CRIME FORECASTING USING DEEP LEARNING

PROJECT REPORT

Submitted by

PRAVEEN ELANGO 2015115099

DIWAKAR 2017115524

submitted to the Faculty of

INFORMATION AND COMMUNICATION ENGINEERING

in partial fulfilment for the award of the degree of

BACHELOR OF TECHNOLOGY

in

INFORMATION TECHNOLOGY

DEPARTMENT OF INFORMATION SCIENCE AND TECHNOLOGY

COLLEGE OF ENGINEERING, GUINDY

ANNA UNIVERSITY

CHENNAI 600 025

MAY 2021

ANNA UNIVERSITY

CHENNAI – 600 025

BONAFIDE CERTIFICATE

Certified that this project report titled CRIME FORECASTING USING DEEP LEARNING is the bonafide work of PRAVEEN ELANGO (2015115099), DIWAKAR (2017115524) who carried out project work under my supervision. Certified further that to the best of my knowledge and belief, the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or an award was conferred on an earlier occasion on this or any other candidate.

PLACE:

DATE: 25/05/2021

Dr.L.SAI RAMESH
TEACHING FELLOW
PROJECT GUIDE
DEPARTMENT OF IST, CEG
ANNA UNIVERSITY
CHENNAI 600025

COUNTERSIGNED

Dr. SASWATI MUKHERJEE
HEAD OF THE DEPARTMENT
DEPARTMENT OF INFORMATION SCIENCE AND TECHNOLOGY
COLLEGE OF ENGINEERING, GUINDY
ANNA UNIVERSITY
CHENNAI 600025

ABSTRACT

Crime is a significant threat to humankind. There are many crimes that happen at regular intervals of time. Perhaps it is increasing and spreading at an alarming rate. Crimes happen from small villages and towns to big cities. Criminal incidents have increased at an alarming rate and it is the responsibility of the police department to control and reduce the occurrence of crime. An important step taken by police departments in the effort to reduce crime is the use of predictive policing. Predictive policing involves using algorithms to analyze massive amounts of information in order to predict and help prevent potential future crimes. The proposed approach is based on crime data analysis using deep learning methods to forecast crime rates.

திட்டப்பணி ச்சுருக்கம்

குற்றம் என்பது மனிதகுலத்திற்கு ஒரு குறிப்பிடத்தக்க அச்சுறுத்தலாகும். சரியான நேரத்தில் இடைவெளியில் பல குற்றங்கள் நடக்கின்றன. ஒருவேளை அது ஆபத்தான விகிதத்தில் அதிகரித்து பரவுகிறது. சிறிய கிராமங்கள் மற்றும் நகரங்கள் முதல் பெரிய நகரங்கள் வரை குற்றங்கள் நடக்கின்றன. குற்ற சம்பவங்கள் ஆபத்தான விகிதத்தில் அதிகரித்துள்ளன, மேலும் குற்றங்கள் ஏற்படுவதைக் கட்டுப்படுத்துவதும் குறைப்பதும் காவல் துறையின் பொறுப்பாகும். குற்றங்களைக் குறைக்கும் முயற்சியில் பொலிஸ் திணைக்களங்கள் எடுக்கும் ஒரு முக்கியமான படியாக முன்கணிப்பு பொலிஸின் பயன்பாடு உள்ளது. முன்கணிப்பு பொலிஸ் என்பது எதிர்காலத்தில் நிகழக்கூடிய குற்றங்களைத் தடுக்கவும் தடுக்கவும் பாரிய அளவிலான தகவல்களை பகுப்பாய்வு செய்ய வழிமுறைகளைப் பயன்படுத்துவதை உள்ளடக்குகிறது. முன்மொழியப்பட்ட அணுகுமுறை குற்ற விகிதங்களை முன்னறிவிக்க ஆழ்ந்த கற்றல் முறைகளைப் பயன்படுத்தி குற்ற தரவு பகுப்பாய்வை அடிப்படையாகக் கொண்டது.

ACKNOWLEDGEMENT

First and foremost, we would like to express our deep sense of gratitude to our guide **Dr. L.SAI RAMESH**, Teaching Fellow, Department of Information Science and Technology, Anna University for his excellent guidance, counsel, continuous support and patience. He helped us with this topic and guided us in the development of this project. He gave us the moral support to finish out creative and innovative project in a successful manner.

We express our sincere gratitude to **Dr. SASWATI MUKHERJEE**, Professor and Head, Department of Information Science and Technology, Anna University, for her kind support and for providing necessary facilities to carry out the work.

We also express our gratitude to our project committee coordinator **Dr.S.SWAMYNATHAN**, Professor and the committee members **Dr.J.INDUMATHI**, Professor, **Dr.SELVI RAVINDRAN**, Assistant Professor, **Dr.P.GEETHA**, Assistant Professor, **MS.S.KANIMOZHI**, Teaching Fellow, **Ms.R.L.JASMINE**, Teaching Fellow, **Dr.B.SENTHILNAYAKI**, Teaching Fellow, Department of Information Science and Technology, Anna University, Chennai, for their valuable guidance and technical support.

PRAVEEN ELANGO

DIWAKAR

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Police departments in some of the largest U.S. cities have been experimenting with predictive policing as a way to forecast criminal activity. Predictive policing involves using algorithms to analyze massive amounts of information in order to predict and help prevent potential future crimes. It uses computer algorithms to predict future crimes and can assist in addition to instincts of police officers. It can also be a more cost-efficient method to reduce criminal incidents.

In the proposed approach, the recurrent neural network models, Long short-term memory (LSTM) and NeuralProphet and the mathematical model, Autoregressive integrated moving average (ARIMA) shall be used for deep learning. The main idea is to obtain a crime dataset concerned with a particular region over a period of time and transform the dataset into ordered time series with specific target variables. The time series shall then be split into training and testing sets and trained with the models specified previously. After training is completed for each model, forecasts shall then made for each target variable and the performance of each model shall be analyzed based on the margin of error between the training and testing set.

1.1 MOTIVATION

There has been a rampant increase in criminal incidents worldwide. The motivation for the development of the project is providing automatically an efficient way to analyze and forecast crime rates. The proposed solution is based on deep learning techniques and can make the case-solving and investigation process easier and faster.

1.2 PROBLEM STATEMENT

A research paper may belong to more than one category at the same time. In existing literature, many deep learning methods have been applied for forecasting crime rates. But there is a limitation in identifying patterns and factors of crime incidents over a period of time. Advancements in deep learning algorithms can traverse various datasets and reveal new information.

1.3 OBJECTIVE

The project aims to find an approachable and efficient method for forecasting of crime rates. When multiple deep learning models are trained for crime forecasting of specific target variables, forecasts made by each model shall be observed. The discrepancies in results between each model shall be inferred for a better understanding of the task to be accomplished. The performance of each model shall then be analyzed based on the margin of error between the training and testing set and a conclusion shall be reached about the model best suited for forecasting.

1.4 ORGANIZATION OF THE REPORT

The project report is organized as follows,

chapter 2 discusses the existing systems and various methods required for the proposed system.

chapter 3 discusses the various concepts used in the proposed system along with the overall system architecture.

chapter 4 discusses the implementation details of the proposed system along with the necessary algorithms and the experimental results

chapter 5 concludes the report by summarizing the total result and proposes possible enhancements that can be done in the future.

CHAPTER 2

LITERATURE SURVEY

A literature survey is done by surveying research papers. The limitations and the knowledge gained from the papers will help us to create a better system.

2.1 A COMPARATIVE STUDY AND ANALYSIS OF TIME SERIES FORECASTING TECHNIQUES

This paper proposes to perform three different types of time series forecasting [8]. They have applied regression, LSTM, ARIMA, CNN, fuzzy-based method weighted MVFTS and CBLSTM on three discrete datasets. The performance of the above-mentioned methods for forecasting, over all the three different datasets have shown the best results by LSTMs, CNNs and weighted MVFTS and CBLSTM. ARIMA too performed significantly well; however, regression was not able to perform well.

2.2 FORECASTING CRIME WITH DEEP LEARNING

This paper proposes to use deep neural networks, including variations that are suited to the spatial and temporal aspects of the crime prediction problem, in order to make next day crime count predictions in a fine-grain city partition [2]. The crime counts were broken into 10 bins and the model predicted the most likely bin for each spatial region at a daily level. Predictions were made using Chicago and Portland crime data, which were augmented with additional datasets covering weather, census data, and public transportation. The models include a Feed forward network, a Convolutional network, a Recurrent network and a Recurrent convolutional network. With their best model, they were able to predict the correct bin for overall crime count with 75.6% and 65.3% accuracy for Chicago and Portland, respectively.

2.3 PREDICTING INCIDENTS OF CRIME THROUGH LSTM NEURAL NETWORKS IN SMART CITY DOMAIN

In this research work [9], a deep learning based approach is proposed for the classification of incidents of a crime of public safety through predictive analysis. The predictive model is based on a neural network Long Short-Term Memory (LSTM), trained with a small group of attributes, enabling the prediction of the class label in the validation stage, with a high percentage of prediction accuracy. The proposed approach is evaluated through a big data set (real data) of type open data, which contains historical information about the crimes of a smart city. Their deep learning approach achieved high performance in the final model with 87.84% accuracy based on the validation data. Furthermore, the final LSTM model achieved an average loss function of 0.0376 on validation data, using 20% of the data set for the testing stage.

2.4 A COMPARISON OF TIME SERIES MODEL FORECASTING METHODS ON PATENT GROUPS

This paper proposes to create a technology forecasting model based on the sequence of patents issued over a given time period [7]. The focus was to apply time series modeling techniques to a collection of USPTO patents from 1996 to 2013. The techniques used were Holt-Winters Exponential Smoothing and Autoregressive Integrated Moving Averages (ARIMA). Cross validation methods were used to determine the best fitting models and ultimately whether or not patent data could be modeled as a time series. For each model and 15-month forecast, four error statistics were calculated: Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), and Mean Absolute Scaled Error (MASE). According to the experimental results, the MASE values indicated that all of the models had adequate forecasting capabilities. The results suggested that ARIMA acted as a better predictor for the GCSSA and DFMDS data while the AI patent data seemed to be better suited for an Exponential Smoothing model.

2.5 PREDICITVE ANALYSIS OF CRIME DATA USING DEEP LEARNING

In this work [3], crime is predicted using Recurring LSTM networks. An existing structured dataset was used to predict the crimes. The data was split into training and testing data. Both the testing and training undergo the training process. The resultant training and testing data was compared with the actual crime count and was visualized. The obtained output was compared with an existing system. The accuracy of this model was compared with an existing model and concluded that the LSTM model provided better efficiency than the existing model.

CHAPTER 3

SYSTEM DESIGN

3.1 PROPOSED SYSTEM ARCHITECTURE

The proposed system follows a tightly coupled architecture comprising of the following modules.

- Dataset
- Time series transformation
- Data preparation
- Model forecasting
- Model Evaluation

3.2 ARCHITECTURE DIAGRAM

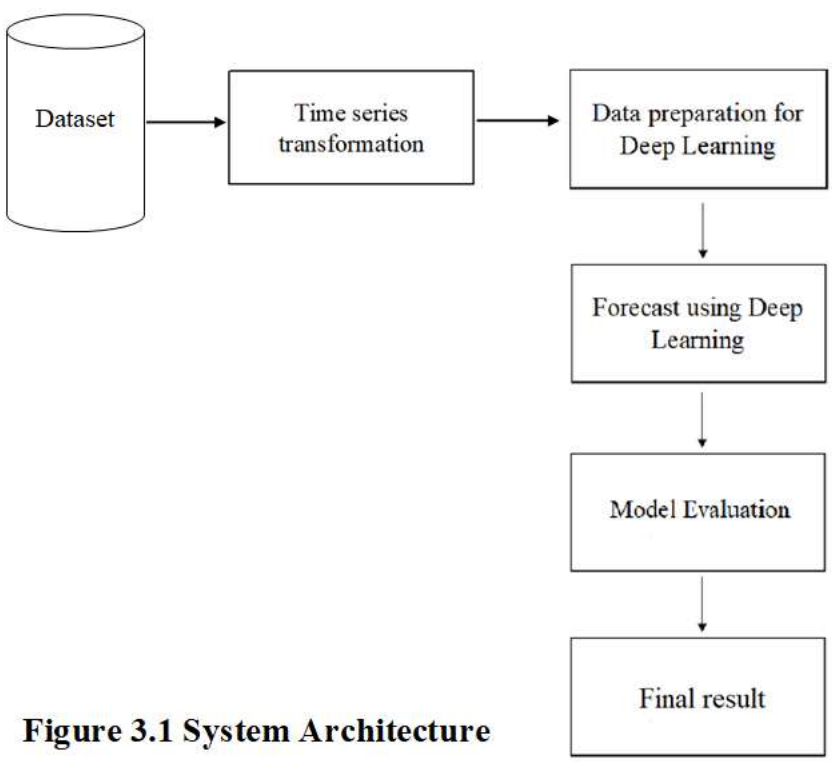


Figure 3.1 System Architecture

3.3 MODULE DESCRIPTION

The description of the modules is mentioned below.

