# Navigating Urban Safety with Data: An Improved Machine Learning Perspective

Jerry Liu
*Computer Science Department*
*San Jose State University*
San Jose, CA
jerry.liu@sjsu.edu

Jatin Unecha
*Computer Science Department*
*San Jose State University*
San Jose, CA
jatin.unecha@sjsu.edu

***Abstract*—With the increase in crimes, pedestrian and public safety has become an important concern. This paper presents a web-based application which can work like an early warning system by informing pedestrians about potential crimes at their location. This application uses San Francisco city's police calls data and predicts crimes based on spatial as well as temporal features. Random Forest Classifier and other classification machine learning models are used for predicting the crime which is most likely to occur based on the current location and time of day.**

***Index Terms*—Big Data, Machine Learning, Random Forest Classifier, Crime Prediction**

## I. Introduction

### A. Background

Crime constitutes an act or omission that violates established legal norms that involves behavior that is deemed harmful or threatening to individuals, communities, or society at large. Criminal activities range from minor offenses, such as petty or vehicle theft, to major crimes, like assault, robbery, or kidnappings. In the context of a metropolis city like San Francisco, California, crime is influenced by several factors, including population density, income disparity, and resource availability.

Crime prediction has gained traction with municipalities and law enforcement to develop predictive models. The advent of big data and machine learning techniques has provided avenues for developing intrinsic solutions to determine crime hotspots and predict crime patterns. Traditional methods employed for crime hotspots and predicting crime have utilized kernel density estimation, machine learning, and deep learning algorithms like Naïve Bayes, Decision Tree Classifier, SVM, and LSTM neural networks.

The motivation for this study lies in the design of an early warning system method to enhance public safety, analyzing specifically for a traveler and public safety and exploring methodologies for crime prediction and handling large data volumes. This approach can integrate real-time data to identify patterns and anomalies indicating criminal activity.

This study presents various methods, such as Logistic Regression, Gradient Boosting Classifier, Random Forest Classifier, and Long Short Term Memory neural network. It also corroborates previous findings and enhances machine learning algorithms by incorporating the impact of temporal features.

### B. Report Organization

The rest of the report is organized as follows: Section II presents a summary of current research in the related topic and an understanding of what methods and techniques have been used to predict crime. Section III describes the dataset utilized and the contents within each dataset. Section IV further describes the data preprocessing measures to prepare the data prediction. Section V explains the machine learning and deep neural networks used to create predictive models. Section VI presents prediction results, accompanied by the classification matrix, confusion matrix, and accuracy specifics. Section VII provides a comprehensive summary of the application constructed to display the predicted results for a given location and crime hotspots. The report concludes with Section VIII, which discusses conclusions and future work.

## II. Related Works

Wajiha Safat *et al* [1] aimed at enhancing crime prediction and forecasting by employing various machine learning algorithms and time series analysis techniques. The study focused on analyzing crime data from two major cities, Chicago and Los Angeles, using logistic regression, support vector machine (SVM), Naïve Bayes, k-nearest neighbors (KNN), decision tree, multilayer perceptron (MLP), random forest, and eXtreme Gradient Boosting (XGBoost) algorithms. In addition to the machine learning algorithms, the study employed time series analysis techniques, specifically Long-Short Term Memory (LSTM) and autoregressive integrated moving average (ARIMA) models focusing on temporal patterns and making accurate predictions for future crime trends. The performance evaluation of the techniques was conducted using metrics such as root mean square error (RMSE) and mean absolute error (MAE), ensuring the reliability of the predictive models. The study uses exploratory data analysis to predict over 35 crime types and reveals interesting insights about crime patterns in Los Angeles and Chicago. The LSTM model effectively captured complex temporal patterns in crime data, providing valuable insights for the early identification of crime, identification of hotspots with higher crime rates, and

accurate forecasting of future trends. The findings highlight the improved predictive accuracy of the LSTM model and provide early identification of crime, hotspots with higher crime rates, and future trends.

