Spartan Security: Crime Forecasting using Time Series Features

Nikita Bairagi, Tejas Madappa, Divya Sidhabathuni, Swayam Swaroop Mishra
Computer Engineering Department
San José State University (SJSU)
San José, CA, USA
Email: {nikita.bairagi, tejas.madappa, divya.sidhabathuni, swayam.mishra}@sjsu.edu

***Abstract*—Crime analytics is important for the safety and well-being of people. Violence and harassment in the form of theft, abuse, hate crime, etc. occur on a daily basis. According to the uniform crime reporting program in 2017, a violent crime occurred every 24.6 seconds. Government and officials have placed multiple legislation's and practices in an effort to reduce and provide more safety for people everywhere however more work is required. With the rapid technological advancement in machine learning, a predictive model can be created using past crime statistics to predict future crimes. Further, this predictive model can be integrated with a web platform which will help prevent and warn people of future crimes. Existing techniques used in crime prediction tools such as Predpol and CrimeScan focus on offender's identity, victims of crime, location and time of occurrence, as well as the details of a crime. Other factors are often not considered such as income disparity, number of CCTV deployed, number of police deployed in a particular area. In this project, our goal is to built a web platform based on the predictive model that focuses not only on location, time of occurrence, and details of a crime but also on census data and CCTV camera location in a city. For building the model, we are using standard machine learning pipeline and victim based crime data combined with census data of the Baltimore City. The platform will include an accurate predicting model that is integrated with a web application for better accessibility and user-friendly experience. The final product is expected to help individuals to be aware and stay safe.**

***Index Terms*—CrimeScan, Crime Prediction, Machine Learning, Predpol, Time Series Analysis, Predictive Analysis**

I. Introduction

Crime analytics refers to a systematic study and research to identify and analyze patterns and trends in crime. Analysis and patterns can help law enforcement agencies more efficiently distribute resources and help police locate and arrest offenders. Using the predictive analysis and time series forecasting, a software solution can be developed which analyzes crime patterns and trends and provides real time prediction of crime locations.

The topic of predictive analysis is gaining a lot of momentum[1][2] and can help in forecasting the future more reliably. Predictive analysis is a branch of advanced analytics to make predictions about future events. It takes in a variety of inputs and predicts the future behavior where all the data points may or may not be numbers. Predictive Analysis gives more actionable insights than traditional descriptive statistical analysis.

Time series forecasting technique uses aggregated crime numbers and predicts the value for the data based on trends. We can use this method to predict the number of crimes at a location in conjunction with census data based on income disparity, population, alcohol sale, etc because certain crimes are neither Poisson distributed nor have seasonal trend[4]. Using these methods in crime prediction can help law enforcement and citizens be prepared. A serious crime can be detected before it occurs and can help warn the people of the area as well as send more police forces for security.

The purpose of this project is to research and develop a machine learning model to forecast the next crime incident and help provide more awareness and safety for those areas at risk. In this project we aim to use the data points such as crime location, time and nature of crime from victim based crime data and combine these data points with the census data. Combining census data with victim based crime data will help us understand the impact of population density on crime. Applying predictive analysis on this data set, we will be building machine learning model to predict number of crimes on a location. To make predictions more specific we plan to train different models for predicting different type of crimes. User interface for the project will be a web application that gives crime analytics insights via maps and graphs and provides flexibility to view predictions based on specific crime and location.

Machine learning algorithms are essential to building a model. Using the most ideal one will help in the learning and building of an efficient system. Currently many modeling techniques are used such as decision trees, regression techniques, and neural networks. Understanding the specifics and the reasoning behind these algorithms is vital toward having a good accuracy percentage and efficient model. Moreover, it is essential to communicate the analytics to the user using data visualization and charts. It is important that the predictions and analytics are presented in intuitive and flexible way to the user. Data visualization can help in drawing inferences, which can be further utilized to improve the model performance. This project provides a web application that provides crime visualization on maps. The application is based on an accurate model for predicting crime and allows students and individuals to feel safer.

II. Project Description

A. Problem Statement

In this project we are building a time series model, that predicts the number of crimes per 1000 residents of a police district in the Baltimore City. For building the model, we first need to find the population of each police district in Baltimore. The census data of the Baltimore provides information on population of each neighborhood in the city. In this project, we are going to combine the victim based crime data and the census data, and find intersections between neighborhoods and police districts. Using this intersection details we aim to find the population of each police district. Along with this, we aim to provide predictive and descriptive analytics on a user friendly web platform. As a user, a person can use this application to view where the crime hot spots are and locations where it is most likely to be unsafe in the future. By using the effective evaluation metrics we can ensure that the results are optimal and accurate as possible. With the rapid increase in crime rate, this application helps provide a solution toward giving users a platform to go to in order to avoid locations where the crime rate may be high.