3.3.1 Dataset

The most important part of any deep learning based project is the dataset. The dataset used in this project reflects reported incidents of crime that occurred in the city of Chicago from January 1, 2010 to December 31, 2018. The dataset is obtained from the official Chicago Data Portal (https://data.cityofchicago.org/).

3.3.2 Time Series Transformation

In this module, a datetime object is created in the format YYYY/MM/DD in order to transform the dataframe into ordered time series data for forecasting [5]. The target variables to be used for forecasting are **arrest**, **domestic** and **total crime count**. Therefore, three separate univariate time series are created. All the other attributes are dropped except the date and respective target variables of the series to be created. The dataframes are grouped by the required value of the target variables and hence count of each target variable corresponding to every date from 2010-2018 are obtained. Finally, the date attribute is set as the index to form the ordered time series.

3.3.3 Data Preparation and Model Forecasting

In this module, the initial step is to prepare the data for training. Firstly, the time series created are checked for stationarity using Augmented Dickey-Fuller test. The time series are then split into training set and testing set in the ratio of 80% and 20%. They are then converted into a NumPy array and the values are scaled using a scaler if required for the deep learning accordingly. After the parameters of each model are specified respectively, the models are fit on the training sets and trained [1]. After training, the models are applied on the testing sets and the training and testing mean square error/mean absolute error are obtained. This process is performed individually for each model and each time series.

The following deep learning models are used for training and testing:

i) Long Short-Term Memory (LSTM)

ii) Autoregressive integrated moving average (ARIMA)

iii) NeuralProphet

3.3.4 Model Evaluation

After the training and testing mean square error/mean absolute error are obtained for each model and time series, the model parameters are adjusted in order to reduce the margin of error and overfitting/underfitting as much as possible.

3.3.5 Final Result

The final forecast of all three variables of each model is plotted.

CHAPTER 4

IMPLEMENTATION AND RESULTS

4.1 TOOLS USED

The following tools, libraries and environments are used in this project.

4.1.1 Pandas

Pandas is a software library written for the Python programming language for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series.

4.1.2 NumPy

NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.

4.1.3 Matplotlib

Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy. It is used for creating static, animated, and interactive visualizations in Python.

4.1.4 Seaborn

Seaborn is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics.

4.1.5 Scikit-learn

Scikit-learn is a free software machine learning library for the Python programming language. It features various classification, regression and clustering and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy.

4.1.6 Keras

Keras is an open-source software library that provides a Python interface for artificial neural networks. Keras acts as an interface for the TensorFlow library.

4.1.7 Pmdarima

Pmdarima (originally pyramid-arima, for the anagram of 'py' + 'arima') is a statistical library designed to fill the void in Python's time series analysis capabilities.

4.1.8 Plotly

Plotly is an interactive, open-source, and browser-based graphing library for Python.

4.1.9 Statsmodels

Statsmodels is a Python module that provides classes and functions for the estimation of many different statistical models, as well as for conducting statistical tests, and statistical data exploration.

4.1.10 NeuralProphet

NeuralProphet is a python library for modeling time-series data based on neural networks. It’s built on top of PyTorch and is heavily inspired by Facebook Prophet and AR-Net libraries.

4.1.11 Livelossplot

Livelossplot is a library to plot live training in Jupyter Notebook for Keras, PyTorch and other frameworks. It is an open-source Python package that can be used to observe each epoch of training.

4.1.12 Google Colab Notebooks

Colaboratory, or “Colab” for short, is a product from Google Research. Colab allows anybody to write and execute arbitrary python code through the browser, and is especially well suited to machine learning, data analysis and education.

4.2 MODELS USED

4.2.1 Long short-term memory

Long short-term memory (LSTM) is an artificial recurrent neural network (RNN) architecture used in the field of deep learning. LSTM networks are well-suited to classifying, processing and making predictions based on time series data, since there can be lags of unknown duration between important events in a time series.

4.2.2 Autoregressive Integrated Moving Average

An Autoregressive Integrated Moving Average (ARIMA) model is a class of statistical models for analyzing and forecasting time series data. It explicitly caters to a suite of standard structures in time series data, and as such provides a simple yet powerful method for making skillful time series forecasts.

4.2.3 NeuralProphet

NeuralProphet is a new framework that extends on the original Prophet framework, addresses pain points such as scale, customization, and extensibility. It incorporates traditional statistical and neural network models for time series modeling, used in forecasting and anomaly detection.

4.3 PROJECT WORK IMPLEMENTATION

4.3.1 Getting Started

In the initial step, the libraries and tools mentioned previously are imported in the Colab notebook. The dataset to be used for the project is then retrieved and imported and a data frame is created as shown in Figure 4.1.

	Case Number	Date	Block	IUCR	Primary Type	Description	Location Description	Arrest	Domestic	Beat	Ward	FBI Code	X Coordinate	Y Coordinate	Year	Latitude	Longitude	Location
0	HY416556	09/09/2010 08:10:00 PM	074XX S MARYLAND AVE	1154	DECEPTIVE PRACTICE	FINANCIAL IDENTITY THEFT $300 AND UNDER	RESIDENCE	False	False	323	6.0	11	1183191.0	1855830.0	2010	41.759594	-87.604169	(41.759593809, -87.604169095)
1	HY431076	01/15/2010 12:01:00 AM	044XX S OAKLEY AVE	1754	OFFENSE INVOLVING CHILDREN	AGG SEX ASSLT OF CHILD FAM MBR	RESIDENCE	True	False	924	12.0	02	1161779.0	1875003.0	2010	41.812678	-87.682112	(41.812678317, -87.682111569)
2	HS594517	11/01/2010 10:35:00 AM	029XX W WILCOX ST	2093	NARCOTICS	FOUND SUSPECT NARCOTICS	SIDEWALK	True	False	1124	2.0	18	1156795.0	1899213.0	2010	41.879216	-87.699738	(41.87921551, -87.699737903)
3	HY435598	06/01/2010 12:00:00 PM	005XX N MICHIGAN AVE	1153	DECEPTIVE PRACTICE	FINANCIAL IDENTITY THEFT OVER $ 300	SMALL RETAIL STORE	False	False	1834	42.0	11	1177342.0	1903838.0	2010	41.891466	-87.624153	(41.891465732, -87.624153044)
4	HS587770	10/28/2010 07:21:53 PM	008XX N AVERS AVE	2024	NARCOTICS	POSS: HEROIN(WHITE)	APARTMENT	True	False	1112	27.0	18	1150571.0	1905280.0	2010	41.895988	-87.722433	(41.895987813, -87.722432928)
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
2687185	JD448774	12/10/2018 12:01:00 AM	107XX S RIDGEWAY AVE	1752	OFFENSE INVOLVING CHILDREN	AGGRAVATED CRIMINAL SEXUAL ABUSE BY FAMILY MEMBER	RESIDENCE	True	True	2211	19.0	17	1153258.0	1833037.0	2018	41.697690	-87.714474	(41.697689632, -87.714474258)
2687186	JB555242	12/16/2018 02:10:00 AM	041XX W 30TH ST	051A	ASSAULT	AGGRAVATED - HANDGUN	STREET	False	False	1031	22.0	04A	1149311.0	1884390.0	2018	41.838688	-87.727602	(41.838687711, -87.727602064)
2687187	JB269440	05/18/2018 11:34:00 PM	071XX S EUCLID AVE	0110	HOMICIDE	FIRST DEGREE MURDER	STREET	True	False	333	5.0	01A	1190280.0	1857899.0	2018	41.765103	-87.578122	(41.765103492, -87.578121854)
2687188	JB456068	09/29/2018 12:00:00 PM	042XX N MAJOR AVE	0610	BURGLARY	FORCIBLE ENTRY	RESIDENCE	True	False	1624	38.0	05	1137586.0	1927566.0	2018	41.957387	-87.769587	(41.957386868, -87.769586624)
2687189	JB108111	01/08/2018 12:30:00 AM	085XX S COTTAGE GROVE AVE	0610	BURGLARY	FORCIBLE ENTRY	OTHER (SPECIFY)	False	False	632	8.0	05	1183049.0	1848622.0	2018	41.739818	-87.604913	(41.739817594, -87.604913162)

2687190 rows × 18 columns

Figure 4.1 Data frame

4.3.2 Data Preprocessing

The dataset is checked for null and duplicate values and dropped. Inconsistencies in attribute names are handled and primary key type attributes are removed as they are of no use for analysis. The date column is converted to datetime object to get the day of the week, month and year of the crime for creation of time series [4]. The attributes that are not required for time series creation are dropped. The crimes are mapped under one general group of crime and by location [10].

	date	arrest	domestic	year	location	crimes	location_type
0	2010-09-09	False	False	2010	(41.759593809, -87.604169095)	CRIME	RESIDENCE
1	2010-01-15	True	False	2010	(41.812678317, -87.682111569)	CRIME	RESIDENCE
2	2010-11-01	True	False	2010	(41.87921551, -87.699737903)	CRIME	PUBLIC_AREA
3	2010-06-01	False	False	2010	(41.891465732, -87.624153044)	CRIME	BUSINESS
4	2010-10-28	True	False	2010	(41.895987813, -87.722432928)	CRIME	RESIDENCE
...	...	...	...	...	...	...	...
2685660	2018-06-28	False	False	2018	(41.972101049, -87.654899332)	CRIME	RESIDENCE
2685661	2018-12-10	True	True	2018	(41.697689632, -87.714474258)	CRIME	RESIDENCE
2685662	2018-12-16	False	False	2018	(41.838687711, -87.727602064)	CRIME	PUBLIC_AREA
2685663	2018-05-18	True	False	2018	(41.765103492, -87.578121854)	CRIME	PUBLIC_AREA
2685664	2018-09-29	True	False	2018	(41.957386868, -87.769586624)	CRIME	RESIDENCE

2685665 rows × 7 columns

Figure 4.2 Dataset after preprocessing

4.3.3 Time Series Transformation

To assist training using deep learning models, the data is to be transformed into ordered time series. The variables **arrest**, **domestic** and **total crime count** each are used as target variables and individual time series are created. This is done with the help of the datetime object that is already created and by using groupby() and count() functions of Python [5].

	arrest
date	
2010-01-01	338
2010-01-02	232
2010-01-03	214
2010-01-04	225
2010-01-05	311
...	...
2018-12-27	115
2018-12-28	172
2018-12-29	151
2018-12-30	122
2018-12-31	117

3287 rows × 1 columns

	domestic
date	
2010-01-01	233
2010-01-02	121
2010-01-03	141
2010-01-04	119
2010-01-05	102
...	...
2018-12-27	94
2018-12-28	121
2018-12-29	119
2018-12-30	125
2018-12-31	99

3287 rows × 1 columns

	crimes
date	
2010-01-01	1355
2010-01-02	792
2010-01-03	759
2010-01-04	855
2010-01-05	910
...	...
2018-12-27	591
2018-12-28	694
2018-12-29	666
2018-12-30	642
2018-12-31	610

3287 rows × 1 columns

Figure 4.3 Time series created from 1 Jan 2010 – 31 Dec 2018

4.3.4 Data Preparation and Model Forecasting

Firstly, the time series created are checked for stationarity using Augmented Dickey-Fuller test. A stationary time series is one whose properties do not depend on the time at which the series is observed.

```
adf_test = ADFTest(alpha = 0.05)
adf_test.should_diff(df1)

(0.01, False)
```

```
adf_test = ADFTest(alpha = 0.05)
adf_test.should_diff(df2)

(0.01, False)
```

```
adf_test = ADFTest(alpha = 0.05)
adf_test.should_diff(df3)

(0.01, False)
```

Figure 4.4 Augmented Dickey-Fuller test

From Figure 4.4, it can be observed that the p-value is less than 0.05 and the null hypothesis that there is a unit root, is rejected for all the time series. Hence it can be concluded the time series created are stationary.

```
df1 = df1.astype('float32')
df1 = np.reshape(df1, (-1, 1))
scaler = MinMaxScaler(feature_range=(0, 1))
df1 = scaler.fit_transform(df1)
train_size = int(len(df1) * 0.80)
test_size = len(df1) - train_size
train, test = df1[0:train_size,:], df1[train_size:len(df1),:]

def create_dataset(df1, look_back=1):
    X, Y = [], []
    for i in range(len(df1)-look_back-1):
        a = df1[i:(i+look_back), 0]
        X.append(a)
        Y.append(df1[i + look_back, 0])
    return np.array(X), np.array(Y)

look_back = 30
X_train, Y_train = create_dataset(train, look_back)
X_test, Y_test = create_dataset(test, look_back)

X_train = np.reshape(X_train, (X_train.shape[0], 1, X_train.shape[1]))
X_test = np.reshape(X_test, (X_test.shape[0], 1, X_test.shape[1]))
```

Figure 4.5 Data preparation for LSTM model

As shown in Figure 4.5, for LSTM model, the time series are converted into a NumPy array and the values are fitted after being scaled using a MinMax scaler [9]. The time series are then split into training set and testing set in the ratio of 80% and 20% respectively for training.