Araujo *et al* [2] present a web-based application ROTA-Analytics, which outputs crime incidence forecasting and helps patrol supervisors dispatch patrol units based on crime predictions. The work suggests that machine learning can help to detect changes in crime trends and help the police department to adapt to them. The authors divided the city into subregions based on crime distribution and generated a time series for each subregion based on time granularity. They used forecasting methods, such as AutoRegressive Integrated Moving Average (ARIMA) and MultiLayer Perceptron (MLP). Besides using Mean Squared Error (MSE), the authors created new metrics named Regression Accuracy Score (RAS) and Regression Precision Score (RPS) to evaluate the models. This work used ten years (2006-2016) of crime records in Natal city. Adopting the new ROTA modules that the authors added produced strategic and operational results and brought innovative and better planning of police resource distribution in the city.

Shuyu Yao *et al* [3] addresses crime prediction and hotspots using a random forest classifier. With the utilization of historical data might no longer sufficient, the study explores the notion of non-historical covariate data to be used in the model to improve and update the accuracy of the prediction. The study adopts the random forest algorithm as the main predictive model. Initially, the study divides the study areas into four categories based on the distribution of crime hotspots in the historical crime data: frequent hot areas, common hot areas, occasional hot areas, and non-hot areas, and then integrates representative covariates from the non-historical crime data. Real data is analyzed, and the experimental results demonstrate the effectiveness of integrating covariates into the prediction model. The accuracy of the crime prediction model with covariates is improved compared to the model that solely relies on historical crime data.

Elluri *et al*. [4] employed statistical analysis methods and machine learning models to predict crimes in New York City. The authors examined the effects of temporal and weather-related factors on crime frequency. After extracting and cleansing the dataset, they used feature selection techniques such as forward and backward stepwise selection. The authors then use multiple classification methods for prediction, using Keras and TensorFlow libraries for deep learning models such as CNN, RNN, and LSTM. They compared the performance with models like Multilayer Perceptron, Decision Trees, Logistic Regression, Random Forest, and SVM. The results demonstrated that Decision Trees, Multilayer Perceptron, neural networks, and Logistic Regression performed well, achieving the highest AUC of 1 and the lowest RMSE of 0.035. Interestingly, the authors observed that the impact of weather-related attributes on the predictions was negligible, despite their apparent relevance based on the feature selection techniques employed.

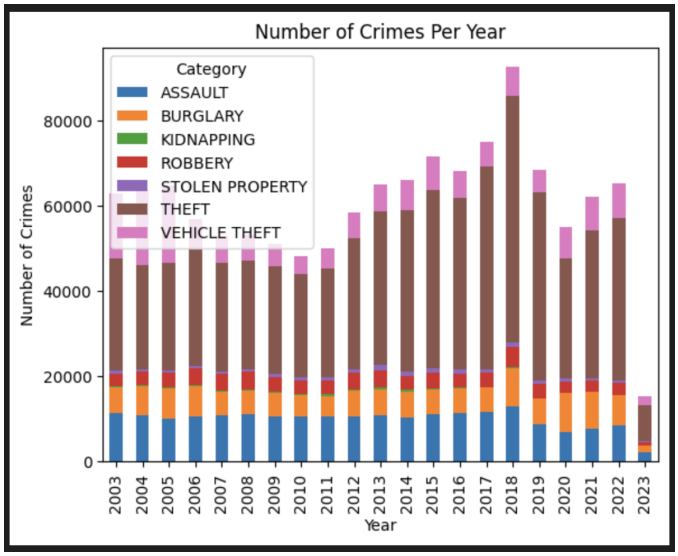



Fig. 1: Chart showing crime over the years

## III. Dataset Description

Accurate predictions of crime trends rely on comprehensive and extensive datasets. Given the dynamic nature of trends, up-to-date data is needed to ensure the relevance of the model. In this paper, we utilize two police service calls datasets provided by San Francisco Police in California. The two datasets capture both historical crime patterns and current trends. These two datasets encompass both historical crime patterns and current trends. Figure 1 shows the historical crime occurrences each year from 2003 to the present.

