Crime Rate Analysis and Prediction

Pinakin Nimavat
Illinois Institute of Technology
pnimavat@hawk.iit.edu

Rahul Sharma
Illinois Institute of Technology
rsharma11@hawk.iit.edu

I Abstract

It is important to predict crime in real time. However, predicting when and where the next crime will occur is difficult. There is no known physical model that can accurately approximate such a complex structure. Historical crime data are limited in space and time, and the signal of interest is small. In today's era, nearly 30% of people live in mega-cities. In these areas, criminal activities are much higher and violent. The frequency of crime is determined by a variety of dynamic variables. Crime is rare in comparison to certain predictable occurrences. Crime distributions show significantly different patterns at different spatiotemporal scales. In this work we will use several models to predict and classify the crime rate. Our models are divided into two parts, first is preprocessing of data and visualizing the data. In second part we will build different models for crime prediction.

II Problem Description

In this project we will analyze data provided by the government of Chicago https://data.cityofchicago.org/ to compare different neighbourhood and FBI (Federal Bureau of Investigation) https://crime-data-explorer.fr.cloud.gov/ to compare the same for the city of Chicago, statics of crime rate and future crime prediction. We will perform two tasks: 1) Predicting Crime types 2)Using time-series forecast methods we will try to predict the rate of crimes in the future. We will try to measure and compare accuracy among different machine learning models such as SVM, Random Forest Regressor, Random Forest Classification, RNN-LSTM, CNN, MLP Classifier and Prophet. Univariate Time Series is a dataset comprising a single series of observations with a temporal ordering. To predict the next value in the time series it is required to learn from the series of past observations in temporal ordering.

Expanding Window:
In the field of machine learning generally it is assumed that there is no relation between sequences of data, so the K fold cross validation method works fine. Time series inherit the dependence of sequence or say temporal ordering, so the K fold cross validation does not work properly with time series data. To work around this problem we use expanding windows or rolling windows to train our model with incoming new data. So in a way K fold cross validation in general is equivalent to time series expanding window.

III Data

i. Description of Data

In this work, we consider all types of crime in Chicago for the time period of 2001 to 2017. This dataset has been taken from the Chicago Police Department's CLEAR(Citizen Law Enforcement Analysis and Reporting) system. In total there were 79,41,282 number of rows in the dataset. The dataset contained many 'NA' values. Each crime record includes crime start, end times and location. To avoid ambiguity, we regard start time of each event as the time

slot. Geographically, the latitude and longitude will help differentiating event which have similar starting time. The spatial distribution is highly heterogeneous. A large portion of the area contains only a little crime whereas for some location in smaller area, there are higher number of crimes.

ii. Data Analysis & Preprocessing

We have have removed the null values as the part of pre-processing. Factorized the dataset into numerical and categorical values. We have grouped the data into classes based on the amount(number) of records such the type of crime whose number was less were grouped together. We also wanted to use the dataset as a uni-variate time series forecast that is why we used "Pandas.DataFrame.Resample() which gave us two columns: 1)Date 2)Corresponding number of crimes.

The below figure shows the count of null values analyzed during preprocessing.

```
Unnamed: 0                   0
ID                           0
Case Number                  7
Date                         0
Block                        0
IUCR                         0
Primary Type                 0
Description                  0
Location Description      1990
Arrest                       0
Domestic                     0
Beat                         0
District                    91
Ward                    700224
Community Area          702091
FBI Code                     0
X Coordinate            105573
Y Coordinate            105573
Year                         0
Updated On                   0
Latitude                105573
Longitude               105574
Location                105574
dtype: int64
```

```
Unnamed: 0              0
ID                      0
Case Number             0
Date                    0
Block                   0
IUCR                    0
Primary Type            0
Description             0
Location Description    0
Arrest                  0
Domestic                0
Beat                    0
District                0
Ward                    0
Community Area          0
FBI Code                0
X Coordinate            0
Y Coordinate            0
Year                    0
Updated On              0
Latitude                0
Longitude               0
Location                0
dtype: int64
```

Figure 1: *A) With null B) Without null values*

The following is some of the data visualization graphs to better understand the dataset and to find the trends and main points in the data.