The model parameters are specified and the model is fitted and trained on the training set and validated on the testing set as shown in Figure 4.6.

```
model = Sequential()
model.add(LSTM(100, input_shape=(X_train.shape[1], X_train.shape[2])))
model.add(Dropout(0.2))
model.add(Dense(1))
model.compile(loss='mean_squared_error', optimizer='adam')

history = model.fit(X_train, Y_train, epochs=20, batch_size=70, validation_data=(X_test, Y_test),
                    callbacks=[EarlyStopping(monitor='val_loss', patience=10)], verbose=1, shuffle=False)

model.summary()
```

```
Epoch 1/20
38/38 [==============================] - 3s 26ms/step - loss: 0.1513 - val_loss: 0.0035
Epoch 2/20
38/38 [==============================] - 0s 5ms/step - loss: 0.0146 - val_loss: 0.0034
Epoch 3/20
38/38 [==============================] - 0s 5ms/step - loss: 0.0125 - val_loss: 0.0034
Epoch 4/20
38/38 [==============================] - 0s 5ms/step - loss: 0.0123 - val_loss: 0.0034
Epoch 5/20
38/38 [==============================] - 0s 5ms/step - loss: 0.0121 - val_loss: 0.0033
Epoch 6/20
38/38 [==============================] - 0s 5ms/step - loss: 0.0124 - val_loss: 0.0033
Epoch 7/20
38/38 [==============================] - 0s 5ms/step - loss: 0.0112 - val_loss: 0.0033
Epoch 8/20
38/38 [==============================] - 0s 5ms/step - loss: 0.0108 - val_loss: 0.0033
Epoch 9/20
38/38 [==============================] - 0s 5ms/step - loss: 0.0109 - val_loss: 0.0032
Epoch 10/20
38/38 [==============================] - 0s 5ms/step - loss: 0.0109 - val_loss: 0.0032
Epoch 11/20
38/38 [==============================] - 0s 5ms/step - loss: 0.0101 - val_loss: 0.0032
Epoch 12/20
38/38 [==============================] - 0s 5ms/step - loss: 0.0097 - val_loss: 0.0032
Epoch 13/20
38/38 [==============================] - 0s 5ms/step - loss: 0.0097 - val_loss: 0.0032
Epoch 14/20
38/38 [==============================] - 0s 5ms/step - loss: 0.0096 - val_loss: 0.0032
Epoch 15/20
38/38 [==============================] - 0s 5ms/step - loss: 0.0097 - val_loss: 0.0032
Epoch 16/20
38/38 [==============================] - 0s 5ms/step - loss: 0.0096 - val_loss: 0.0032
Epoch 17/20
38/38 [==============================] - 0s 5ms/step - loss: 0.0093 - val_loss: 0.0031
Epoch 18/20
38/38 [==============================] - 0s 5ms/step - loss: 0.0096 - val_loss: 0.0032
Epoch 19/20
38/38 [==============================] - 0s 5ms/step - loss: 0.0096 - val_loss: 0.0032
Epoch 20/20
38/38 [==============================] - 0s 5ms/step - loss: 0.0096 - val_loss: 0.0031
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm (LSTM)                  (None, 100)               52400
_________________________________________________________________
dropout (Dropout)            (None, 100)               0
_________________________________________________________________
dense (Dense)                (None, 1)                 101
=================================================================
Total params: 52,501
Trainable params: 52,501
Non-trainable params: 0
```

Figure 4.6 Training using LSTM model

The entire process is followed for all the time series that are created, for training using LSTM model.

For training using ARIMA model, the time series are split into training set and testing set in the ratio of 80% and 20% respectively. The **auto_arima** function is then used to fit the ARIMA model with the lowest AIC value by performing a stepwise search with several different combinations of the specified parameters. After the model is fitted, predictions are made on the test set. This process is followed for all the time series that are created, for training using ARIMA model.

```
arima_model1 =  auto_arima(train,start_p = 1, start_q = 1,
                          max_p = 3, max_q = 3, m = 12,
                          start_P = 0, seasonal = True,
                          d = None, D = 1, trace = True,
                          error_action ='ignore',
                          suppress_warnings = True,
                          stepwise = True)
```

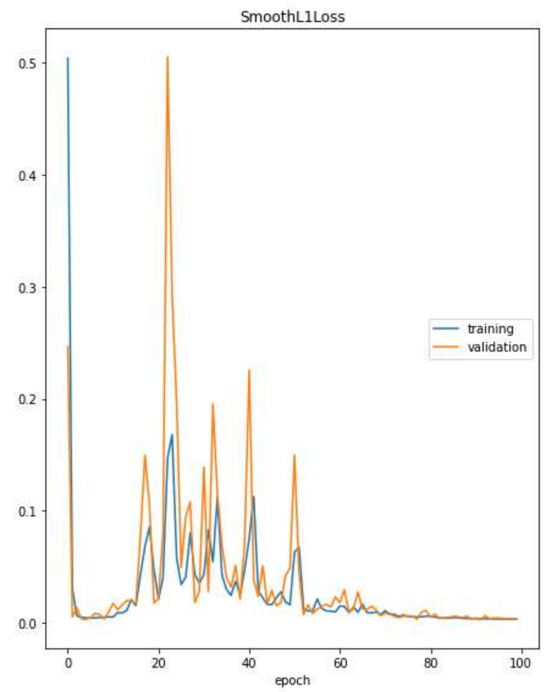
```
Performing stepwise search to minimize aic
 ARIMA(1,0,1)(0,1,1)[12] intercept   : AIC=inf, Time=18.94 sec
 ARIMA(0,0,0)(0,1,0)[12] intercept   : AIC=28362.749, Time=0.14 sec
 ARIMA(1,0,0)(1,1,0)[12] intercept   : AIC=27345.128, Time=6.54 sec
 ARIMA(0,0,1)(0,1,1)[12] intercept   : AIC=26665.918, Time=13.17 sec
 ARIMA(0,0,0)(0,1,0)[12]             : AIC=28361.213, Time=0.10 sec
 ARIMA(0,0,1)(0,1,0)[12] intercept   : AIC=28029.760, Time=1.26 sec
 ARIMA(0,0,1)(1,1,1)[12] intercept   : AIC=inf, Time=16.52 sec
 ARIMA(0,0,1)(0,1,2)[12] intercept   : AIC=inf, Time=33.15 sec
 ARIMA(0,0,1)(1,1,0)[12] intercept   : AIC=27362.919, Time=6.57 sec
 ARIMA(0,0,1)(1,1,2)[12] intercept   : AIC=inf, Time=62.01 sec
 ARIMA(0,0,0)(0,1,1)[12] intercept   : AIC=27168.675, Time=4.28 sec
 ARIMA(0,0,2)(0,1,1)[12] intercept   : AIC=inf, Time=19.58 sec
 ARIMA(1,0,0)(0,1,1)[12] intercept   : AIC=inf, Time=9.32 sec
 ARIMA(1,0,2)(0,1,1)[12] intercept   : AIC=inf, Time=29.23 sec
 ARIMA(0,0,1)(0,1,1)[12]             : AIC=26687.393, Time=3.12 sec

Best model:  ARIMA(0,0,1)(0,1,1)[12] intercept
Total fit time: 223.975 seconds
```

Figure 4.7 Training using ARIMA model

For training using NeuralProphet model, the model is fitted on the time series data with the specified parameters. After the model is fitted, predictions are made on the validation set.

```
model = NeuralProphet()
metrics = model.fit(df3, validate_each_epoch=True, freq="D", epochs=100, plot_live_loss=True, valid_p=0.3)
```




```
Epoch[100/100]: 100%|██████████| 100/100 [00:18<00:00,  5.37it/s, SmoothL1Loss=0.00309, MAE=53.7, RegLoss=0, MAE_val=61.2, SmoothL1Loss_val=0.00351]
```

Figure 4.8 Training using NeuralProphet model

4.3.5 Model Evaluation

The following results are obtained after training and testing each model on the time series. The respective model parameters are adjusted in order to reduce the margin of error and overfitting/underfitting as much as possible.

a) LSTM

i) Arrest

```
Train Mean Absolute Error: 22.342642923290754
Test Mean Absolute Error: 14.28537202287654
```

Figure 4.9 LSTM results for Arrest variable

From Figure 4.9, it can be observed that the margin of error between the training set and testing set is very less, hence it can be concluded that the LSTM model has performed reasonably well in forecasting for the arrest variable.

ii) Domestic

```
Train Mean Absolute Error: 12.865650764046686
Test Mean Absolute Error: 11.860252346087414
```

Figure 4.10 LSTM results for Domestic variable

From Figure 4.10, it can be observed that the margin of error between the training set and testing set is almost none, hence it can be concluded that the LSTM model has performed excellently in forecasting for the domestic variable.

iii) Total Crime Count

```
Train Mean Absolute Error: 53.38488915164647
Test Mean Absolute Error: 42.34300517494848
```

Figure 4.11 LSTM results for Total Crime Count variable

From Figure 4.11, it can be observed that the margin of error between the training set and testing set is relatively less, hence it can be concluded that the LSTM model has performed well in forecasting for the total crime count variable.

b) ARIMA

i) Arrest

```
mean_absolute_error(test["arrest"], prediction)
```

```
20.734251768917304
```

Figure 4.12 ARIMA results for Arrest variable

From Figure 4.12, it can be observed that the margin of error between the testing set and the forecast is large, hence it can be concluded that the ARIMA model has performed poorly in forecasting for the arrest variable.

ii) Domestic

```
mean_absolute_error(test["domestic"], prediction)
```

```
20.279980768003036
```

Figure 4.13 ARIMA results for Domestic variable

From Figure 4.13, it can be observed that the margin of error between the testing set and the forecast is as large as the arrest variable, hence it can be concluded that the ARIMA model has again performed poorly in forecasting for the domestic variable.

iii) Total Crime Count

```
mean_absolute_error(test["crimes"], prediction)

89.33179547288115
```

Figure 4.14 ARIMA results for Total Crime Count variable

From Figure 4.14, it can be observed that the margin of error between the testing set and the forecast is huge, hence it can be concluded that the ARIMA model has performed the worst in forecasting for the total crime count variable.

c) NeuralProphet

i) Arrest

```
SmoothL1Loss=0.00472, MAE=21.9, RegLoss=0, MAE_val=23.3, SmoothL1Loss_val=0.00495]
```

Figure 4.15 NeuralProphet results for Arrest variable

From Figure 4.15, it can be observed that the margin of error between the training set and testing is almost negligible, hence it can be concluded that the NeuralProphet model has performed very well in forecasting for the arrest variable.

ii) Domestic

```
SmoothL1Loss=0.00877, MAE=12.3, RegLoss=0, MAE_val=16.9, SmoothL1Loss_val=0.0146]
```

Figure 4.16 NeuralProphet results for Domestic variable

From Figure 4.16, it can be observed that the margin of error between the training set and testing is quite less, hence it can be concluded that the NeuralProphet model has performed well in forecasting for the domestic variable.

iii) Total Crime Count

```
SmoothL1Loss=0.00309, MAE=53.7, RegLoss=0, MAE_val=62.6, SmoothL1Loss_val=0.00366]
```

Figure 4.17 NeuralProphet results for Total Crime Count variable

From Figure 4.17, it can be observed that the margin of error between the training set and testing is relatively less, hence it can be concluded that the NeuralProphet model has still performed well in forecasting for the total crime count variable.