B. Project Architecture

1) Machine Learning Architecture: Machine learning architecture explains different components of time series prediction technique used in this project. The ML time series model predicts the number of crimes for upcoming week and the results are stored in a csv file, which is then later used by the web application for visualizing the prediction results. The ML architecture is divided into three blocks which represents the standard machine learning pipeline:

a) Data Collection: The data for crimes that occurred in the past is present on the Baltimore Open Data website in the form of victim based crimes and actual crimes that occurred in the city. The data for actual crime that occurred in the city is used for prediction. This data set also contains police district information and accurate location of the crime in the form of longitude and latitude. Additional data like census data is also collected based on neighbourhood.

b) Data Preprocessing: : This block has 4 steps.

1) Data Cleaning : In the data cleaning step the missing and noisy data is removed.
2) Data Transformation:
   a) Time Series Data Transformation: The actual crime data is aggregated in a time series format i.e. the dataset have features such as police district, frequency(daily, weekly, monthly, etc) and number of crimes and different types of crimes and their frequency in that police district. From the time series data created different time series features e.g. absolute energy, tsf-lag, mean change, etc.
   b) Census and Police District Data Transformation: In this step, we label encode the string values in the dataset.
3) After data transformation the raw dataset is created by joining the police district column of the time series feature dataset and police district column from census dataset.
4) In the data reduction step, raw dataset features with low importance are removed using recursive feature elimination (RFE). This step helped in our prediction model to predict more quickly because of the lower dimension of the dataset.

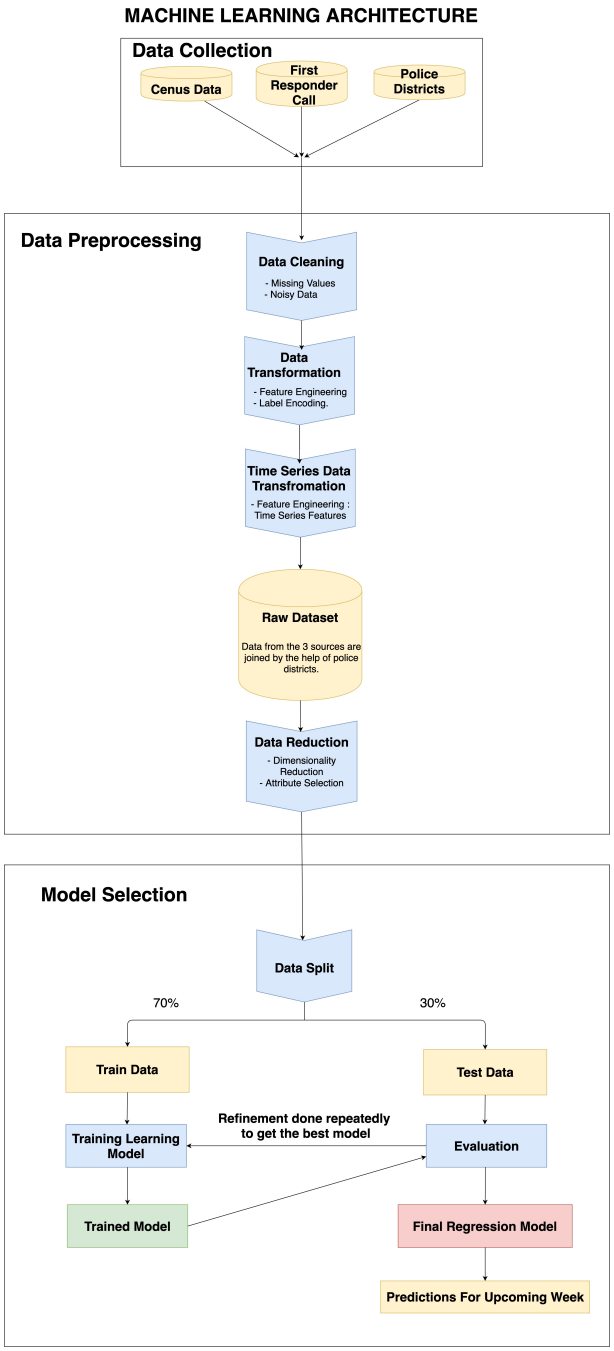

Fig. 1. Machine Learning Architecture

2) Web Application Architecture: Web Application Architecture has three layers: Data Layer, Application Layer and Presentation Layer.