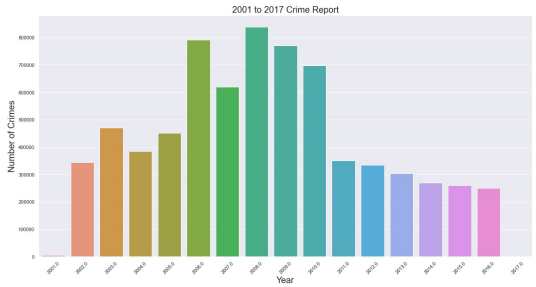


Figure 2: *Crimes trend per year*

Looking at the figure 2 we can see that, Number of crime was higher in 2008.

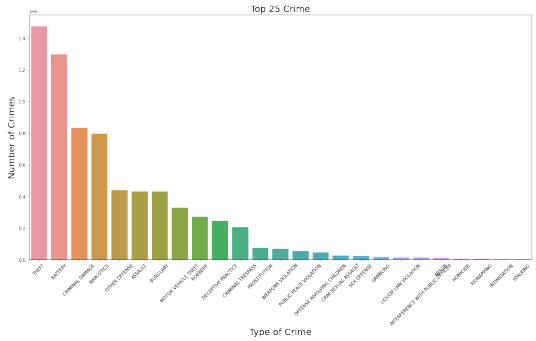


Figure 3: *Top 25 Crimes*

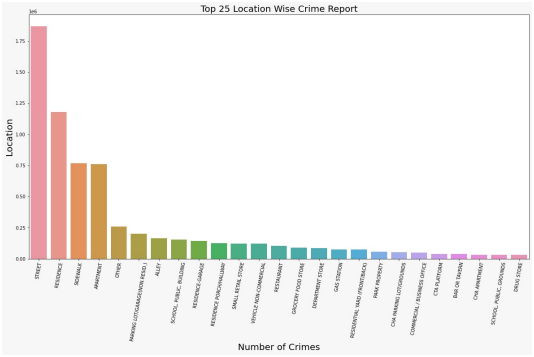


Figure 4: *Top 25 Crimes location wise*

By looking at figure 3 we can see that, higher number of crimes occurred were in the streets and households.

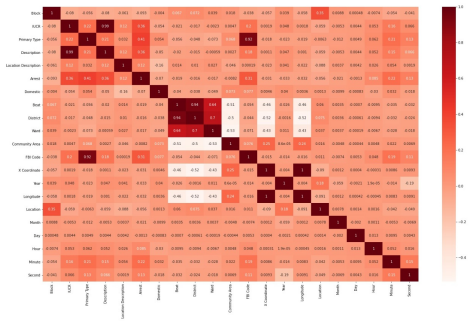


Figure 5: *Heat map for feature selection*

Pearson correlation was used to find the correlation between multiple features. The heat map shows how each features are correlated accordingly. The main features which we selected whose correlation is > 0.4 are: Primary Type, IUCR, FBI Code and Description.

IV Models Used

i. Random Forest Regressor

Random forest regressor is a ensamble machine learning algorithm. Ensemble method is a technique which combines the predictions from multiple machine learning algorithms. Random forest illustrates the power of combining many decision trees into one model. It uses bootstrap aggregation means random sampling with replacement. Here we have used it as single time step model. Here we have taken meta estimator as n estimator = 100.

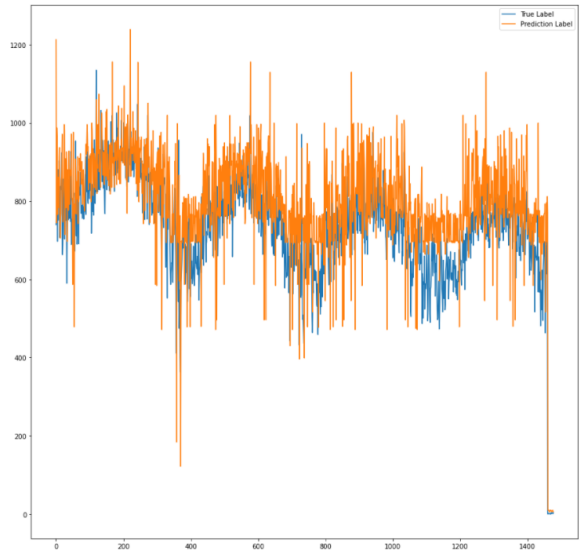


Figure 6: *Accuracy graph*

As we can see from the graph that this model failed to converge all the trend points. We got and R^2 Score of 0.076 and MAE: 101.22. There is a high variance which shows that this method is unreliable while predicting future trends.

ii. Convolutional Neural Network

Cnn takes its name from mathematical linear operation between matrices called convolution. It has multiple layers: convolutional layer, pooling layer, fully connected layer. CNN 1D model can work as a regression model for univariate time series, where number of steps and number of features are input. For Univariate time series the number of features will be 1.