4.3.6 Final Result

The forecasts by each model for the time series are plotted corresponding to the results obtained in the previous module.

a) LSTM

i) Arrest

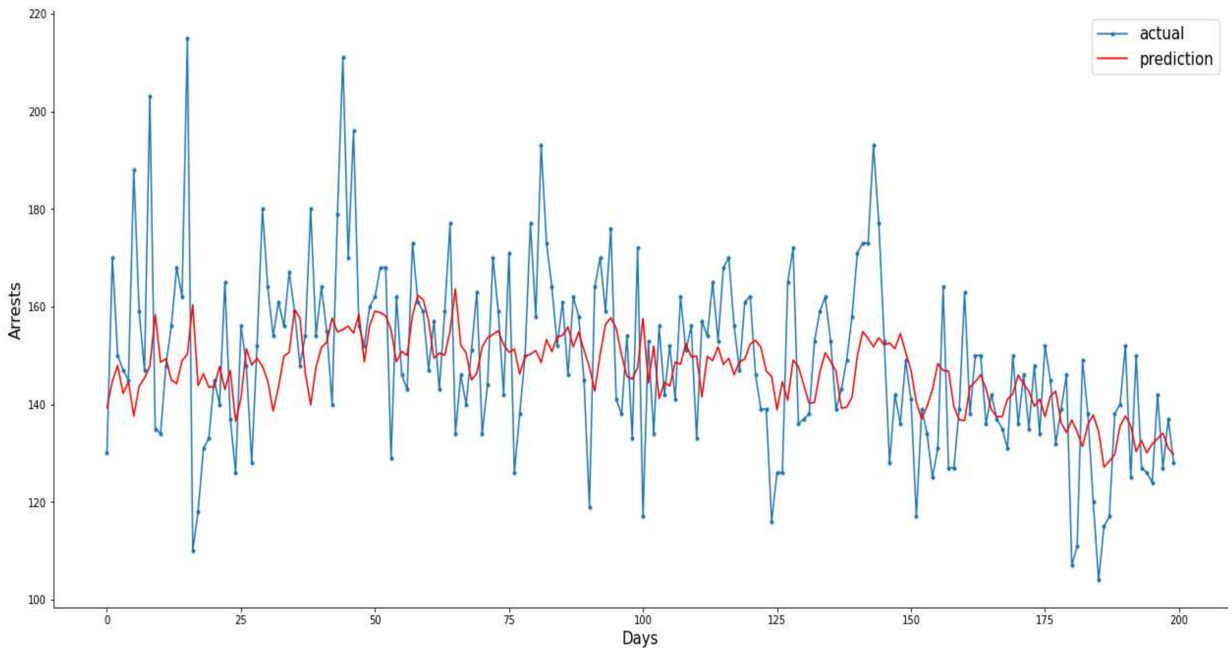


Figure 4.18 LSTM forecast for Arrest variable

ii) Domestic

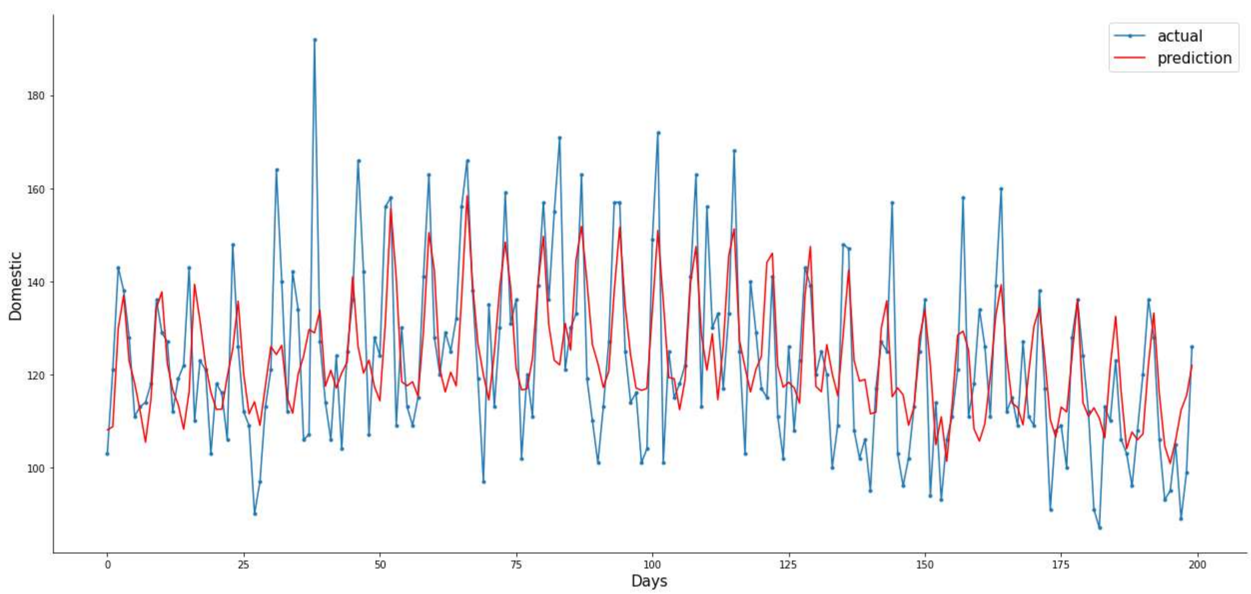


Figure 4.19 LSTM forecast for Domestic variable

iii) Total Crime Count

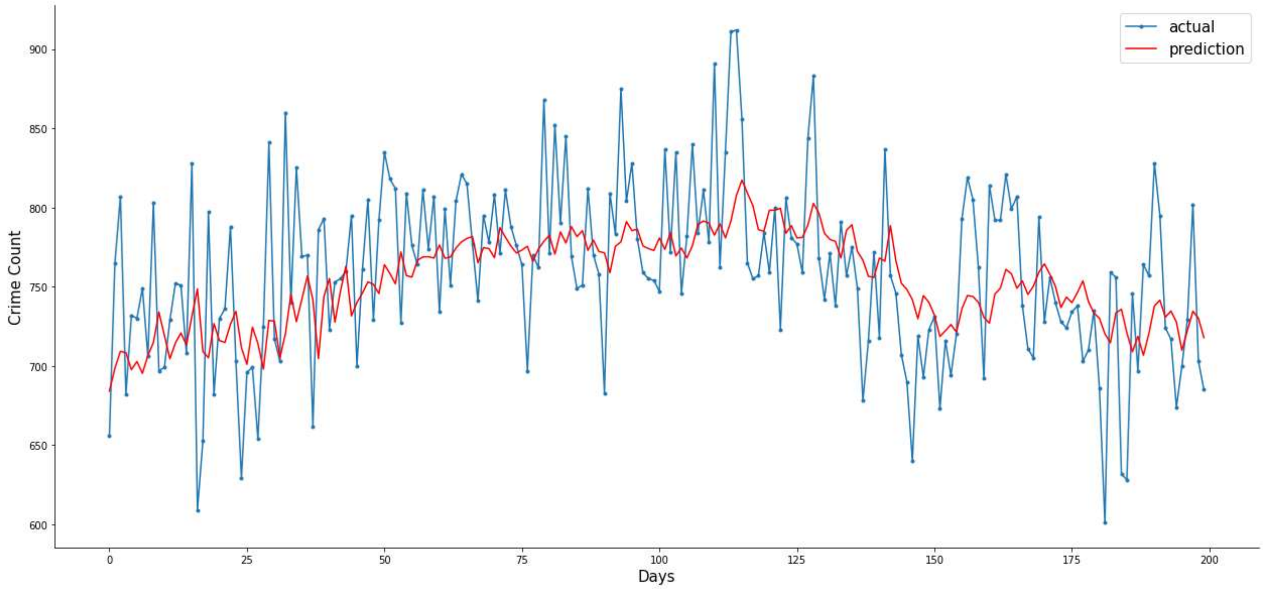


Figure 4.20 LSTM forecast for Total Crime Count variable

As observed from Figures 4.18, 4.19 and 4.20, the forecasted data follow most of the general trends and fluctuations of the original data, especially well for the domestic variable, which indicates that the margin of error between the training set and testing set is very less. Hence it corroborates with the conclusion in the previous section that the LSTM model has performed very well in forecasting for all three variables.