a) Data Layer: The past crime records for the Baltimore city and prediction results for upcoming week are stored in csv

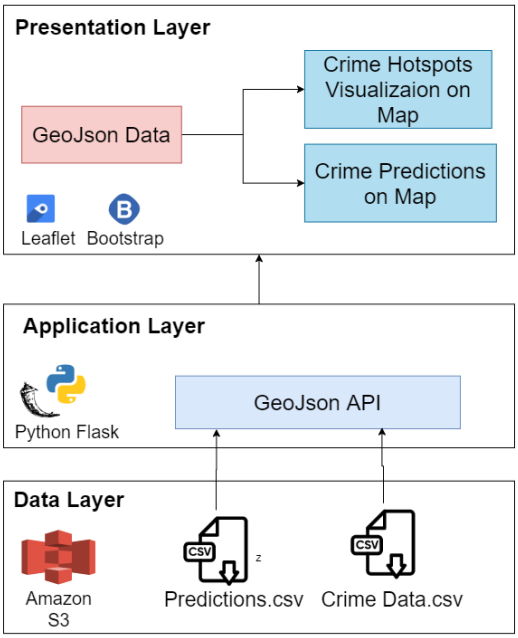

Fig. 2. Web Application Architecture

format on the cloud using Amazon S3 service.

b) Application Layer: The application layer uses Python server to serve the web platform and to store and fetch data from S3 repository. The data in csv file is converted to GeoJson using Public API. This GeoJSON data is used to create visualizations using Leaflet Maps.

c) Presentation Layer: The presentation layer provides the descriptive analytics and predictive analytics of crime in Baltimore Police Districts. Descriptive analytics include visualization for number of crimes in a Police District, crime hot spots and CCTV camera locations. Filters on the map can be used to visualize the pattern for particular crime type. Predictive analytics section displays the number of crimes in a Police District predicted for upcoming week.

III. Method(s) / Technologies Used

A. Integration of Crime and Census data

In order to build the machine learning model that predicts the crime per 1000 people in a police district, we first need to find the population of each police district. Each police district consist of multiple neighborhoods and population of each neighborhood can be obtained from the census data. But various neighborhoods in Baltimore city fall under 2 or 3 police districts. For example, in fig.3 it can be seen that Barclay neighborhood falls under 3 police districts, Northern, Central and Eastern. Therefore, its necessary to find the intersection area of the neighborhood that falls under a particular police district. We have used Turf JavaScript library to find the intersection of police district and neighborhood. The Intersection module of the turf library takes the multi polygon

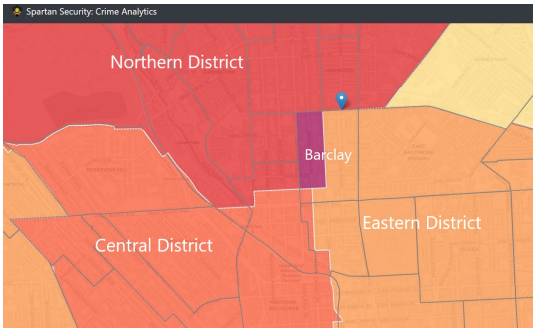

Fig. 3. Police Districts and Neighborhood intersection

geometry of two shapes as input and provides the geometry of intersection between two shapes.

$$var\ intersection = turf.intersect\ (\ policeDistrictShape,\ neighborhoodShape);\quad (1)$$

The area of the intersection is then calculated using Area module which takes shape of intersection as input and gives area in square meters.

$$var\ area = turf.area(\ intersection);\quad (2)$$

Once the intersection area of a the neighborhood is calculated, we can the find the population in that intersection area by using the population density of the neighborhood. The population is divided on the premise that each person is uniformly spread over the area.

$$Population = Population\ Density \times Area()\quad (3)$$

After calculating the population of intersected neighborhoods, we can easily find the population of a police district by adding population of all neighborhoods in that police district. This can be further used by the police to deploy units according to the population for a specific police district to ensure that a stable civilian to police ratio is maintained for effective management of Law and Order situations.This will ensure the efficiency of the beat system.

B. Machine Learning Implementation

1) Data Collection: The victim based crimes dataset of Baltimore city has the location, time of occurrence and details of a crime, neighborhood and police district in which crime occurred. The second dataset, Location of CCTV cameras in Baltimore contains the Latitude and Longitude of the location of the CCTV's. These are overlays used for a base layer for the City of Baltimore.