*a) Dataset 1: 2003 to 2018:* The first dataset spans between 2003 to 2018, containing over 2.18 million records of police service calls. Each of the datasets has up to 14 columns, including the date and time of the incident, the incident category, a description of the incident, and the longitude and latitude of the crime.

*b) Dataset 2: 2018 to Present:* In 2018, San Francisco City updated how police service calls records were recorded, gathering more data regarding each call. The update gives a more diverse dataset, providing up to 27 columns. These columns provide detailed information, such as the date and time of the incident (DateTime), the category and description of the event, the specific neighborhood where the crime occurred, whether the incident was filed online, and the resolution of the incident.

## IV. Dataset Preprocessing

### A. Feature Selection

In examining the two datasets, a crucial step was identifying shared attributes to combine the two datasets. We identified three common features present in both datasets: crime category, spatial information (latitude and longitude), and temporal information (date and time of the incident).

### B. Data Cleaning

As our model primarily focuses on public and traveler safety, we specifically targeted a set of crimes within both datasets. The categories we focused on include assault, theft, robbery, stolen property, vehicle theft, burglary, and kidnapping.

*1) Dataset 1: 2003 to 2018:* For dataset 1, we initially extracted only the relevant columns, namely "Category" for crime classification, "Date" and "Time" for temporal data, and "X" and "Y" for spatial data. Subsequently, we filtered out rows that did not correspond to the selected crime categories and removed any rows with missing values (NaN).

*2) Dataset 2: 2018 to Present:* In dataset 2, we began by identifying the different types of incident categories before determining which rows to retain. We observed under certain crimes, there were specific subsets, such as three different subcategories for "Human Trafficking". In the context of our dataset, we ultimately merged these subcategories with kidnapping. After generalizing some call types, we eliminated rows that did not align with our specific crime categories and removed any rows with missing values (NaN).

### C. Visualization

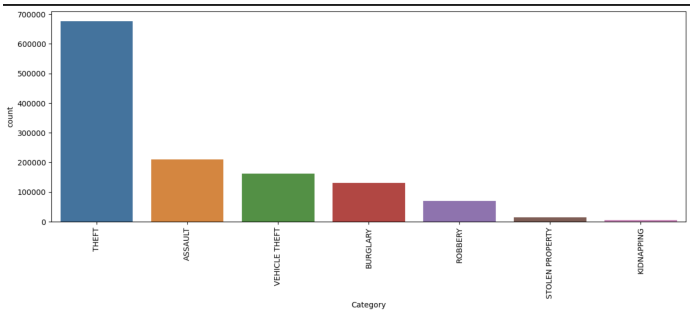
To provide an overview of the crime statistics, we created a bar chart illustrating the occurrence of crimes each year, as well as the overall distribution of crime types. Upon closer examination of the charts, we observed the following reported crime figures in San Francisco from 2003 to the present: 676,130 reported thefts, 210,417 reported assaults, 162,450 reported vehicle thefts, 130,803 reported burglaries, 70,308 reported robberies, 14,977 reported cases of stolen property, and 4,434 reported kidnappings.

### D. Dataset Balancing

Figure 2 shows the imbalance in our dataset. We can see that while the top crime category- "Assault" has 210410 occurrences, the least occurring crime, "Stolen Property" has only 14975 occurrences. This imbalance in the dataset can make our model biased and predict the majority classes more often than the minority classes. We used the Synthetic Minority Over-sampling Technique (SMOTE) to balance our dataset. SMOTE creates synthetic samples of the minority classes to balance the class distribution and provide the model with more representative and diverse training data.

### E. Features Scaling

For feature scaling, we employed StandardScaler from the sklearn preprocessing toolkit. This rescales each feature's values with a mean of 0 and a standard deviation of 1. By doing so, we prevent the regression model from being biased towards features with larger or smaller values, promoting fair treatment of different features. The categorical feature IsHoliday was appropriately encoded before scaling to incorporate it into the classification model.




