CRIME HOTSPOT ANALYSIS AND PREDICTION

[1]B. Siva Kanaka Raju,[2]T. Chandana sai,[3]B. Manohar,[4]B. Nagesh

[1]Assistant Professor, [2,3,4]UG scholar

Department of Computer Science and Engineering (AI & ML)

Vignan Institute of Technology and Science, Nalgonda, Telangana-508284

Email:[1]bathula.kanna@gmail.com
[2]chandanasai1672004@gmail.com,[3]manoharbavandla123@gmail.com,[4]chintuboya002@gmail.com

ABSTRACT

The rise of crime in many cities have negative impact on public safety and welfare. To address this issue, the current study is aimed at forecasting crime incidence in selected Indian cities using a machine learning model integrated into the easy Flask web application. With the help of this application, users can determine the city, type of crime, and year to obtain predictions about the crime incidence for such selection.

The existing system showed the importance of spatiotemporal data in predicting crime trends on location and time. Despite promising advancements, the paper outlines several challenges in crime prediction research including data quality issues, model interpretability and risk bias in prediction. It directs the future research like improving algorithm transparency.

To overcome the issues and improve the prediction rate of crimes, we tested five different algorithms and compared the performance of each algorithm. To make the model more accurate, the model is supposed to make 1% annual population growth, this is included into the adjustments to be made in every particular subsequent year. For crime prediction purposes, areas are placed into several categories which include basic and raised crime rate areas while ratios are used to provide estimations of how many crime cases will be recorded in a particular year.

Using this project, authorities and citizens would be able to visualize crime risks in the future in simple way. The prediction model can help in predicting the crime actions to prevent crime or at least in informing the population.

Finally, with this system, crime pattern analysis assists in technology applications with better decision-making and improved security.

Keywords- *Crime Rate Prediction, Machine Learning, Flask Web Application, Population Growth Adjustment, Risk Categorization.*

I. INTRODUCTION

Increase in crime rates is a big problem for societies all over the world. This is a serious concern where we should look forward because knowing about crime help in planning and taking actions. If we can predict when and where the crime rate is high or low we can enhance the law and can be prepared better. We can focus these resources and efforts in right areas, and take steps to prevent crimes before they happen. With this modern technology, predicting crimes has become easier, which is it is possible to stop crimes or reduce them more effectively.

Keeping this in view the project aims to design a machine learning model that builds upon historical data and attempts to predict crime rates. The model will rely upon a few relevant inputs such as Area or place, period when offence occured, and past data to input into the prediction model. Our model will output the predictions which will help in crime reduction and improve the safety of an area. A Flask Web application act as an interface that provide interaction to user with the crime rate prediction model.

The web interface come with a set of inputs which consist of City name, City type and the year in which the user wants to know the crime rate. After the user enter all the inputs an

output is displayed showing the relevant information of number of crimes happened in the year, if it is past or current year else it gives the estimated crime and estimated number of cases for the future year.

This simple and user-friendly web interface allows the user to interact with the web page easily and it can be used very simply. The algorithm used in the web interface is chosen among few algorithms based on the accuracy it is giving. The more accurate algorithm (Random Forest Regressor) is stored as model.pkl format. We initially train and test the model which we called Analysis and then predicting the crimes by choosing the accurate model is called prediction.

In this proposed system, the main aim is to predict offences based on previous data effectively and create a smooth web application. The project will also include a user manual to help people understand how to use the application.

II. RELATED WORK

This study indicates Crime prediction is a multicomplex problem that calls for decent predictive tools to avoid loopholes in the detection mechanism. Usage of machine learning models helped in predicting the crime accurately.[1]

The research focuses on analysing crime patterns with models using computer vision where this approach benefits the people by allowing them to know the areas with offences and can stay safe priorly.[2]

The paper introduced spatio temporal data for predicting crime rate, which is multidimensional data which is collected across space and time. The vast data is used in analysis as well as prediction that help in the classification of different crimes.[3]

This study evaluates previous machine learning techniques for detecting and classifying multiple types of crimes using text analytics. It identifies gaps and recommending improvements in detection accuracy and stability for crimes or offences. [4]

This study develops a gradient boosting technique that had performed better than the other three models under comparison by predicting criminality in urban areas, being the best model to use in predicting crime rates. [5]