b) ARIMA

i) Arrest

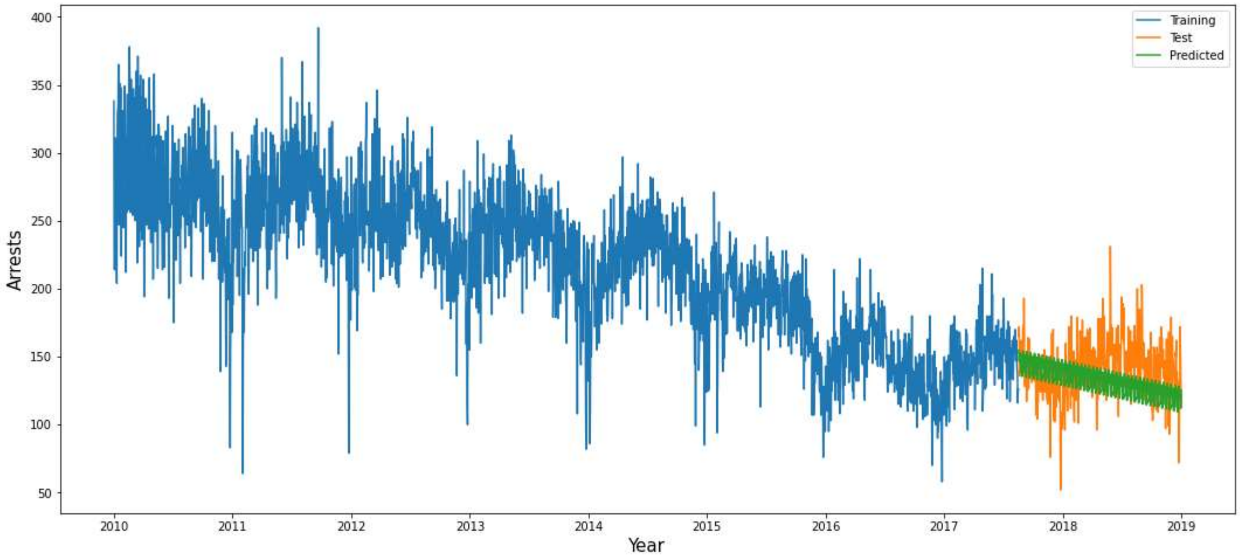


Figure 4.21 ARIMA forecast for Arrest variable

ii) Domestic

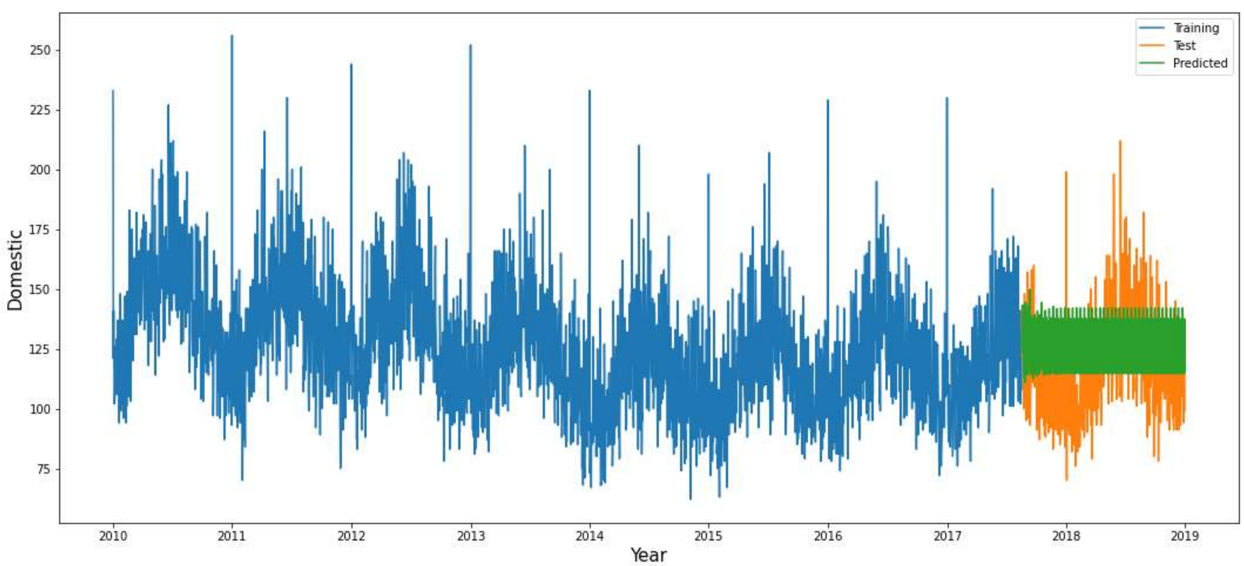


Figure 4.22 ARIMA forecast for Domestic variable

iii) Total Crime Count

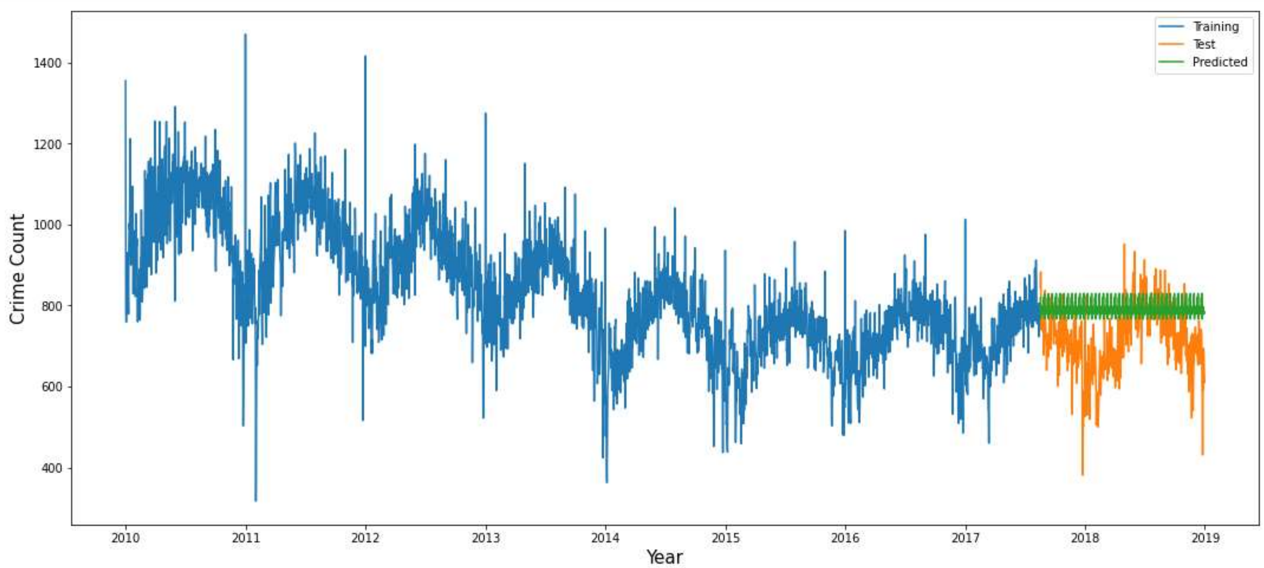


Figure 4.23 ARIMA forecast for Total Crime Count variable

As observed from Figures 4.21, 4.22 and 4.23, the forecasted data highly deviate and far off from the nuances of the original data, in particular for the total crime count variable, which indicates that the margin of error is quite large, hence it corroborates with the conclusion in the previous section that the ARIMA model has performed poorly in forecasting for all three variables.

c) NeuralProphet

i) Arrest

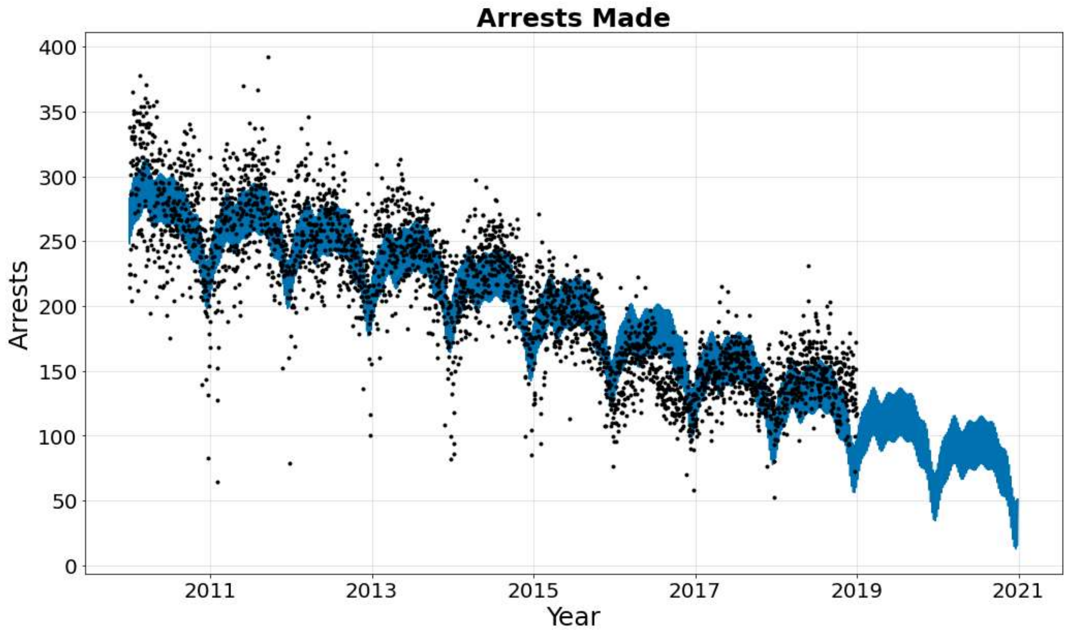


Figure 4.24 NeuralProphet forecast for Arrest variable

ii) Domestic

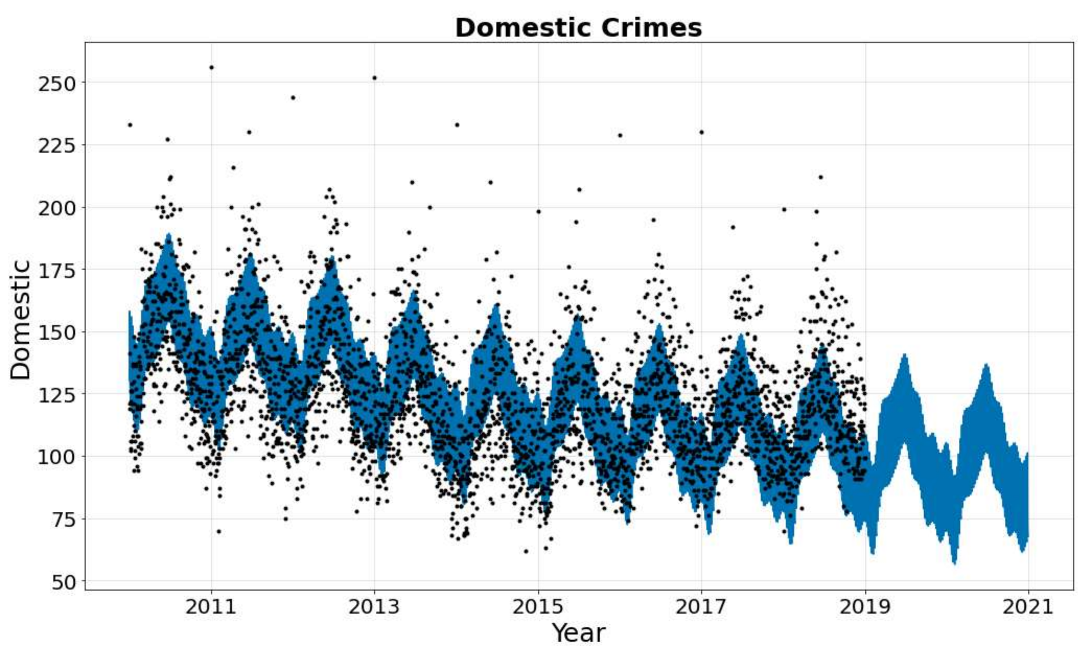


Figure 4.25 NeuralProphet forecast for Domestic variable

iii) Total Crime Count

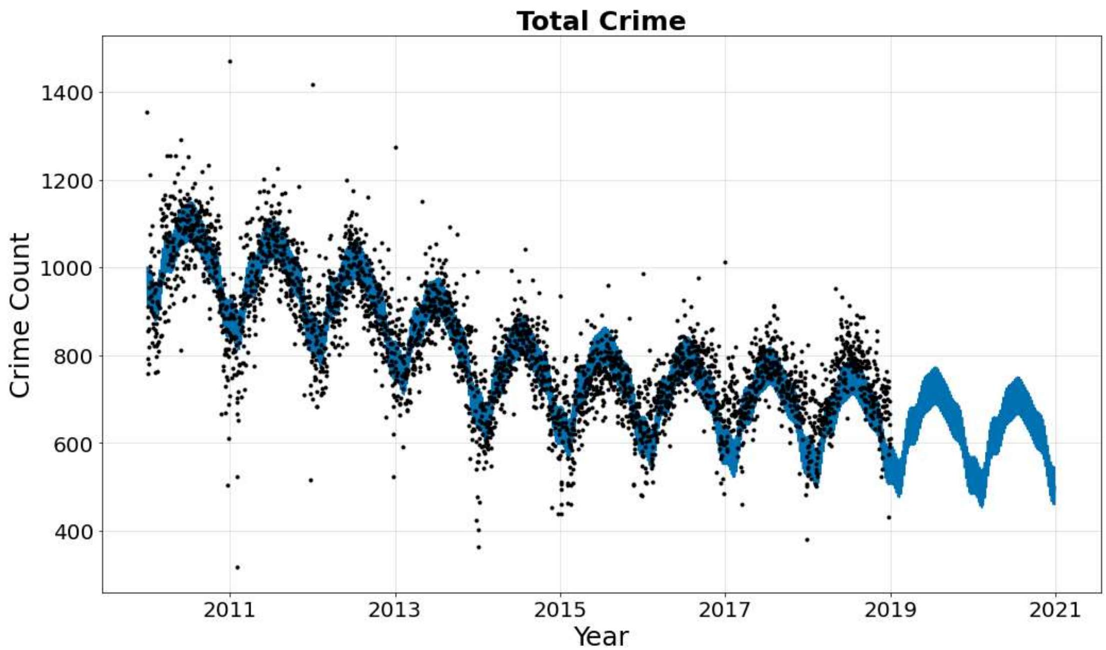


Figure 4.26 NeuralProphet forecast for Total Crime Count variable

As observed from Figures 4.24, 4.25 and 4.26, the forecasted data follow almost all of the general trends and fluctuations of the original data, which indicates that the margin of error between the training set and validation set is the lowest of all the models used thus far. Hence it corroborates with the conclusion in the previous section that the NeuralProphet model has exceptionally well in forecasting for all three variables.

CHAPTER 5

CONCLUSION AND FUTURE WORK

With the help of deep learning models, analysis and forecasting of crime rates has been made easier. This research work involves crime analysis and forecasting. The use of different deep learning models, each having their own parameters and training processes helps in identifying the most optimal model that can be used for forecasting . After using several deep learning models for training and testing, the margin of error between the training set and validation set was the lowest for the NeuralProphet model, hence it can be concluded that the NeuralProphet model has performed the best in forecasting for the time series. This work can be used in predictive policing to assist in crime reduction.

The future work of the project includes incorporating spacio-temporal, economic and weather data of the region concerned with the dataset. Also, the expansion of the area, time period and other features that can be dealt with shall greatly minimize any discrepancies and further enhance the scope of the project.

REFERENCES

[1] A. Rummens, W. Hardyns, and L. Pauwels. The use of predictive analysis in spatiotemporal crime forecasting: Building and testing a model in an urban context. *Applied Geography, vol. 86, pp. 255– 261,* 2017.

[2] Alexander Stec and Diego Klabjan. Forecasting Crime with Deep Learning. *Northwestern University, 2145 Sheridan Road, Evanston, IL 60208 USA,* 2018.

[3] Anish Krishnan, Aditya Sarguru and A. C. Shantha Sheela. Predicitve Analysis of Crime Data using Deep Learning. *International Journal of Pure and Applied Mathematics, Volume 118, No. 20, 4023-4031,* 2018.

[4] Charlie S. Marzan, Maria Jeseca C. Baculo, Remedios de Dios Bulos, and Conrado Ruiz. Time Series Analysis and Crime Pattern Forecasting of City Crime Data. *International Conference on Algorithms, Computing and Systems (ICACS '17),* 2017.

[5] G. Borowik, Z. M. Wawrzyniak and P. Cichosz. Time series analysis for crime forecasting. *2018 26th International Conference on Systems Engineering(ICSEng),Sydney,NSW,Australia,pp.110,10.1109/ICSENG.2018.8638179,* 2018.