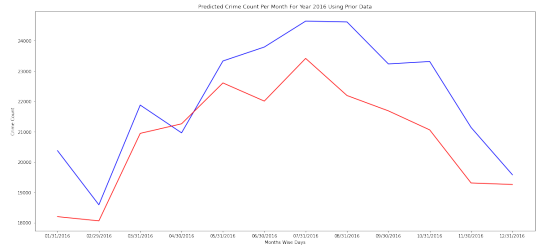


Figure 8: *Cnn*

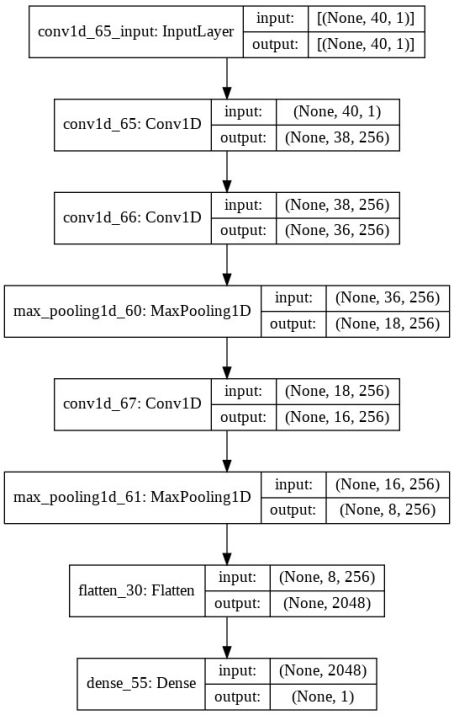


Figure 7: *Model Summary*

Univariate Time series is prepared using 30 timestamps as input data and 1 timestamp as output or prediction data. Each timestamp represents the number of crimes on a particular month.

Results are produced using the last 12 months of predictions with rmse score of 1527.658 and r2 score of 0.362.

iii. Recurrent Neural Network

RNN models with LSTM are well suited to work on time series problems as RNNs are capable of learning features and long term dependencies from sequential and time-series data.

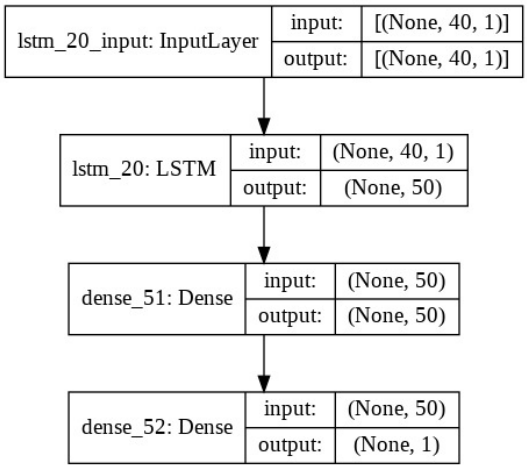


Figure 9: *Model*

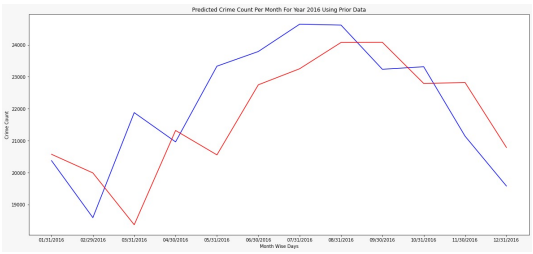


Figure 10: *RNN*

Univariate Time series is prepared using 30 timestamps as input data and 1 timestamp as output or prediction data. Each timestamp represents the number of crimes on a particular month.

Results are produced using the last 12 months of predictions with rmse score of 1554.604 and r2 score of 0.339.

iv. SVM as Regressor

In machine learning, support-vector machines are supervised learning models with associated learning algorithms that analyze data for classification and regression analysis.This algorithm is more flexible than usual machine learning algorithms. The key is the hyper-plane, boundary line, and kernel.
Kernel: The function used to map a lower dimensional data into a higher dimensional data.