It focuses on predicting crime using the XG Boost algorithm which is a machine learning model that helped in in crime prediction. it highlights advancements in modelling and predicting crimes under the crime attacks focusing on challenges and future directions. By acquiring some techniques [6]

This research defines a multimodule approach and the fact is exempt on two grounds: the weak data abstraction of the modules and the prediction of crimes on the datasets of Chicago and New York Crimes. [7]

In the given paper based on the data like spatiotemporal data, the authors developed a regression model using convolutional neural networks and regression can itself give superior accuracy and real-time performance when combined with the spatiotemporal neural network. [8]

This paper relies on the real time application where detecting the objects is done which help the official authorities to do their work more faster. By the direct scene prediction, the investigators are benefited the most. [9]

The work provides comprehensive explanation of cross validation as a statistical model for evaluating and optimizing machine learning models. They can then leverage this information to better explore the crime, which will help them devise proactive measures in preventing criminal activities. [10]

III. METHODOLOGY

Historical crime data from openly accessible sources is collected. These include government databases, police department records, or open data portals. This data includes type of offences occurred, action where it happened, and other attributes such as whether the crime was violent or non-violent, additionally, outside data sources are also considered, like population density, economic factors, and weather conditions during the time of the crime. After the data has been collected, pre-processing is done to clean the data. It includes spotting missing values, removing inconsistencies or duplicates, and formatting raw data for usability. For example, if we take categorical data, such as crime type, it needs some kind of encoding into numerical values while temporal attributes would require more usable representations such as day of the week or seasonal data. In this step, the irrelevant data is removed to avoid skewing the expected classifications by the model.

Now to explore patterns Exploratory data analysis helps in identifying the presence of patterns and relationships within the data. The second step includes the data preprocessing where it involves the usage of graphical and other forms of presentation to indicate trends and duplications based on the contexts, types, and locations of the crimes occurring during a particular time period, such as whether or not certain types of crime are more concentrated during specific times of the day or certain districts or areas. Next when we move to Feature engineering, it helps in constructing of features (variables) that will help in predicting the crime rate in a better way. This includes when the crime happened and whether it is a crime against elder or robbery or other and also other factors such as which month of the year it is or what weather conditions exist can be equally important. The selection of factors helps in improving the predictive model accuracy and also Exploratory Data Analysis. The project aims to integrate with all the relevant features that is used to feed the machine learning model to create and improve the predictions accuracy.

We build the machine learning model now, after the above steps feature engineering and data cleaning have been completed. The data is split into two parts: one is used to define the training set, followed by the training of the model, and the second, used to test for performance only. Crime rate prediction methods include five different techniques: Support Vector Machines, Decision Trees, Random Forests, and Linear Regression. In order to minimize errors and learn from the data to adjust their internal parameters during the training phase, the models had to understand the pattern and relationships in a given data set. The performance of these models is assessed throughout the training phase by a number of metrics, where we tried to offer a comparison and between the models underneath the various respective metrics.

The trained models are assessed against one or a combination of variants of the metrics like Mean Absolute Error, Mean Squared Error, and R-squared error, where these metrics are used to check the performance of above mentioned models and this help us in selecting a better model for prediction. If there is poor performance by the model, hyperparameter tuning is performed to guide the model. Hyperparameters are those configurations that guide the process of learning by the model, and tuning of a hyperparameter can yield better results, else we used some cross-validation techniques so that the model can give accurate results to new, or unseen data. After all these steps we verified that the model had passed all the tests with performance required and it is stored in a pkl format, and integrated with flask.

The below given flowchart in Figure 1 tells the workflow of the project where each step is followed and starting with Data Collection where data is collected through multiple resources, Data preprocessing, Feature Selection and target variables where relevant features are chosen, Data Splitting, Model Training where the chose model is trained, Model Evaluation where the model is evaluated, Model Saving, Web Application Integration where the chosen model is stored in pickle file in model.pkl format and it is used for web application integration and Result Interpretation where results are interpreted where users can give input and the output is generated, if it is future year, the estimated output is shown else if it is current year it shows the given values of ongoing crimes.