(a) Chart showing Imbalance

| | |
|---|---|
| THEFT | 676130 |
| ASSAULT | 210417 |
| VEHICLE THEFT | 162450 |
| BURGLARY | 130803 |
| ROBBERY | 70308 |
| STOLEN PROPERTY | 14977 |
| KIDNAPPING | 4434 |

(b) Occurrence counts for each crime

Fig. 2: Imbalance in the Dataset

### F. Prepping the data

We used One-Hot Encoding to encode the values in ['Day_of_Week', 'Part_of_Day'], and Label Encoding to encode the values of the target column ['Category'].

## V. Prediction Models

### A. Logistic Regression

Logistic Regression is a popular classification algorithm and linear model that predicts the probability of an instance belonging to a particular class. By applying a logistic function (sigmoid) to a linear combination of the input features, it maps the output to a probability value between 0 and 1. A threshold is then applied to make the final class prediction. Logistic Regression is often used as a baseline model and building block in more complex algorithms.

Logistic Regression assumes that the relationship between the features and the target is linear. In this case, it performs poorly because of the non-linear relationships between the features and the target variable.

### B. Gradient Boosting Classifier

Gradient Boosting Classifier is an ensemble learning algorithm that combines multiple weak learners or classifiers to create a strong classification model. This is done by sequentially adding models that correct the mistakes of the previous model. In each iteration, the algorithm calculates the gradient of a loss function, such as deviance, to determine the direction in which update the model should be updated. By continuously refining the ensemble, the Gradient Boosting

Classifier gradually improves its predictive performance and effectively handles complex patterns in the data.

In this problem, although Gradient Boosting Classifier performed better than Logistic Regression, the performance is still bad, and we think that this is because this model tends to suffer from overfitting and needs proper hyperparameter tuning, which we could not perform due to limited memory resources.

### C. LSTM Neural Network

The LSTM (Long Short-Term Memory) neural network is a type of recurrent neural network (RNN) architecture that is particularly effective for sequence-based classification tasks. It addresses the limitation of traditional RNNs by incorporating memory cells, allowing it to capture long-range dependencies and handle vanishing or exploding gradients. The LSTM network uses a complex gating mechanism to control the flow of information, including input, forget, and output gates. This enables the model to selectively remember or forget information over time, making it well-suited for processing sequential data. By learning patterns and dependencies in the input sequence, the LSTM network can make accurate predictions for classification tasks, such as sentiment analysis, speech recognition, and natural language processing.

LSTM neural networks are usually used for sequential data like time series, natural language processing, etc. In the given problem context, although LSTM performed better than the above models, it did not perform very well because the dataset does not have a sequential nature. The problem is essentially a supervised classification problem, where the type of crime is to be predicted based on latitude, longitude, and time of day. LSTMs are not the best choice for such problems as they are designed to learn patterns in sequential data.

### D. Random Forest Classifier

Random Forest Classifier is an ensemble machine learning algorithm that combines multiple decision trees to make predictions. It randomly selects subsets of features and data samples for each tree, reducing correlation and improving robustness. By aggregating the predictions of individual trees, it produces a final prediction through voting. Random Forest is known for its effectiveness, scalability to large datasets, and ability to handle high-dimensional feature spaces. It is widely used for classification tasks, providing reliable predictions and feature importance insights.

The classification report for this model is shown in Table I. Random Forest Classifier was the best model for crime prediction with an accuracy of 97% due to the following characteristics:

- Ability to handle non-linear relationships: Random Forest Classifier can capture the non-linear relationships between input features such as latitude, longitude, and time of day and the output variable Category (of crime).
- Ability to handle high-dimensional data: Random Forest Classifier can effectively handle high-dimensional data and can deal with both continuous and categorical variables.
- Robustness to noise and outliers: Random Forest Classifier is less sensitive to noise and outliers in the data as it is built from multiple decision trees. Outliers and noise in the dataset can negatively affect other models, such as Logistic Regression and Gradient Boosting Classifier.
- Less prone to overfitting: Random Forest Classifier can avoid overfitting by randomly selecting subsets of features and samples during training, thus reducing the variance in the model. This is particularly important when dealing with high-dimensional data.