[6] Pirajján, Freddy & Fajardo, Andrey & Melgarejo, Miguel. Towards a Deep Learning Approach for Urban Crime Forecasting. *10.1007/978-3-030-31019-6_16,* 2019.

[7] Smith, M. and R. Agrawal. A Comparison of Time Series Model Forecasting Methods on Patent Groups. *MAICS,* 2015.

[8] Srihari Athiyarath, Mousumi Paul and Srivatsa Krishnaswamy. A Comparative Study and Analysis of Time Series Forecasting Techniques. *SN Computer Science. 1. 10.1007/s42979-020-00180-5,* 2020.

[9] Ulises M. Ramirez-Alcocer, Edgar Tello-Leal and Jonathan A. Mata-Torres. Predicting Incidents of Crime through LSTM Neural Networks in Smart City Domain. *SMART 2019 : The Eighth International Conference on Smart Cities, Systems, Devices and Technologies,* 2019.

[10] Wang, Bao & Zhang, Duo & Zhang, Duanhao & Brantingham, P. & Bertozzi, Andrea. Deep Learning for Real Time Crime Forecasting, 2017.

Curiginal

Document Information

Analyzed document	fyp_report.docx (D106452773)
Submitted	5/26/2021 6:33:00 AM
Submitted by	s.sabena
Submitter email	sabenazulficker@gmail.com
Similarity	13%
Analysis address	sabenazulficker.annauniv@analysis.urkund.com

Sources included in the report

W	URL: https://link.springer.com/article/10.1007/s42979-020-00180-5 Fetched: 8/2/2020 7:10:44 PM	6
W	URL: https://ad-publications.cs.uni-freiburg.de/theses/Master_Nishat_Fariha_Rimi_2019.pdf Fetched: 8/5/2020 1:29:04 PM	1
W	URL: https://www.diva-portal.org/smash/get/diva2:1366957/FULLTEXT02 Fetched: 6/6/2020 2:52:52 PM	4
SA	**038_FRP.Gurkaranvir.pdf** Document 038_FRP.Gurkaranvir.pdf (D97461885)	2
W	URL: https://sabrinriaz.com/centrelink-public-4nmcu/arima-model-in-python-kaggle.html Fetched: 1/29/2021 5:00:42 PM	1
W	URL: https://dokumen.pub/machine-learning-for-time-series-forecasting-with-python-9781119682363-9781119682370-9781119682387-2020947403.html Fetched: 1/31/2021 6:13:39 PM	1
W	URL: https://dokumen.pub/applied-neural-networks-with-tensorflow-2-api-oriented-deep-learning-with-python-1st-ed-9781484265123-9781484265130.html Fetched: 5/21/2021 4:47:24 PM	1
SA	**Thesis report_Shruti Chouksey.docx** Document Thesis report_Shruti Chouksey.docx (D51990728)	1
SA	**Thesis presentation_Shruti Chouksey_10389935.pptx** Document Thesis presentation_Shruti Chouksey_10389935.pptx (D52304141)	1
W	URL: http://etd.lib.metu.edu.tr/upload/12625526/index.pdf Fetched: 1/4/2021 6:34:31 PM	1
W	URL: https://www.hindawi.com/journals/scn/2021/5587511/ Fetched: 5/20/2021 9:37:44 AM	2

Entire Document

CRIME FORECASTING USING DEEP LEARNING

iii

ABSTRACT

Crime is a significant threat to humankind. There are many crimes that happen at regular intervals of time. Perhaps it is increasing and spreading at an alarming rate. Crimes happen from small villages and towns to big cities. Criminal incidents have increased at an alarming rate and it is the responsibility of the police department to control and reduce the occurrence of crime. An important step taken by police departments in the effort to reduce crime is the use of predictive policing. Predictive policing involves using algorithms to analyze massive amounts of information in order to predict and help prevent potential future crimes. The proposed approach is based on crime data analysis using deep learning methods to forecast crime rates.

1

CHAPTER 1

INTRODUCTION

Police departments in some of the largest U.S. cities have been experimenting with predictive policing as a way to forecast criminal activity. Predictive policing involves using algorithms to analyze massive amounts of information in order to predict and help prevent potential future crimes. Place-based predictive policing, the most widely practiced method, typically uses pre-existing crime data to identify places and times that have a high risk of crime. Predictive policing

uses computer algorithms to predict future crimes and can assist in addition to instincts of police officers. It can also be a more cost-efficient method to reduce criminal incidents.

In the proposed approach, the recurrent neural network models, Long short-term memory (LSTM) and NeuralProphet and the mathematical model, Autoregressive integrated moving average (ARIMA) shall be used for deep learning. The main idea is to obtain a crime dataset concerned with a particular region over a period of time and transform the dataset into ordered time series with specific target variables. The time series shall then be split into training and testing sets and trained with the models specified previously. After training is completed for each model, forecasts shall then made for each target variable and the performance of each model shall be analyzed based on the margin of error between the training and testing set.

2

1.1 MOTIVATION

There has been a rampant increase in criminal incidents worldwide.

The motivation for the development of the project is providing automatically an efficient way to analyze and

forecast crime rates. The proposed solution is based on deep

learning techniques and can make the case-solving and investigation process easier and faster.

1.2 PROBLEM STATEMENT

A research paper may belong to more than one category at the same time. In existing literature, many deep learning methods have been applied for forecasting crime rates. But there is a limitation in identifying patterns and factors of crime incidents over a period of time. Advancements in deep learning algorithms can traverse various datasets and reveal new information.

1.3 OBJECTIVE

The project aims to find an approachable and efficient method for forecasting of crime rates.

1.4 ORGANIZATION OF THE REPORT

The project report is organized as follows, chapter 2 discusses the existing systems and various methods required for the proposed system. chapter 3 discusses the various concepts used in the proposed system along with

the

overall system architecture. chapter 4 discusses the implementation details of the proposed system along with the necessary algorithms

and the experimental results chapter 5 concludes the report by summarizing the total result and proposes possible enhancements that can be done in the future.

4

CHAPTER 2

LITERATURE SURVEY

A literature survey is done by surveying research papers. The limitations and the knowledge gained from the papers will help us to create a better system.

2.1

100%	MATCHING BLOCK 1/21	W	https://link.springer.com/article/10.1007/s429 ...

A COMPARATIVE STUDY AND ANALYSIS OF TIME SERIES FORECASTING TECHNIQUES

This paper proposes to perform

92%	MATCHING BLOCK 5/21	W	https://link.springer.com/article/10.1007/s429 ...

three different types of time series forecasting, namely short-period fore-casting, medium-period forecasting and long-period forecasting [1]. Short-period forecasting refers to a time frame that ranges from a few days to a couple of weeks, medium period refers to a few months, and the long period considers a time period of more than a couple of years.

They

85%	MATCHING BLOCK 2/21	W	https://link.springer.com/article/10.1007/s429 ...

have applied regression, LSTM, ARIMA, CNN, fuzzy-based method weighted MVFTS and CBLSTM on three discrete datasets. The performance of the above-mentioned methods for short-period forecasting, over all the three different datasets have shown the best results by LSTMs, CNNs and weighted MVFTS and CBLSTM. ARIMA too performed significantly well; however, regression was not able to perform well.

68%	MATCHING BLOCK 3/21	W	https://link.springer.com/article/10.1007/s429 ...

In the case of medium-period forecasting, regression continued to fail and perform poorly whereas ARIMA and weight MVFTS and CBLSTM performed extraordinarily well and likewise CNN and LSTMs did moderately well.

82%	MATCHING BLOCK 4/21	W	https://link.springer.com/article/10.1007/s429 ...

Finally, in long-period forecasting, over all the three different datasets have shown that the best 5 results were similar to medium period, ARIMA, weighted MVFTS, CBLSTM and CNN consistently performed well. From the very high RMSE and R2 values, it was evident that regression was the most ineffective method to use. 2.2

FORECASTING CRIME WITH DEEP LEARNING

This paper proposes to use deep neural networks, including variations that are suited to the spatial and temporal aspects of the crime prediction problem, in order to make next day crime count predictions in a fine-grain city partition [2]. The crime counts were broken into 10 bins and the model predicted the most likely bin for each spatial region at a daily level. Predictions were made using Chicago and Portland crime data, which were augmented with additional datasets covering weather, census data, and public transportation. The models include a Feed forward network, a Convolutional network, a Recurrent network and a Recurrent convolutional network. With their best model, they were able to predict the correct bin for overall crime count with 75.6% and 65.3% accuracy for Chicago and Portland, respectively.

2.3 PREDICTING INCIDENTS OF CRIME THROUGH LSTM

NEURAL NETWORKS IN SMART CITY DOMAIN

In this research work [3], a deep learning based approach is proposed for the classification of incidents of a crime of public safety through predictive analysis. The predictive model is based on a neural network Long Short-Term Memory 6

(LSTM), trained with a small group of attributes, enabling the prediction of the class label in the validation stage, with a high percentage of prediction accuracy. The proposed approach is evaluated through a big data set (real data) of type open data, which contains historical information about the crimes of a smart city. Their deep learning approach achieved high performance in the final model with 87.84% accuracy based on the validation data. Furthermore, the final LSTM model achieved an average loss function of 0.0376 on validation data, using 20% of the data set for the testing stage.

2.4 A COMPARISON OF TIME SERIES MODEL FORECASTING

METHODS ON PATENT GROUPS

This paper proposes to create a technology forecasting model based on the sequence of patents issued over a given time period [4]. The focus was to apply time series modeling techniques to a collection of USPTO patents from 1996 to 2013. The techniques used were Holt-Winters Exponential Smoothing and Autoregressive Integrated Moving Averages (ARIMA). Cross validation methods were used to determine the best fitting models and ultimately whether or not patent data could be modeled as a time series. For each model and 15-month forecast, four error statistics were calculated:

93%	MATCHING BLOCK 6/21	W	https://ad-publications.cs.uni-freiburg.de/the ...

Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), and

Mean Absolute Scaled Error (MASE). According to the experimental results, the MASE values indicated that all of the models had adequate forecasting capabilities. The results suggested that ARIMA acted as a better predictor for the GCSSA and DFMDS data while the AI patent data seemed to be better suited for an Exponential Smoothing model. 7

2.5 PREDICITVE ANALYSIS OF CRIME DATA USING DEEP

LEARNING

In this work [5], crime is predicted using Recurring LSTM networks. An existing structured dataset was used to predict the crimes. The data was split into training and testing data. Both the testing and training undergo the training process. The resultant training and testing data was compared with the actual crime count and was visualized. The obtained output was compared with an existing system. The accuracy of this model was compared with an existing model and concluded that the LSTM model provided better efficiency than the existing model.

8

CHAPTER 3

SYSTEM DESIGN

3.1 PROPOSED SYSTEM ARCHITECTURE

The proposed system follows a tightly coupled architecture comprising of the following modules.

• Dataset • Data

preprocessing • Time series transformation • Data preparation • Model forecasting • Model Evaluation

3.2 ARCHITECTURE DIAGRAM

Figure 3.1 System Architecture

9 3.3

MODULE DESCRIPTION

The description of the modules is mentioned below.

3.3.1 Dataset

The most important part of any deep learning based project is the dataset. The dataset used in this project reflects reported incidents of crime that occurred in the city of Chicago from January 1, 2010 to December 31, 2018. The dataset is obtained from the official Chicago Data Portal (https://data.cityofchicago.org/).

3.3.3

Time Series Transformation

In this module, a datetime object is created in the format YYYY/MM/DD in order to transform the dataframe into ordered time series data for forecasting. The target variables to be used for forecasting are arrest, domestic and total crime count. Therefore, three separate univariate time series are created. All the other attributes are dropped except the date and respective target variables of the series to be created. The dataframes are grouped by the required value of the target variables and hence count of each target variable corresponding to every date from 2010-2018 are obtained. Finally, the date attribute is set as the index to form the ordered time series.

12 3.3.4 Data Preparation and

Model Forecasting

In this module, the initial step is to prepare the data for training. Firstly, the

time series created are checked for stationarity using Augmented Dickey-Fuller test. The time series are

then split into training set and testing set in the ratio of 80% and 20%.

They are then converted into a NumPy array and the values are scaled using a scaler if required for the deep learning accordingly. After the parameters of each model are specified respectively, the models are fit on the training sets and trained. After training, the models are applied on the testing sets and the training and testing mean square error/mean absolute error are obtained. This process is performed individually for each model and each time series.