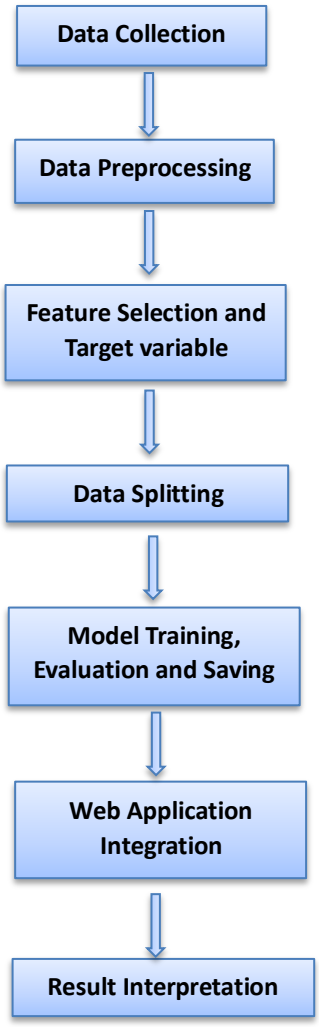

Figure 1: Working Flowchart

Now the chosen model became integrated into a Flask Web application to allow for the model deployment. The interface included inside this Flask Web application allows user to interact with the Model. Here, the users should enter the relevant inputs such as location (the city or place where the user want to know the crime happened), year when the user want to know in which year the crime happened, and crime type to obtain the predicted crime rates from particular areas. Through the Flask app, the model collects the inputs given by the user so that in real time the model can predict what the crime rate would be. The web interface is designed for simple operation, with easy guidance for data entry.

To check for the accurate algorithm, we tested the Mean Absolute Error, Mean Squared Error, R Squared Score (Error) for few Machine Learning Algorithms like Support Vector Machines, K-Nearest neighbors Regressor, Decision Tree Regressor, Random Forest Regressor, and Multi-Layer Perceptron Regressor. From these testing results the highest accuracy was for Random Forest Regressor. So, this model was saved as model.pkl and used in Flask application.

Algorithm	Mean Absolute Error	Mean Squared Error	R^2 Error
Support Vector Regressor	10.3205	371.7907	-0.1789
K-Nearest Neighbors Regressor	6.8466	150.4450	0.5230
Decision Tree Regressor	2.8871	34.8720	0.8894
Random Forest Regressor	2.4880	21.3630	0.9323
Multi-Layer Perceptron Regressor	12.4248	307.5506	0.0248

Table 1: Comparison of Performances of Algorithms

Mean Absolute Error help us in identifying difference between the outcomes generated by the model that we used for prediction and original values. A smaller Mean Absolute Error means the model is doing a better job of predicting. Mean Squared Error is similar to Mean Absolute Error, but this will square the errors before averaging them. Squaring makes larger errors even bigger, so if mistakes are more the error is more. Like Mean Absolute Error, a smaller Mean Squared Error means the model is more accurate. R Squared Score help in measuring how well the model's predicted values are matching with the actual data. If R^2 is 1, it means the model is perfect and if it is 0, it indicates the model is not accurate to predict values and if it is negative, it indicates model outputs are worse than the values that are generated randomly. The Random Forest Regressor had the smallest errors (MAE: 2.48, MSE: 21.36) and the highest R^2 score (0.93). This tells us that Random Forest Regressor is making very few mistakes and performing better when compared to other algorithms.

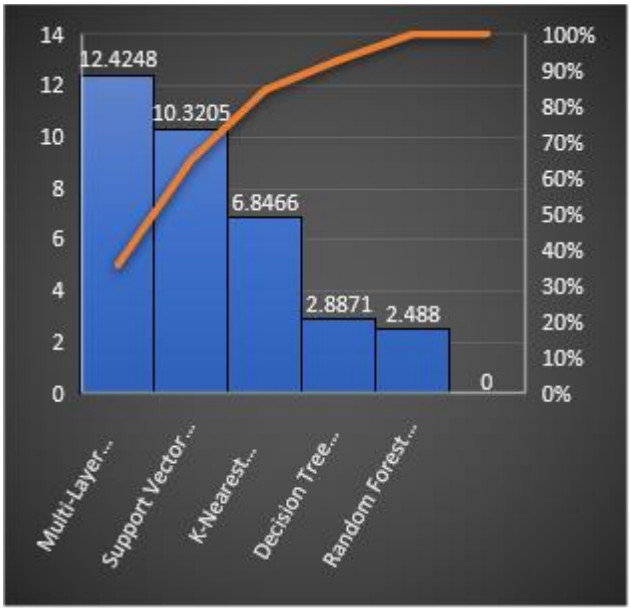

Figure 2: Graphical representation of Performance of Machine Learning Models.

