture the dataset and then graphed the datasets based on the day of the week, month, and the different days (Figure 1). From the visuals, we could determine certain trends in the data. For instance, crime was significantly lower on Sunday, while it was highest on Friday. It is essential that we find patterns in our data since finding trends and irregularities indicates to us the potential benefits or challenges that the model may or may not face.

Our team later implemented he ARIMA model on the non-stationary dataset. Determining the p, d, and q parameters into our data was tricky, as we knew that the ARIMA model would be sensitive to the differencing patterns in the data. Thus, determining the correct parameters was crucial, as the values determined the periods and lags used to train the model (6). In order to determine the parameters, our team had plotted the Auto Regressive model (that relied on the p parameter) and the Moving Average model (that relied on the q parameter) separately in order to implement and test values for those parameters. After each of the model had reached its optimal Root Mean Squared Error (RMSE), we had used that parameter for the ARIMA model. Through our 70 and 30 train and test split, we made predictions on the crime rates of the test data. After we implemented the model, the predictions were then compared with the actual testing data to determine a RMSE of 29.031. While there is still room for improvement, the error was not too radical and far off, indicating that our

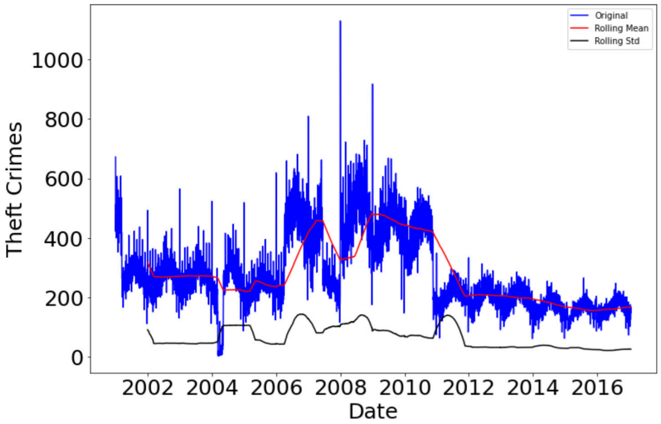

Figure 2: Non-stationary data. Initial rolling mean for the data frame before the log scale and the shift.

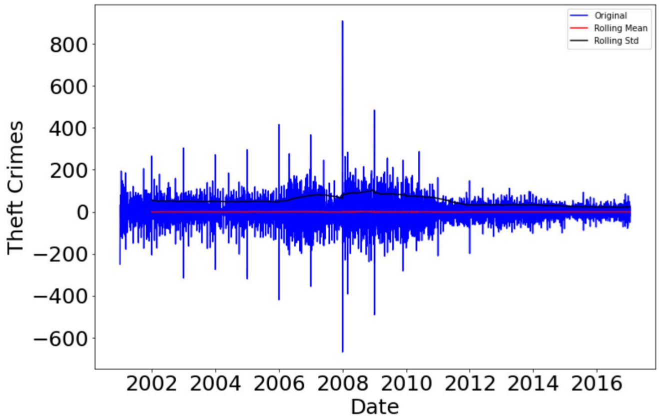

Figure 3: Stationary data. Rolling mean for the data frame after the log scale and the shift. The rolling mean (black line) here has a lower standard deviation than the original data (blue).

ARIMA model made some accurate predictions in future rates. However, when we plotted the graph that had a margin error of over 25 cases per day (when there were 250 cases on average), we found that the model had a higher margin of error on days like Friday and Sunday (days with irregular numbers of theft cases) than other days.

In order to compare the ARIMA model's application on non-stationary data to stationary data, our team also made a separate dataset that contained stationary data to apply the ARIMA model on. In Figure 2, the rolling mean and standard deviation are unstable, as the data points that are a part of a trend rely on the past data points and are dependent on one another. The ARIMA model looks at past data to forecast future points; however, if the past points indicate a correlation or pattern (as shown in Figure 2), the model will not be independent of the data points and may make false predictions because of the trend. Thus, the rolling mean should be smooth and flat so that the data will be independent from a trend or pattern and will not receive any false indications. To make the data stationary, we used the log values of the dataset and also shifted the dataset by two periods. After implementing the two methods, we plotted the new data and noticed that the rolling mean and standard deviation was more consistent (Figure 3). We noted that some of the values were negative due to the shift that was implemented. By using these methods, our team could see that the p-value had decreased from 0.039332 to 1.643561e-26. After the model was trained using the stationary data, the RMSE was 24.072. The error had improved when we implemented the ARIMA model on stationary data, illustrating that the ARIMA model was not able to handle high irregularities in the raw data and had to be made smooth.