Hyper-plane: the line that will will help us predict the continuous value or target value
Boundary line: are two lines other than Hyper Plane which creates a margin . The support vectors can be on the Boundary lines or outside it.
Equation is: $-ey - Wx - b + e$: where e is boundary line.
We have used linear SVR with default hyper parameters. The results of which are presented in the graph below.

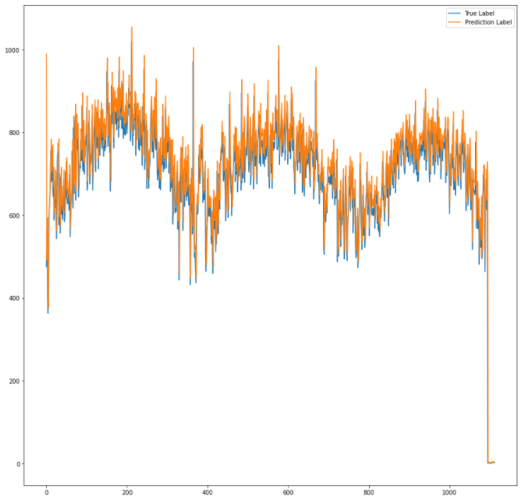


Figure 11: *SVR*

By looking at the graph we can say that SVM has performed quite well. The R^2 score is: 0.64. which shows that up to certain steps our model was generalizing better. Although, being a simple model it worked efficiently. The Mean Absolute Score is: 55.11

v. Prophet

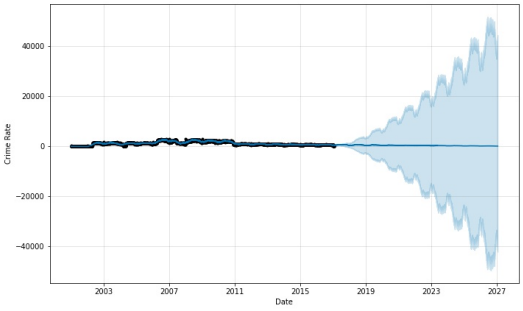
Prophet is a method for forecasting time series data. Here, we have performed the uni-variate time series forecast. Prophet is an additive model Which takes into account non-linear patterns with annual, weekly, and regular seasonality, as well as holiday impacts. Additive model means, each feature is additive but each individual may or may not be a linear feature. It works better with time series that have clear seasonal effects and historical data from several seasons. Prophet is immune to missing data and pattern changes, and it usually manages outliers very well.

$$y(t) = Piecewisetrend(t) + seasonality(t) + holidayeffects(t) + noise$$

Here, Piecewise trend can be a logistic trend or linear trend. To regulate this feature we will use L1 regularization for trend shifts. Seasonality is a trend which repeat itself. To regulate seasonality we perform Fourier series. Holiday effects are the sudden changes in the trend which can be normalized by dummy variables.

It detects trend changes according to the previous changes and forecasts based on it. Interval of these changes is determined by the piecewise function such as output will be 1 when there is shift of trend.When we perform **Fit** it estimates trend changes. When we call **predict** it samples future trend changes from distribution.

- Day wise:



Here we can see in the above graph that it is the prediction for the next 10 years. Here, Black dots are the original data points, dark blue line is the projection and light blue region is trend space.

Here the sudden spikes shows the trend changes. First graph shows the overall trend change. Second one shows days vs week (weeks seasonality) and last graph shows the trend day vs year wise.

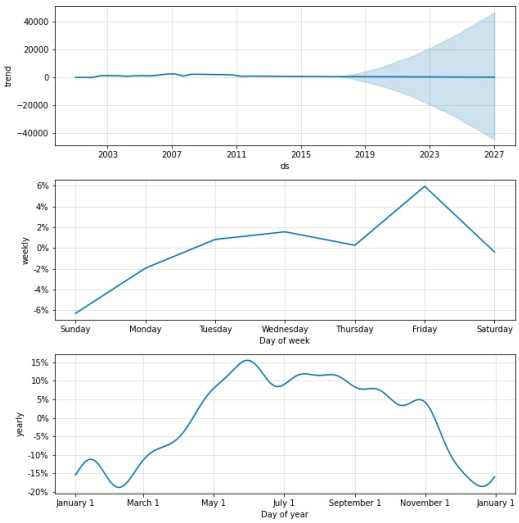


Figure 12: *Prediction for next 10 years, day wise*

- Month wise:

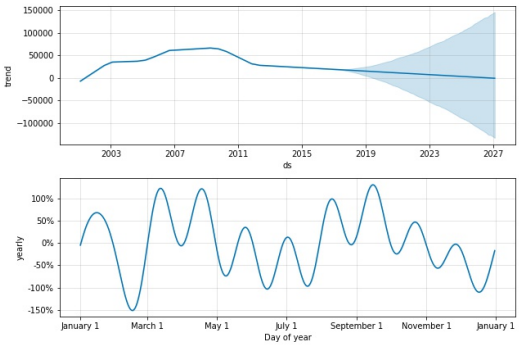


Figure 13: *Prediction for next 10 years, monthwise*

Here this graph content shows the 1) Trend changes derived from monthly data as input 2) Monthly vs Yearly spikes

- Year wise:

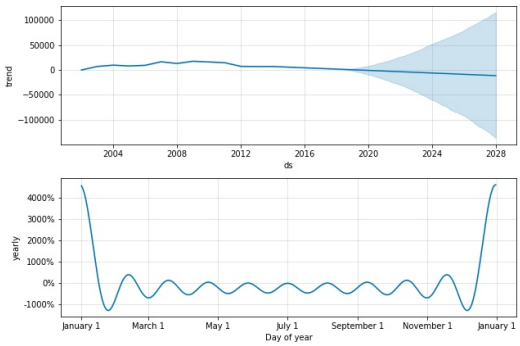


Figure 14: *Prediction for next 10 year, yearwise*

Here this graph content shows the changes which are derived from resampling yearly wise data from the dataset and they are 1) Trend changes 2) Daily vs Yearly spikes

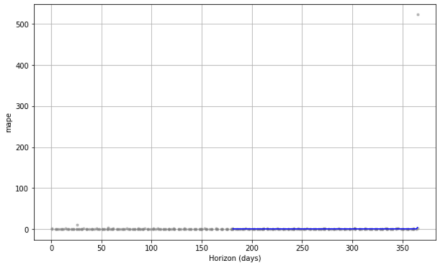


Figure 15: *Shows the accuracy*

	horizon	mape
0	127 days	0.066533
1	132 days	0.338987
2	138 days	0.344199
3	143 days	0.087416
4	148 days	0.070109

Figure 16: *Horizon vs MAPE*

Here, we have taken the 'horizon' as 360 days, 'period' is 180 days and 'initial' as 730

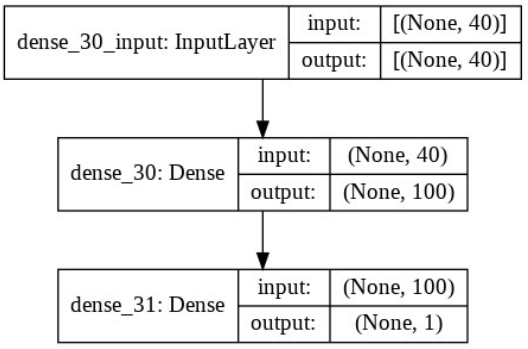


Figure 17: *Model Summary*

days. There were 25 forecast out of which the first 4 entries looked like the above graph.

vi. Multi Layer Perceptron

A typical Artificial Neural Network architecture also known as multilayer perceptron (MLP) contains a series of layers, composed of neurons with their connections. An artificial neuron calculates the weighted sum of its inputs and then applies activation to obtain a signal that will be transferred to the next neuron.

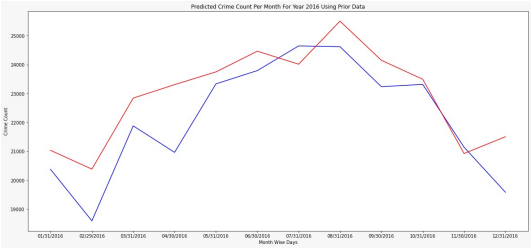


Figure 18: *MLP*

Univariate Time series is prepared using 30 timestamps as input data and 1 timestamp as output or prediction data. Each timestamp represents the number of crimes on a particular month.