In Figure 2 the bar graph shows the error rate of above mentioned models. The blue bar represents error values, orange line represents the cumulative performance trend. From the chart, we can see that MultiLayer Perceptron has highest error and Random Forest has lowest and performed best. In conclusion, the application security testing was also conducted to know the functionality of the application's performance. The red line shows the error percentage of the model, since the error percentage of Random Forest Regressor is between 10 to 20 percent, we are using this algorithm for Crime rate prediction.

Algorithm1: Random Forest Regressor

Start

Step 1: Preprocess the Dataset

1.1 Load the dataset, and handle values.

1.2 Use Label Encoding to convert categorical features (City, Crime Type) into numerical values.

1.4 Compute the Crime Rate:

$$\text{Crime rate} = \frac{\text{Number of Cases}}{\text{Population}}$$

1.5 Normalize population data by adjusting it with a 1% annual growth rate for predictions beyond the base year (2011).

Step 2: Split the Dataset

2.1 Divide the dataset into training and testing.

2.2 Define features (X) as Year, City, Population, and Crime Type, and y as Crime Rate.

Step 3: Train the Random Forest Regressor

3.1 Initialize the Random Forest Regressor with hyperparameters:

- Number of Trees: n_estimators(ex:100).
- Maximum Depth and Random Seed.

3.2 Train the model using the training data (X_train, y_train).

Step 4: Evaluate the Model

4.1 Predict the Crime Rate on the test data (X_test).

4.2 Calculate the performance metrics: Mean Absolute Error, Mean Squared Error, R^2 Score

Step 5: Save the model in .pkl file using pikle.

Step 6: Integrate with Flask Application

6.1 Load the saved model in the Flask web app.

6.2 Take user input: City, Year and Crime Type.

6.3 Adjust the population dynamically using the 1% annual growth formula:

$$\text{Population (Adjusted)} = \text{Population (2011)} \times (1+0.01)^{\text{Year}-2011}$$

6.4 Use Random Forest model to predict Crime Rate.

Step 7: Output the Results

7.1 Classify the prediction into categories:

- **Very Low Crime Area** (Crime Rate ≤ 1)
- **Low Crime Area** (Crime Rate ≤ 5)
- **High Crime Area** (Crime Rate ≤ 15)
- **Very High Crime Area** (Crime Rate > 15).

7.2 Calculate the estimated number of cases:

$$\text{Number of Cases} = \text{Crime Rate} \times \text{Population (Adjusted)}$$

7.3 Display results on the web app:

- Predicted Crime Rate
- Risk Category
- Estimated Number of Cases

End

IV. RESULT AND DISCUSSION

After training and testing the model, accuracy and working of system is checked whether the predictions are correct or not. Random Forest Regressor was chosen as it performed better than the other models due to its property in handling larger datasets and knowing complex patterns for predicting the crime rate. The model predicted the crime rate with less error, as demonstrated by a lesser Mean Absolute Error and Mean Squared Error.

A user-friendly interface is created with basic functionalities of the model, we developed the web application by using the Flask application and made some adjustments. The user has to type in details like location, time, and other offenses for prediction purposes. These will be presented in a method in a way that makes it easier for the users to understand certain crime trends, and show in graphical form as shown in Figure 3. After testing of the Web application, it showed good responses, generating predictions almost accurately with only minor errors.