Our team had graphed the results in order to depict the accuracy that the ARIMA model had created with its crime predictions. The graph shows the predicted values from the expected values (Figure 4). When we graph the predicted values that had a difference of over 25 cases from the expected values, we find that those points were on days which crime cases did not match the normal trend/seasonality. For instance, most of the cases that had a margin of error more than 25 were on Sunday, when the cases were at an all-time low, or on a Friday, when cases were at an all-time high. Thus, we can conclude that the ARIMA model, while somewhat adapting to the data, did not adapt to cases that were unique or different.

These findings will help law enforcement systems and crime mapping professionals who are focused on crime suppression so as to indicate the limitations of ARIMA models and the importance of collecting accurate data. The results can also help future researchers who aim to utilize ARIMA models in a way to make them accurately predict and adjust crime data and statistics.

DISCUSSION

The results stand against our initial hypothesis. We declared that the ARIMA model should maintain a certain level of accuracy when being implemented on non-stationary data. However, from this research the ARIMA model had a higher margin of error when being applied to a non-stationary dataset rather than a stationary dataset.

When we implemented the model on stationary data, the RMSE had decreased, indicating that the model was better able to adapt with smooth and consistent data. The results show that the ARIMA model needs to be supported by Recurrent Neural Networks (RNNs) or other deep learning methods, as the model has a hard time adapting to data that has a lot of statistical indifferences with one another. When it comes to crime, accuracy is crucial if politicians want to use crime mapping to focus their attention on certain areas. Our research could potentially have limitations that could affect the results. The parameters that we entered into the model (p, q, d) could have been off and could have affected the time series forecasting (7). As the parameters are used to train the model to adapt to different periods and lags, the parameters have a substantial influence on how the ARIMA model predicts the crime cases. Thus, the parameters could have been better calculated to fit the model. Moreover, we did not check our data with a SARIMA (Seasonal Auto Regressive Integrated Moving Average) model, as we found no patterns in the data due to seasonality. Future crime mapping research should be done more thoroughly with RNN's LSTM (Long Short-Term Memory) layers or weights to find the complex patterns in the data. In the future, we would like to focus on making our data more accurate by focusing more on the patterns. Our results show that there are still some irregularities that are in our data that need to be made more stationary. We would also like to incorporate other various types of crime, as we only did theft, as well as applying a similar approach to other urban cities and comparing the two. Another goal for the future is that we can try out other models, such as the SARIMA model. The results of our study and its contributions, including the predictions and the modeling, can be used to improve future crime rate trends in different cities which can allow the criminal system/government to make better choices and decisions regarding the crime in their cities.

MATERIALS AND METHODS

Data Collection

Before data collecting, we listed multiple cities that the project would apply to. These were cities with large crime rates and population density. Furthermore, it was essential to look for the time range the project was going to use time series data. Noting these points, we began our search by finding databases that contained crime data for one of the cities we listed. GitHub and Kaggle were excellent sources on finding datasets about cities that were filled with ongoing crime. We then decided on using a Kaggle Chicago Crime dataset that had a range from 2001 to 2016 (8).

Data Cleaning

Initially, one of our data's shapes had 605298 rows along with 23 columns. The 23 columns were reduced to 2 columns, as we removed columns such as the arrests, description, case number, and ID. The date column in our time series data was initially listed as a string(i.e. "1/1/01"); however, we had to convert the string to datetime format(i.e. 2001-01-01) to make it easier for the model to read. By using a package from pandas(pd.to_datetime), we were able to successfully convert the strings to datetime format. Our rows were originally created based on the time the crime was committed, so we reduced the number of rows by clustering the crimes made per day, lowering the number of rows in the dataset. Lastly, we only chose the committed crimes that were related to thefts, as they were the most apparent. Because 21 data columns were dropped and the large number of rows that were dropped/combined, our data matrix had changed significantly: (605298, 23) to (5862, 2).

Data Exploration and Visualization

We used matplotlib, a python package, to plot the dataset in different ways. We first plotted all the points in our data and found our data to be too messy to depict patterns. Thus, we started plotting and organizing our data in different ways to illustrate certain patterns. By using a package of matplotlib (plt.plot.bar), our team was able to depict the dataset successfully through bar graphs. The graphs were depicted to be bar graphs that were organized based on the formatting of the times. We first graphed our data based on the day of the week. Later, we graphed the data based on the day of the month, and then the month itself. The graphs were plotted from the years between 2001 and 2017 (inclusive).

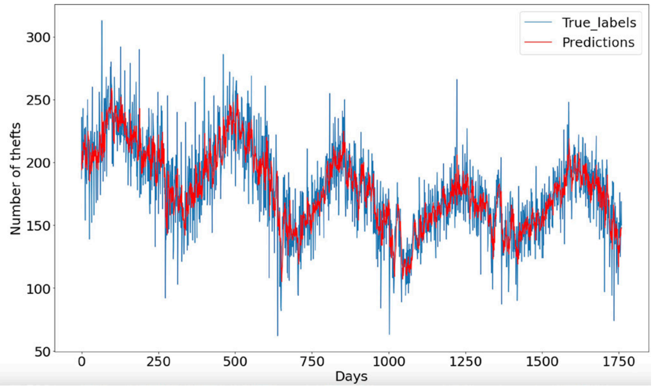

Figure 4: Predictions and the actual results of our data. The red depicts the predictions while the blue depicts the real testing data. The image shows how the red lines and blue lines do not match up, indicating their clear differences.