The following deep learning models are used for training and testing:

i) Long Short-Term Memory (LSTM) ii) Autoregressive integrated moving average (ARIMA) iii) NeuralProphet

3.3.5 Model Evaluation

After the training and testing mean square error/mean absolute error are obtained for each model and time series, the model parameters are adjusted in order to reduce the margin of error and overfitting/underfitting as much as possible.

13 3.3.6 Final Result

The final forecast of all three variables of each model is plotted.

14

CHAPTER 4

IMPLEMENTATION AND RESULTS

4.1

TOOLS USED

The following tools, libraries and environments are used in this project.

4.1.1 Pandas

Pandas is a software library written for the Python programming language for data manipulation and analysis.

91%	MATCHING BLOCK 7/21	W	https://www.diva-portal.org/smash/get/diva2:13 ...

In particular, it offers data structures and operations for manipulating numerical tables and time series. 4.1.2 NumPy NumPy is

a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a

94%	MATCHING BLOCK 9/21	W	https://www.diva-portal.org/smash/get/diva2:13 ...

large collection of high-level mathematical functions to operate on these arrays. 4.1.3 Matplotlib Matplotlib is a plotting library

for the Python programming language and its numerical mathematics extension NumPy. It is used for creating static, animated, and interactive visualizations in Python.

15 4.1.4

100%	MATCHING BLOCK 8/21	W	https://www.diva-portal.org/smash/get/diva2:13 ...

Seaborn Seaborn is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics. 4.1.5

78%	MATCHING BLOCK 10/21	SA	038_FRP.Gurkaranvir.pdf (D97461885)

Scikit-learn Scikit-learn is a free software machine learning library for the Python

programming language. It features various classification, regression and clustering

and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy.

4.1.6

Keras

Keras is an open-source software

94%	MATCHING BLOCK 11/21	SA	038_FRP.Gurkaranvir.pdf (D97461885)

library that provides a Python interface for artificial neural networks. Keras acts as an interface for the TensorFlow library. 4.1.7

Pmdarima

100%	MATCHING BLOCK 12/21	W	https://sabrinriaz.com/centrelink-public-4nmcu ...

Pmdarima (originally pyramid-arima, for the anagram of 'py' + 'arima') is a statistical library designed to fill the void in Python's time series analysis capabilities. 4.1.8

Plotly

Plotly is an

interactive, open-source, and browser-based graphing library for Python.

4.1.9

100%	MATCHING BLOCK 13/21	W	https://dokumen.pub/machine-learning-for-time- ...

Statsmodels Statsmodels is a Python module that provides classes and functions for the estimation of many different statistical models, as well as for conducting statistical tests, and statistical data exploration. 4.1.10

NeuralProphet

NeuralProphet is a python library for modeling time-series data based on neural networks. It's built on top of PyTorch and is heavily inspired by Facebook Prophet and AR-Net libraries.

4.1.11

Livelossplot

Livelossplot is a library to plot live training in Jupyter Notebook for Keras, PyTorch and other frameworks. It is an open-source Python package that can be used to observe each epoch of training.

4.1.12

Google Colab Notebooks

50%	MATCHING BLOCK 14/21	W	https://dokumen.pub/applied-neural-networks-wi ...

Colaboratory, or "Colab" for short, is a product from Google Research. Colab allows anybody to write and execute arbitrary python code through

the browser, and is especially well suited to machine learning, data analysis and education.

18 4.2 MODELS USED

4.2.1

Long short-term memory

82%	MATCHING BLOCK 15/21	SA	Thesis report_Shruti Chouksey.docx (D51990728)

Long short-term memory (LSTM) is an artificial recurrent neural network (RNN) architecture used in the field of deep learning.

LSTM networks are well-suited to classifying, processing and making predictions based on time series data, since there can be lags of unknown duration between important events in a time series.

4.2.2 Autoregressive Integrated Moving Average

An Autoregressive Integrated Moving Average (ARIMA) model

100%	MATCHING BLOCK 16/21	SA	Thesis presentation_Shruti Chouksey_10389935.pptx (D52304141)

is a class of statistical models for analyzing and forecasting time series data. It explicitly caters to a suite of standard structures in time series data,

and as such provides a simple yet powerful method for making skillful time series forecasts.

4.2.3

NeuralProphet

NeuralProphet is a new framework that extends on the original Prophet framework, addresses pain points such as scale, customization, and extensibility. It incorporates traditional statistical and neural network models for time series modeling, used in forecasting and anomaly detection.

4.3

PROJECT WORK IMPLEMENTATION

4.3.1

Getting Started

In the initial step, the libraries and tools mentioned previously are imported in the Colab notebook. The dataset to be used for the project is then retrieved and imported and a data frame is created

as shown in Figure 4.1.

Figure 4.1 Data frame

21 4.3.2 Data Preprocessing

The dataset is checked for null and duplicate values and dropped. Inconsistencies in attribute names are handled and primary key type attributes are removed as they

are of

no use for analysis. The date column is converted to datetime object to get the day of the week, month and year of the crime for

creation of time series. The attributes that are not required for time series creation are dropped. The crimes are mapped under one general group of crime and by location.

Figure 4.2 Dataset after preprocessing

4.3.3 Time Series Transformation

To assist training using deep learning models, the data is to be transformed into ordered time series. The variables arrest, domestic and total crime count each

22 are used as target variables and individual time series are created. This is done with the help of the datetime object that is already created and by using groupby() and count() functions of Python.

Figure 4.3 Time series created

4.3.4 Data Preparation and Model Forecasting

Firstly, the time series created are checked for stationarity using Augmented Dickey-Fuller test.

82%	MATCHING BLOCK 19/21	W	http://etd.lib.metu.edu.tr/upload/12625526/ind ...

A stationary time series is one whose properties do not depend on the time

at which the series

is observed.

88%	MATCHING BLOCK 17/21	W	https://www.diva-portal.org/smash/get/diva2:13 ...

Figure 4.4 Augmented Dickey-Fuller test From Figure 4.4, it can be observed that the p-value is less than 0.05

and the null hypothesis that there is a unit root, is rejected for all the time series. Hence it can be concluded the time series created are stationary.

Figure 4.5 Data preparation for LSTM model

As shown in Figure 4.5, for LSTM model, the time series are converted into a NumPy array and the values are fitted after being scaled using a MinMax scaler. The time series are

then split into training set and testing set in the ratio of 80% and 20%

respectively for training. The model parameters are specified and the model is fitted and trained on the training set and validated on the testing set as shown in Figure 4.6.

Figure 4.6 Training using LSTM model

25 The entire process is followed for all the time series that are created, for training using LSTM model.

For training using ARIMA model, the time series are

split into training set and testing set in the ratio of 80% and 20%

respectively. The auto_arima function is then used to fit the ARIMA model with the lowest AIC value by performing a stepwise search with several different combinations of the specified parameters. After the model is fitted, predictions are made on the test set. This process is followed for all the time series that are created, for training using ARIMA model.

Figure 4.7 Training using ARIMA model

26 For training using NeuralProphet model, the model is fitted on the time series data with the specified parameters. After the model is fitted, predictions are made on the validation set.

Figure 4.8 Training using NeuralProphet model

4.3.5 Model Evaluation

The following results are obtained after training and testing each model on the time series. The respective model parameters are adjusted in order to reduce the margin of error and overfitting/underfitting as much as possible.

27 a) LSTM

i) Arrest

Figure 4.9 LSTM results for Arrest variable From Figure 4.9, it can be observed

that

the margin of error between the training set and testing set is very less, hence it can be concluded that the LSTM model has performed reasonably well

in forecasting for the arrest variable.

ii) Domestic

Figure 4.10 LSTM results for Domestic variable From Figure 4.10, it can be observed

that

the margin of error between the training set and testing set is almost none, hence it can be concluded that the LSTM model has performed excellently in forecasting for the domestic variable.

iii) Total Crime Count

Figure 4.11 LSTM results for Total Crime Count

variable

28 From Figure 4.11, it can be observed

that

the margin of error between the training set and testing set is relatively less, hence it can be concluded that the LSTM model has performed well in forecasting for the

total crime count variable.

b) ARIMA

i) Arrest

Figure 4.12 ARIMA results for Arrest variable From Figure 4.12, it can be observed that

the margin of error between the testing set and the forecast is large, hence it can be concluded that the ARIMA model has performed poorly in forecasting for the arrest variable.

ii) Domestic

Figure 4.13 ARIMA results for Domestic variable From Figure 4.13, it can be observed that the margin of error between the testing set and the forecast is as large as the arrest variable,

hence it can be concluded that the ARIMA model has again performed poorly in forecasting for the domestic variable. 29 iii) Total Crime Count

Figure 4.14 ARIMA results for Total Crime Count variable

From Figure 4.14, it can be observed that

the margin of error between the testing set and the forecast is huge, hence it can be concluded that the ARIMA model has performed

the worst in forecasting for the total crime count variable.

c) NeuralProphet

i) Arrest

Figure 4.15 NeuralProphet results for Arrest variable

From Figure 4.15, it can be observed

that

the margin of error between the training set and testing is almost negligible, hence it can be concluded that the NeuralProphet model has performed very well

in forecasting for the arrest variable.

ii) Domestic

Figure 4.16 NeuralProphet results for Domestic variable 30 From Figure 4.16, it can be observed

that

the margin of error between the training set and testing is quite less, hence it can be concluded that the NeuralProphet model has performed well in forecasting for the domestic variable.

iii) Total Crime Count

Figure 4.17 NeuralProphet results for Total Crime Count

variable

From Figure 4.17, it can be observed

that

the margin of error between the training set and testing is relatively less, hence it can be concluded that the NeuralProphet model has still performed well

in forecasting for the

total crime count variable.

4.3.6

Final Result

The forecasts by each model for the time series are plotted corresponding to the results obtained in the previous module.

31 a) LSTM

i) Arrest

Figure 4.18 LSTM forecast for Arrest variable

ii) Domestic

Figure 4.19 LSTM forecast for Domestic variable

iii) Total Crime Count

Figure 4.20 LSTM forecast for Total Crime Count variable

As observed from Figures 4.18, 4.19 and 4.20,

the forecasted data follow most of the general trends and fluctuations of the original data,

especially well for the domestic variable,

which indicates that the margin of error between the training set and testing set is very less. Hence it corroborates with the conclusion in the previous section that the LSTM model has performed very well in forecasting for all three variables.
b) ARIMA

i) Arrest

32

Figure 4.21 ARIMA forecast for Arrest variable

ii) Domestic

32 iii) Total Crime Count

32 iii) Total Crime Count

Figure 4.22 ARIMA forecast for Domestic variable

iii) Total Crime Count

Figure 4.23 ARIMA forecast for Total Crime Count variable 33 As observed from Figures 4.21, 4.22 and 4.23, the forecasted data highly deviates and is far off from the nuances of the original data, in particular for the total crime count variable,

which indicates that the margin of error is quite large, hence it corroborates with the conclusion in the previous section that the ARIMA model has performed poorly

in forecasting for all three variables. c) NeuralProphet

i) Arrest

Figure 4.24 NeuralProphet forecast for Arrest variable

ii) Domestic

34 Figure 4.25 NeuralProphet forecast for Domestic variable

iii) Total Crime Count

Figure 4.26 NeuralProphet forecast for Total Crime Count variable

As observed from Figures 4.24, 4.25 and 4.26,

the forecasted data follows almost all of the general trends and fluctuations of the original data, which indicates that the margin of error between the training set and validation set is the lowest of all the models used thus far. Hence it corroborates with the conclusion in the previous section that the NeuralProphet model has exceptionally well in forecasting for all three variables.

35

CHAPTER 5

CONCLUSION AND FUTURE WORK

With the help of deep learning models, analysis and forecasting of crime rates has been made easier. This research work involves crime analysis and forecasting. The use of different deep learning models, each having their own parameters and training processes helps in identifying the most optimal model that can be used for forecasting . After using several deep learning models for training and testing, the margin of error between the training set and validation set was the lowest for the NeuralProphet model, hence it can be concluded that the NeuralProphet model has performed the best

in forecasting

for the time series. This work can be used in predictive policing

to assist in

crime reduction.

The future work of the project includes incorporating spacio-temporal, economic and weather data of the region concerned with the dataset. Also, the expansion of the area, time period and other features that can be dealt with shall greatly minimize any discrepancies and further enhance the scope of the project.

36

REFERENCES

[1] Srihari Athiyarath, Mousumi Paul and Srivatsa

Krishnaswamy, "A Comparative Study and Analysis of Time Series Forecasting Techniques", SN

Computer Science. 1. 10.1007/s42979-020-00180-5, 2020.