Results are produced using the last 12 months of predictions with rmse score of 1064.228 and r2 score of 0.625.

vii. Random Forest Classification

Random forest classification is an ensamble machine learning method. Each individual tree in the random forest spits out a class prediction and the class with the most votes becomes our model's prediction. Trees protect each other from their individual errors (as long as they don't constantly all error in the same direction). While some trees may be wrong, many other trees will be right, so as a group the trees are able to move in the correct direction.Bootstrap Aggregation: Decisions trees are very sensitive to the data they are trained on — small changes to the training set can result in significantly different tree structures. Random forest takes advantage of this by allowing each individual tree to randomly sample from the dataset with replacement, resulting in different trees.In laymen term: for n samples of original data we take a random sample of size N with replacement. Here, we have used Pearson correlation to find the features. Also we have sub sampled the dataset as we have higher number of rows and very less features. Input features are IUCR, FBI Code and Description and target is Primary type as we will predict the type of crimes. Below are some outputs.

```
========== Random Forest Results ==
Accuracy    :  0.84
Recall      :  0.84
Precision   :  0.8089935064935065
F1 Score    :  0.8399999999999999
Confusion Matrix:
[[ 0 13  0  0  0  0  0  0  0]
 [ 0 31  0  0  0  0  0  0  0]
 [ 0  0 21  0  0  0  0  2  0]
 [ 0  0  0 10  0  0  0  5  0]
 [ 0  0  0  0 39  0  0  2  0]
 [ 0  0  0  0  2 11  0  1  0]
 [ 0  0  0  0  0  0 21  2  0]
 [ 0  0  0  1  0  0  2 27  0]
 [ 0  0  0  0  1  0  0  1  8]]
```

Figure 19: *Accuracy*

================ Classification Report ================

	precision	recall	f1-score	support
CRIMINAL DAMAGE	0.79	0.48	0.60	31
BATTERY	0.67	0.82	0.74	39
THEFT	1.00	1.00	1.00	39
BURGLARY	0.53	0.90	0.67	10
ASSAULT	0.00	0.00	0.00	11
MOTOR VEHICLE THEFT	0.73	0.89	0.80	9
OTHER OFFENSE	1.00	0.33	0.50	15
ROBBERY	0.36	0.67	0.47	6
NARCOTICS	0.47	0.83	0.60	18
CRIMINAL TRESPASS	0.00	0.00	0.00	7
OTHERS	0.83	1.00	0.91	15
accuracy			0.71	200
macro avg	0.58	0.63	0.57	200
weighted avg	0.70	0.71	0.67	200

Figure 20: *Classification report*

As we can see, by keeping n estimators = 10 we are able to get 71% accuracy.

V Conclusion

We applied different models with different approaches to gain good accuracy for crime prediction. We compared CNN, RNN, and MLP using a univariate time series with expanding window. MLP performed very well compared to CNN and RNN on our dataset and showed promising results for future crime prediction. Comparison helped us to gain confidence in path to get more accurate prediction in the field of crime predictions. We also used SVM, and Random Forest Regressor as univariate time series model with only one time step. Prophet and SVM gave us good result with R^2 Score of 0.64 (for SVM) compared to random forest regressor which failed badly. As for the classification task, Random forest as a classifier gave us the accuracy of 71% which shows that it was nearly able to predict the right types of predictions for crime types.

References

[1] *The Rise of Technology in Crime Prevention*
Dario Ortega Anderez, Eiman Kanjo, Amna Anwar, Shane Johnson

[2] *Spatial-Temporal-Textual Point Processes with Applications in Crime Linkage Detection*
Senzhang Wang, Jiannong Cao, Philip S. Yu

[3] *Socio-economic, built environment associated with crime: A study of multiple cities*
Marco De Nadai, Yanyan Xu, Emmanuel Letouzé, Marta C. González, and Bruno Lepri

[4] *Deep Learning for Spatio-Temporal Data Mining: A Survey*
Senzhang Wang, Jiannong Cao, Philip S. Yu

[5] *Spatial-Temporal-Textual Point Processes with Applications in Crime Linkage Detection*
Shixiang Zhu, Yao Xie

[6] *Forecasting Crimes Using Autoregressive Models*
Eugenio Cesario, Charles E. Catlett, Domenico Talia

[7] *Crime Prediction and Analysis Using Machine Learning*
Alkesh Bharati1, Dr Sarvanaguru RA.K

[8] *Crime Prediction Using Machine Learning*
Riya Rahul Shah