| class | precision | recall | f1-score | support |
|---|---|---|---|---|
| Assault | 0.96 | 0.96 | 0.96 | 31799 |
| Burglary | 0.95 | 0.94 | 0.94 | 19588 |
| Kidnapping | 0.66 | 0.75 | 0.70 | 667 |
| Robbery | 0.90 | 0.91 | 0.90 | 10766 |
| Stolen property | 0.70 | 0.78 | 0.74 | 2350 |
| Theft | 0.99 | 0.99 | 0.99 | 101195 |
| Vehicle theft | 0.98 | 0.98 | 0.98 | 24063 |
| | | | | |
| accuracy | | | 0.97 | 190428 |
| macro avg | 0.88 | 0.90 | 0.89 | 190428 |
| weighted avg | 0.97 | 0.97 | 0.97 | 190428 |

TABLE I: Classification Report

## VI. App Implementation

Figure 3 shows the frontend of our application, which is created using React framework. After the user enters the address in the textbox and hits the Submit button, the address is passed to the by making 2 API calls to the "/plot" and "/predict" endpoints. The POST request to "plot" generates an iFrame that shows the heatmap of crime in San Francisco along with the entered location marked on it. The other POST request to the "predict" endpoint uses Random Forest Classifier model, which we have trained and saved for predicting the most probable crime at the given address at the current time. We use Axios, which is a popular JavaScript library, to make HTTP requests to our server.


Fig. 3: App frontend screenshot

For the backend implementation, we used FastAPI- a modern, high-performance framework for building APIs with Python, and Uvicorn- a lightweight Asynchronous Server Gateway Interface (ASGI) server to run the FastAPI application. The model is trained and saved on the server startup event, which is a functionality of FastAPI that allows us to execute code when the server starts up. Google Maps API is used to get the latitude and longitude from the address provided by the user, and we used Folium- a python library for creating interactive maps and visualizing geospatial data, to generate the map plot.

## VII. Conclusion & Future Work

In this study, we conducted a comprehensive analysis. We determined that the random forest classifier model exhibits high accuracy in predicting crime trends and hotspots using both historical and real-time data. Compared to other models, such as logistic regression, gradient boosting, and LSTM neural network, the random forest classifier outperformed them. The model's capability to handle high-dimensional data and its robustness against noise and outliers make it less prone to overfitting, resulting in an overall strong performance. Although the model in this study was specifically adapted for analyzing crimes related to public and travelers' safety, it can be easily extended to analyze and predict crime patterns for law enforcement and public safety purposes. It is important to note that the model focuses on predicting the probability of crime occurrence rather than providing certain assertions about specific incidents. Future research can explore social, economic, and environmental factors within specific locations, as these factors can significantly influence public safety.

## References


[1] W. Safat, S. Asghar, and S. A. Gillani, "Empirical Analysis for Crime Prediction and Forecasting Using Machine Learning and Deep Learning Techniques," IEEE Access, pp. 1–1, 2021, doi: https://doi.org/10.1109/access.2021.3078117.

[2] A. Araujo, N. Cacho, A. C. Thome, A. Medeiros, and J. Borges, "A predictive policing application to support patrol planning in smart cities," 2017 International Smart Cities Conference (ISC2), Sep. 2017, doi: https://doi.org/10.1109/isc2.2017.8090817.

[3] Yao, Shuyu and Wei, Ming and Yan, Lingyu and Wang, Chunzhi and Dong, Xinhua and Liu, Fangrui and Xiong, Ying. (2020). Prediction of Crime Hotspots based on Spatial Factors of Random Forest. 811-815. 10.1109/ICCSE49874.2020.9201899.

[4] Lavanya Elluri, Varun Mandalapu, and N. Roy, "Developing Machine Learning Based Predictive Models for Smart Policing," IEEE International Conference on Smart Computing, Jun. 2019, doi: https://doi.org/10.1109/smartcomp.2019.00053.