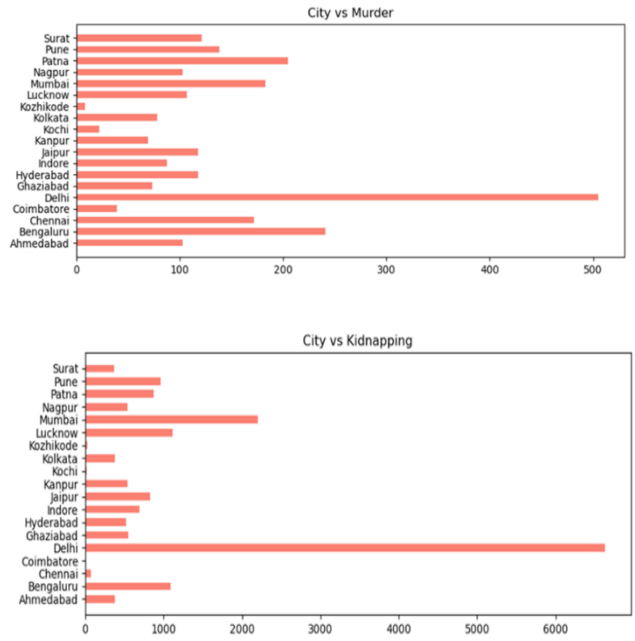

Figure 3: Comparison between the cities and Illegal activities.

The above graphs play vital role to understand and deeply analyze the dataset. Bar graphs were used to visualize the relationship between different cities and various crime types. These graphs provide a clear representation of the distribution of crimes across cities, making the user to

identify patterns easily. For example, in above Figure 3 the "City vs. Crime Type" graphs highlight which cities have higher crime rates for specific categories, helping us in understanding the variations in crime activities depending upon the different places.

Additionally, these visualizations in graph helped us in exploring how crime data varies over time and across different population sizes. By examining these graphs, we gained few knowledge on the overall structure of the dataset, which supports the feature engineering and modeling processes. This type of exploratory data analysis ensures that important patterns are captured and incorporated into the machine learning model for better predictions. The graphs also make the findings even more clear and accessible for users, enhancing the accuracy of the results.

This model for predicting crime rates has proved quite useful for revealing various trends and patterns in different cities to predict different crimes. In conclusion, comparing different models and knowing the accurate model helped us in predicting the crime rate efficiently and creating a user-friendly interface.

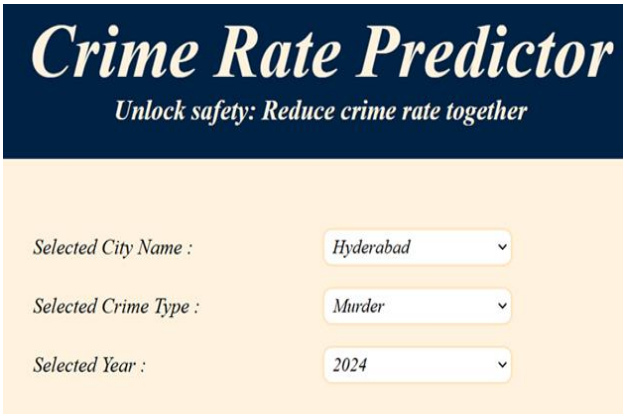

Figure 4: Web Interface of the system

The above web application is a user friendly and interactive page that help the user to predict crime rate based on the criteria like city name, city type and selected year. The model allows to analyse and predict the future crime incidents thereby showing the future scope. When users submit their inputs, the web page communicates with the trained machine learning model to predict the crime rate for the selected parameters. It displays the results in an organized manner, providing insights such as the predicted crime rate, expected number of cases, and whether the area is classified as a low, medium, or high crime area. This real time feedback enables users to make decisions based on predictions. This webpage is developed to be scalable and interactive, making it easy to add future features such as graphical visualizations, comparison tools, or even safety recommendations. The web interface helps to fill the gap between the technical backend and the end user, making the crime prediction system accessible practical and easy for everyday use.

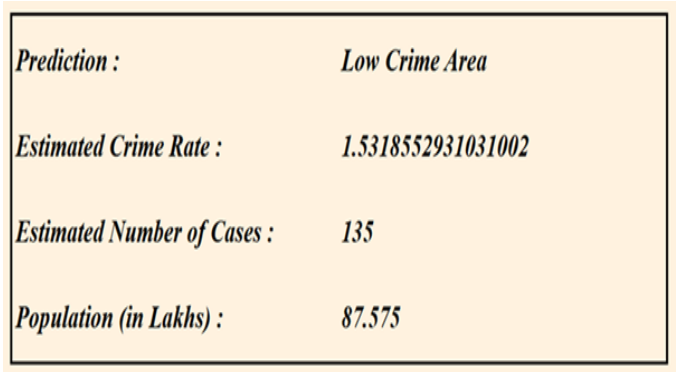

Figure 5: Final Predictions of the User Inputs

The result page gives the estimated crime rate, estimated number of cases and the population based on the input given by the user. Thus, the project help in Analysing and predicting the future crimes and help people, other sophisticated departments in analysing crimes.