Making Data Stationary to Compare

Our team decided to make a copy of the original dataset and make it stationary to train another ARIMA model. To check if our data was stationary, we used ADCF (Augmented Dicky Fuller Test) and the rolling mean and standard deviation. In the tests we used .mean() and .std() to determine the mean and the standard deviation of our dataset. Our team had created a method that had the purposes of graphing the dataset along with its mean and standard deviation. The method also printed the ACDF tests and printed the p-values. Initially, the graph had shown that the mean was not stationary and highly variant; our team could also see that the p-value was higher than it should have normally been, indicating that the data may be statistically significant. Thus, our team had to shift the dataset by two time periods and also used log-values of the data points in order to make the dataset more stationary. More specifically, our team used .shift() to move the periods by 2 and .log(dataset) to lower the varying means and standard deviation of the data points. After implementing the two lines of code, the dataset was shown to be less statistically significant and more stationary. Therefore, the dataset was stationary and ready to be implemented on the ARIMA model.

ARIMA Model

Finally, we crafted the ARIMA model to implement on the datasets. Our data had to initially be split into training and testing data. We developed an original algorithm that sliced the dataset to accumulate 70% of the data towards the training set and 30% to the testing set. The parameters of the ARIMA model we used were determined through the separate implementation of Auto-Regressive models and Moving Average models. Because the p parameter was shown to heavily affect the auto-regressive model, our team had just experimented with the p parameter. After finding that the Auto-Regressive model had reached its optimal RMSE, we determined that the p parameter applied was appropriate: 5. Similarly, the q parameter was determined through the plotting of the Moving Average model. After experimenting the q parameter on the Moving Average model by checking the RMSE after implementing different values for the parameter, our team had found the appropriate value of the parameter: 1. The d parameter was determined through the implementation of the actual ARIMA model. Our team had found that the model had performed better if the model was differencing by just one period and not two. Therefore, our d parameter became 1. Then we trained the model through the training set and then used for loops to iterate through each point in the testing set to print the expected and the model's predicted values. This process was performed for both the non-stationary and stationary data.

Our team used the Root Mean Squared Error to test the Auto-Regressive, Moving Average, and ARIMA model separately. Our team also created a graph to portray the difference between the predicted and expected results of the ARIMA model. This process was done through matplotlib and had clearly shown the difference in the predicted and expected results through the contrasting colors: red for predictions and blue for expected results.

ACKNOWLEDGEMENTS

Our group would like to thank ASDRP for supporting this study. We would also like to thank our advisor, Mr. Suresh Subramaniam, a mentor in the Department of Computer science at ASDRP, for his guidance throughout this project. Note: This work was conducted over a four-month period.

REFERENCES

1. Gramlich, John. “What the Data Says (and Doesn't Say) about Crime in the United States.” Pew Research Center, Pew Research Center, 23 Nov. 2020, www.pewresearch.org/fact-tank/2020/11/20/facts-about-crime-in-the-u-s/.
2. Martinez, Yolanda. “How Hard Is It to Count Violent Crimes?” The Marshall Project, The Marshall Project, 8 Dec. 2017, www.themarshallproject.org/2017/12/08/how-hard-is-it-to-count-violent-crimes.
3. Azfar, Omar, and Tugrul Gurgur. “Police Corruption, Crime and Crime Reporting: A Simultaneous Equations Approach.” SSRN, University of Maryland, 23 Feb. 2009, paers.ssrn.com/sol3/papers.cfm?abstract_id=1348128.
4. Zhang, Xingyu et al. “Applications and comparisons of four time series models in epidemiological surveillance data.” PloS one, vol. 9, no. 2. 5 Feb. 2014, doi:10.1371/journal.pone.0088075
5. Cheung, Yin-Wong, and Kon S. Lai. “Lag Order and Critical Values of the Augmented Dickey-Fuller Test.” Journal of Business & Economic Statistics, vol. 13, no. 3, 1995, pp. 277–280. JSTOR, www.jstor.org/stable/1392187. Accessed 14 Apr. 2021.
6. Brockwell, Peter J., and Richard A. Davis. Introduction to Time Series and Forecasting. Springer, 2010.
7. Holmberg, Daniel. “ARIMA Forecasting in Python.” Medium, Towards Data Science, 11 June 2020, towardsdatascience.com/arima-forecasting-in-python-90d36c2246d3.
8. Currie, David. “Crimes in Chicago.” Kaggle, Chicago, 28 Jan. 2017, www.kaggle.com/currie32/crimes-in-chicago.

Article submitted: February 27, 2021
Article accepted: April 6, 2021
Article published: July 20, 2021