2) Time Series Feature Extraction: Time Series Features (TSF) is a function that takes past n_features of time as input and maps it down to a single numeric value [10].One of the salient feature of this work is determining the time series features which grant the most for time series crime forecasting. A total of 754 time series features were generated by ts_fresh [9] from the time series dataset. ts_fresh is a python package used for generating time series features. For narrowing down the time series features we used recursive feature elimination with

the model with best prediction performance. The following are the 10 most important time series features obtained from recursive feature elimination.

TSF1 Variance larger Standard Deviation : It denotes a boolean variable for the variance of x if it is greater than its standard deviation.

TSF2 Symmetry Looking : It denotes a boolean variable for distribution of x it looks symmetric. It is represented as,

$$|mean(X) - median(X)| < r * (max(X) - min(X)) \quad (4)$$

TSF3 Large Standard Deviation : It denotes a boolean variable for the following equation,

$$std(X) > r * (max(X) - min(X)) \quad (5)$$

TSF4 Aggregate Linear Trend : It calculates the linear least square regression for values of time series which are collected over blocks versus the range from 0 to the length of blocks minus one.

TSF5 Change Quantiles : It calculates the average, absolute value of successive changes of the time series X inside the quantiles ql and qh of the distribution of X.

TSF6 Mean Change : It is the mean difference between consecutive time series values. It represented as,

$$\frac{1}{n\_feat} \sum_{t=1,\ldots,n\_feat-1} y_{t+1} - y_t \quad (6)$$

TSF7 Variance : It calculates how distant a time series data are spread out from their mean value.

$$S^2 = \frac{\sum (X_i - \bar{X})^2}{n-1} \quad (7)$$

TSF8 Length : It is the length of the time series data.

TSF9 Absolute Energy : It is denoted as the summation over squared values of time series features.

$$E = \sum_{t=1,\ldots,n\_feat} y_t^2 \quad (8)$$

TSF10 Autocorrelation : Serial correlation or auto correlation, is the correlation of a time series data with respect to a slow copy of itself as a function of delay. It is represented as,

$$\frac{1}{(n-l)\sigma^2} \sum_{t=1}^{n-l} (X_t - \mu)(X_t + 1 - \mu) \quad (9)$$

3) Baseline Implementation: The research paper, "Time Series Features for Predictive Policing" served as our baseline. In the paper authors used K-means Clustering to find the different crime hotspots in the Baltimore city. From the time series dataset, time series features which were used as training input to different machine learning regression models were extracted.

In our approach, we used police districts as crime hotspots and created a time series data set from the actual crime dataset of Baltimore City. Using the time series dataset we extracted the time series features using python's ts_fresh module. To reduce the dimensionality we dropped columns with constant features. After dropping, our dimension dropped drastically from 754 features to 20 features.

Using train-test split we checked the performance of three different models, Random Forest Regressor(RFR), Multi-Layer Perceptron Regressor(MLPR) and Support Vector Regressor(SVR). First we trained the models on 284 weeks or 4 years of time series feature dataset and tested on 52 weeks or 1 year of time series feature dataset. Our MLPR and SVR outperformed the baseline model which is explained in details in results section.

4) Expanding Window Forecasting: Expanding window refers to a machine learning model which is trained on available historic data and makes the forecast. It is called expanding window because its size increases as more real observations are collected.

Similar to baseline we created a time series dataset for each police district. The dataset has 5 years of historical crime data for every week that occurred in each police district in Baltimore City. Before the dataset is used for extracting time series features we made forecasting frames using make_forecasting_frames from ts_fresh module. The function takes a singular time series X and creates a dataframe and target y which is used for time series forecasting task. The created dataframe contains each and every time stamp in X, the last max_timeshift data points as a brand new time series which can be utilized to fit a timeseries forecasting model[11].

Once the make_forecasting_frames data was created it was passed to extract_features function of ts_fresh python module to create time series features. A total of 754 features were generated, however based on their uniqueness, the feature list was further narrowed down to 349. Since, expanding window forecasting is compute intensive and time consuming, first we tested our models that are Regressor(RFR), Multi-Layer Perceptron Regressor(MLPR) and Support Vector Regressor(SVR) on one year of data. First we trained the model in 10 weeks time series features dataset and tested on 42 weeks of time series features dataset. For training, the model was trained on 10 weeks of data and predicted number of crimes for 11_{th} week. When the total number of crime for 11_{th} was available it was merged with the previous 10 weeks of data and trained again. After training it predicts the data for 12_{th} week and this process is repeated for all the 42 weeks of testing data.