V. CONCLUSION AND FUTURE SCOPE

This project on crime prediction showcases how machine learning is applied in predicting trends in crime based on historical data. This made use of an application of machine learning utilizing data collection and processing including crime incidents, the training of an appropriate predictive model like random Forest Regressor, and its deployment in an easy, user friendly Flask web application that provided valuable insight into crime patterns. Thus, it divided areas and durations quickly becoming easy to predict and estimate.

In the future, the model will rely on real time data sources, like weather, web community, and live event information, to accommodate sharp changes in the crime pattern. Expansion of the data sources would help build up a clear view of crime patterns along some aspects of social and indications, particularly focused on underrepresented regions. The creation of a mobile application to assist the community with access to crime predictions anywhere or anytime would also

represent a commendable upgrade, assisting in making the system a potent tool for law enforcement and public safety planning.

VII. REFERENCES

[1] N. Shah, N. Bhagat, and M. Shah, ‘‘Crime forecasting: A machine learning and computer vision approach to crime prediction and prevention,’’ Vis. Comput. Ind., Biomed., Art, vol. 4, no. 1, pp. 1–14, Apr. 2021.

[2] P. William, A. Shrivastava, N. S. Karpagam, T. Mohanaprakash, K. Tongkachok, and K. Kumar, ‘‘Crime analysis using computer vision approach with machine learning,’’ in Mobile Radio Communications and 5G Networks. Berlin, Germany: Springer, 2023, pp. 297–315

[3] S. Hossain, A. Abtahee, I. Kashem, M. M. Hoque, and I. H. Sarker, ‘‘Crime prediction using spatio-temporal data,’’ in Computing Science, Communication and Security. Gujarat, India: Springer, 2020, pp. 277–289.

[4] M. Saraiva, I. Matijosaitiene, S. Mishra, and A. Amante, ‘‘Crime prediction and monitoring in Porto, Portugal, using machine learning, spatial
and text analytics,’’ ISPRS Int. J. Geo-Inf., vol. 11, no. 7, p. 400, Jul. 2022.

[5] A. R. C. da Silva, I. C. D. P. Junior, T. L. C. da Silva, J. A. F. de Macedo, and W. C. P. Silva, ‘‘Prediction of crime location in a Brazilian city using regression techniques,’’ in Proc. IEEE 32nd Int. Conf. Tools Artif. Intell. (ICTAI), Nov. 2020, pp. 331–336.

[6] X. Zhang, L. Liu, M. Lan, G. Song, L. Xiao, and J. Chen, ‘‘Interpretable machine learning models for crime prediction,’’ Comput., Environ. Urban Syst., vol. 94, Jun. 2022, Art. no. 101789.

[7] N. Tasnim, I. T. Imam, and M. M. A. Hashem, ‘‘A novel multi-module approach to predict crime based on multivariate spatio-temporal data using attention and sequential fusion model,’’ IEEE Access, vol. 10, pp. 48009–48030, 2022.

[8] X. Zhou, X. Wang, G. Brown, C. Wang, and P. Chin, ‘‘Mixed spatiotemporal neural networks on real-time prediction of crimes,’’ in Proc. 20th IEEE Int. Conf. Mach. Learn. Appl. (ICMLA), Dec. 2021, pp. 1749–1754.

[9] M. Nakib, R. T. Khan, Md. S. Hasan, and J. Uddin, ‘‘Crime scene prediction by detecting threatening objects using convolutional neural network,’’ in Proc. Int. Conf. Comput., Commun., Chem., Mater. Electron. Eng., Feb. 2018, pp. 1–4.

[10] K. B. Sahay, B. Balachander, B. Jagadeesh, G. A. Kumar, R. Kumar, and L. R. Parvathy, ‘‘A real time crime scene intelligent video surveillance systems in violence detection framework using deep learning techniques,’’ Comput. Electr. Eng., vol. 103, Oct. 2022, Art. no. 108319.