[2] Alexander Stec and Diego Klabjan, “Forecasting Crime with Deep Learning”, Northwestern University, 2145 Sheridan Road, Evanston, IL 60208 USA, 2018.

[3] Ulises M. Ramirez-Alcocer, Edgar Tello-Leal and Jonathan A. Mata-Torres, "Predicting Incidents of Crime through LSTM Neural Networks in Smart City Domain", SMART 2019 : The Eighth International Conference on Smart Cities, Systems, Devices and Technologies, 2019 .

[4] Smith, M. and R. Agrawal, “A Comparison of Time Series Model Forecasting Methods on Patent Groups”, MAICS, 2015.

[5] Anish Krishnan, Aditya Sarguru and A. C. Shantha Sheela, "Predicitve Analysis of Crime Data using Deep Learning", International Journal of Pure and Applied Mathematics, Volume 118, No. 20, 4023-4031, 2018.

[6] A. Rummens, W. Hardyns, and L. Pauwels, “The use of predictive analysis in spatiotemporal crime forecasting: Building and testing a model in an urban context”, Applied Geography, vol. 86, pp. 255– 261, 2017.

[7]

G. Borowik, Z. M. Wawrzyniak and P. Cichosz, "Time series analysis for crime forecasting", 2018 26th International Conference on Systems Engineering (ICSEng), Sydney, NSW, Australia,

pp. 1-10, doi: 10.1109/ICSENG.2018.8638179, 2018.

[8] Charlie

73%	MATCHING BLOCK 21/21	W	https://www.hindawi.com/journals/scn/2021/5587511/

S. Marzan, Maria Jeseca C. Baculo, Remedios de Dios Bulos, and Conrado Ruiz, “Time Series Analysis and Crime Pattern Forecasting of City Crime Data”, International Conference on Algorithms, Computing and Systems (

ICACS '17), 2017.

37

[9] Pirajàn, Freddy & Fajardo, Andrey & Melgarejo, Miguel, “Towards a Deep Learning Approach for Urban Crime Forecasting”, 10.1007/978-3-030-31019-6_16, 2019.

[10] Wang, Bao & Zhang, Duo & Zhang, Duanhao & Brantingham, P. & Bertozzi, Andrea, “Deep Learning for Real Time Crime Forecasting”, 2017.

Curiginal

Hit and source - focused comparison, Side by Side

Submitted text	As student entered the text in the submitted document.
Matching text	As the text appears in the source.

1/21	SUBMITTED TEXT 10 WORDS	100% MATCHING TEXT 10 WORDS
	A COMPARATIVE STUDY AND ANALYSIS OF TIME SERIES FORECASTING TECHNIQUES	A Comparative Study and Analysis of Time Series Forecasting Techniques |

W https://link.springer.com/article/10.1007/s42979-020-00180-5

5/21	SUBMITTED TEXT 57 WORDS	92% MATCHING TEXT 57 WORDS
	three different types of time series forecasting, namely short-period fore-casting, medium-period forecasting and long-period forecasting [1]. Short-period forecasting refers to a time frame that ranges from a few days to a couple of weeks, medium period refers to a few months, and the long period considers a time period of more than a couple of years.	three different types of time series forecasting, namely short-period forecasting, medium-period forecasting and long-period forecasting. Short-period forecasting refers to a time frame that ranges from a few days to a couple of weeks, medium period refers to a few months, and the long period considers a time period of more than a couple of years.

W https://link.springer.com/article/10.1007/s42979-020-00180-5

2/21	SUBMITTED TEXT 58 WORDS	85% MATCHING TEXT 58 WORDS
	have applied regression, LSTM, ARIMA, CNN, fuzzy-based method weighted MVFTS and CBLSTM on three discrete datasets. The performance of the above-mentioned methods for short-period forecasting, over all the three different datasets have shown the best results by LSTMs, CNNs and weighted MVFTS and CBLSTM. ARIMA too performed significantly well; however, regression was not able to perform well.	have applied regression, LSTM, ARIMA, CNN, fuzzy-based method weighted MVFTS and CBLSTM on all these three discrete datasets and observed the following. The performance of the above-mentioned methods for short-period forecasting, over all the three different datasets show that the best results by LSTMs, CNNs and weighted MVFTS and CBLSTM. ARIMA too performed significantly well; however, regression was not able to perform well.

W https://link.springer.com/article/10.1007/s42979-020-00180-5

3/21	SUBMITTED TEXT 32 WORDS	68% MATCHING TEXT 32 WORDS
	In the case of medium-period forecasting, regression continued to fail and perform poorly whereas ARIMA and weight MVFTS and CBLSTM performed extraordinarily well and likewise CNN and LSTMs did moderately well.	In the case of medium-period forecasting, it is clear from Tables 1, 2, 3 well as the graphs that regression continues to fail and perform poorly where are ARIMA and weight MVFTS and CBLSTM performed extraordinarily well and likewise CNN and LSTMs do moderately well (

W https://link.springer.com/article/10.1007/s42979-020-00180-5

4/21	SUBMITTED TEXT	100 WORDS	82%	MATCHING TEXT	100 WORDS

Finally, in long-period forecasting, over all the three different datasets have shown that the best 5 results were similar to medium period, ARIMA, weighted MVFTS, CBLSTM and CNN consistently performed well. From the very high RMSE and R2 values, it was evident that regression was the most ineffective method to use. 2.2

Finally, in long-period forecasting, over all the three different datasets show that the best results were similar to medium period, ARIMA, weighted MVFTS, CBLSTM and CNN consistently performed well. From the very high RMSE and R2 values, it is evident that regression is the most ineffective method to use (

W https://link.springer.com/article/10.1007/s42979-020-00180-5

6/21	SUBMITTED TEXT	16 WORDS	93%	MATCHING TEXT	16 WORDS

Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), and

root mean squared error (RMSE), the mean absolute error (MAE), the mean absolute percentage error (MAPE) and

W https://ad-publications.cs.uni-freiburg.de/theses/Master_Nishat_Fariha_Rimi_2019.pdf

7/21	SUBMITTED TEXT	25 WORDS	91%	MATCHING TEXT	25 WORDS

In particular, it offers data structures and operations for manipulating numerical tables and time series. 4.1.2 NumPy NumPy is

In specific, it offers data structures and operations for manipulating numerical tables and time series [44]. Numpy Numpy is

W https://www.diva-portal.org/smash/get/diva2:1366957/FULLTEXT02

9/21	SUBMITTED TEXT	34 WORDS	94%	MATCHING TEXT	34 WORDS

large collection of high-level mathematical functions to operate on these arrays. 4.1.3 Matplotlib Matplotlib is a plotting library

large collection of high-level mathematical functions to operate on these arrays [28]. Matplotlib Matplotlib is a Python 2D plotting library

W https://www.diva-portal.org/smash/get/diva2:1366957/FULLTEXT02

8/21	SUBMITTED TEXT	41 WORDS	100%	MATCHING TEXT	41 WORDS

Seaborn Seaborn is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics. 4.1.5

Seaborn Seaborn is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics [46].

W https://www.diva-portal.org/smash/get/diva2:1366957/FULLTEXT02

Ouriginal

10/21	SUBMITTED TEXT	25 WORDS	78% MATCHING TEXT	25 WORDS

Scikit-learn Scikit-learn is a free software machine learning library for the Python

Scikit-learn – Sk-learn is a free machine learning library for the Python.

SA 038_FRP.Gurkaranvir.pdf (D97461885)

11/21	SUBMITTED TEXT	25 WORDS	94% MATCHING TEXT	25 WORDS

library that provides a Python interface for artificial neural networks. Keras acts as an interface for the TensorFlow library. 4.1.7

library that provides a Python interface for working with artificial neural networks. Keras acts as an interface for the TensorFlow library.

SA 038_FRP.Gurkaranvir.pdf (D97461885)

12/21	SUBMITTED TEXT	31 WORDS	100% MATCHING TEXT	31 WORDS

Pmdarima (originally pyramid-arima, for the anagram of 'py' + 'arima') is a statistical library designed to fill the void in Python's time series analysis capabilities. 4.1.8

Pmdarima (originally pyramid-arima, for the anagram of 'py' + 'arima') is a statistical library designed to fill the void in Python's time series analysis capabilities.

W https://sabrinriaz.com/centrelink-public-4nmcu/arima-model-in-python-kaggle.html

13/21	SUBMITTED TEXT	32 WORDS	100% MATCHING TEXT	32 WORDS

Statsmodels Statsmodels is a Python module that provides classes and functions for the estimation of many different statistical models, as well as for conducting statistical tests, and statistical data exploration. 4.1.10

Statsmodels: Statsmodels is a Python module that provides classes and functions for the estimation of many different statistical models as well as for conducting statistical tests and statistical data exploration.

W https://dokumen.pub/machine-learning-for-time-series-forecasting-with-python-9781119682363-978111 ...

14/21	SUBMITTED TEXT	23 WORDS	50% MATCHING TEXT	23 WORDS

Colaboratory, or "Colab" for short, is a product from Google Research. Colab allows anybody to write and execute arbitrary python code through

Colaboratory, or Colab for short, is a Google product, which allows developers to write and execute Python code through

W https://dokumen.pub/applied-neural-networks-with-tensorflow-2-api-oriented-deep-learning-with-pyt ...

Ouriginal

15/21	SUBMITTED TEXT	20 WORDS	82% MATCHING TEXT	20 WORDS

Long short-term memory (LSTM) is an artificial recurrent neural network (RNN) architecture used in the field of deep learning.

Long short-term memory) (Gulli, A. and Pal, is an artificial recurrent neural network (RNN) architecture used in the field of deep learning.

SA Thesis report_Shruti Chouksey.docx (D51990728)

16/21	SUBMITTED TEXT	27 WORDS	100% MATCHING TEXT	27 WORDS

is a class of statistical models for analyzing and forecasting time series data. It explicitly caters to a suite of standard structures in time series data,

is a class of statistical models for analyzing and forecasting time series data. It explicitly caters to a suite of standard structures in time series data.

SA Thesis presentation_Shruti Chouksey_10389935.pptx (D52304141)

19/21	SUBMITTED TEXT	15 WORDS	82% MATCHING TEXT	15 WORDS

A stationary time series is one whose properties do not depend on the time

A stationary time series is the one whose statistical do not depend on the time,

W http://etd.lib.metu.edu.tr/upload/12625526/index.pdf

17/21	SUBMITTED TEXT	20 WORDS	88% MATCHING TEXT	20 WORDS

Figure 4.4 Augmented Dickey-Fuller test From Figure 4.4, it can be observed that the p-value is less than 0.05

Figure 5.1: Augmented Dickey-Fuller Test the figure 5.1, it can be noted that the p-value is less than 0.5,

W https://www.diva-portal.org/smash/get/diva2:1366957/FULLTEXT02

18/21	SUBMITTED TEXT	14 WORDS	95% MATCHING TEXT	14 WORDS

Krishnaswamy, "A Comparative Study and Analysis of Time Series Forecasting Techniques", SN

Krishnaswamy, S. A Comparative Study and Analysis of Time Series Forecasting Techniques. SN

W https://link.springer.com/article/10.1007/s42979-020-00180-5

20/21	SUBMITTED TEXT	26 WORDS	81% MATCHING TEXT	26 WORDS
G. Borowik, Z. M. Wawrzyniak and P. Cichosz, "Time series analysis for crime forecasting", 2018 26th International Conference on Systems Engineering (ICSEng), Sydney, NSW, Australia,			G. Borowik, Z. M. Wawrzyniak, and P. Cichosz, "Time series analysis for crime forecasting," in Proceedings of the 2018 26th International Conference on Systems Engineering (ICSEng), pp. 1–10, IEEE, Sydney, Australia, 2018.	

W https://www.hindawi.com/journals/scn/2021/5587511/

21/21	SUBMITTED TEXT	32 WORDS	73% MATCHING TEXT	32 WORDS
S. Marzan, Maria Jeseca C. Baculo, Remedios de Dios Bulos, and Conrado Ruiz, "Time Series Analysis and Crime Pattern Forecasting of City Crime Data", International Conference on Algorithms, Computing and Systems (			S. Marzan, M. J. C. Baculo, R. de Dios Bulos, and C. Ruiz, "Time series analysis and crime pattern forecasting of city crime data," in Proceedings of the International Conference on Algorithms, Computing and Systems,	

W https://www.hindawi.com/journals/scn/2021/5587511/