Using this approach Random Forest Regressor outperformed all other models and thus we used it as our final prediction model and tested its performance on 5 year of data which is explained below.

Our model was trained using 284 weeks or 4 years of historical crime data and tested on 52 weeks or 1 year of data. For training, first the model was trained on 284 weeks of data and predicted number of crimes for 285_{th} week. When the total number of crime for 285_{th} was available it was merged with the previous 284 week of data and trained again. After training it predicts the data for 286_{th} week and this process is repeated for all the 52 weeks of testing data. Performance of expanding window forecasting outperformed baseline approach which is mentioned in details in section V.

C. Web Platform Implementation

The Web platform is developed using Python Flask as the backend server. The API's are developed for fetching and storing the data on Amazon S3 repository. The past crime data and prediction results are stored on the repository using this API. When the application server starts all the data from repository is fetched and converted to Geojson format. This Geojson data is used for generating the Leaflet maps and showing analytics of Baltimore crime. The application consist of three tabs, Predictive analytics, Descriptive Analytics and Results. The results tab include the results from ML model, to shows the accuracy of the prediction model.

1) Data Visualization using Maps: The descriptive analytics tab consist of two maps showing Crimes by Police District and Crimes by Neighborhood. User can hover on a particular district or neighborhood and see the number of crimes and location of crimes. The police district map also shows hot spots for past crimes. Clicking on a particular hot spot allows user to see the exact crime location. Descriptive analytics also shows various graphs showing number of incidents based on zip code, number of crimes per month in a police district. This visuals provide meaningful insights on the crime patter in Baltimore city. For these visuals the also user has the ability to filter the data according to what district or zip code they want to see. Another aspect of descriptive analytics lies in the historical data page with visuals of results from the baseline approach as well as the graphs depicting the accuracy of the predictions from the actual number. With the help of the multiple features such as CCTV locations[13], the type of crime, etc, all the visualizations were created in order to show the analytics.

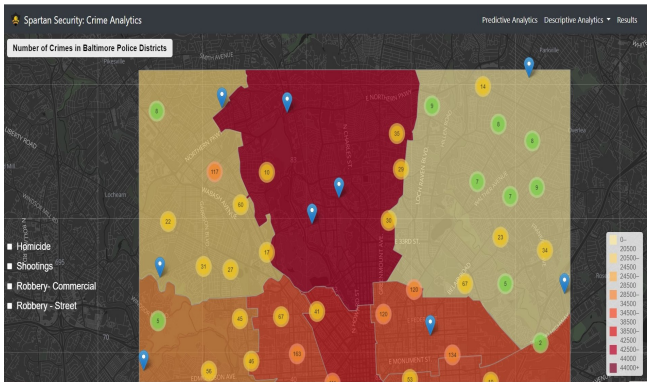

Fig. 4. Web Application displaying crime hotspots

2) Visualization of Forecasting results using Maps: In order to display the number of crimes for the upcoming week generated from the machine learning model, we have a used a map with color coded districts. Each district shows the number of crimes predicted in that district for upcoming week. These maps display the city of Baltimore with pop ups that show the number of crimes to the respective location the user has clicked on. The maps show the level of crime intensity as well as the crime hot spots throughout Baltimore.

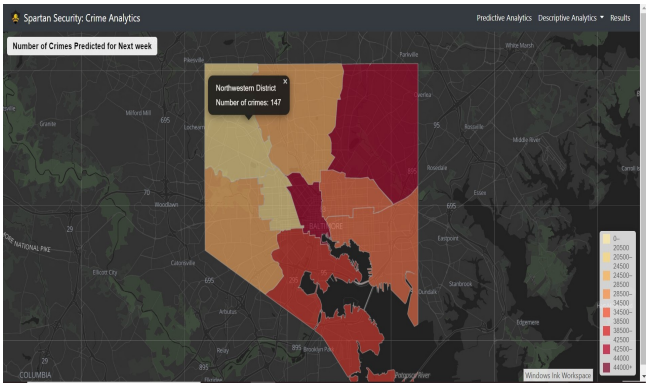

Fig. 5. Web Application displaying prediction results

D. Technologies Used

1) Leaflet: Leaflet is a JavaScript library for building interactive maps. It works efficiently on all mobile and web platforms. We have used leaflet choropleth maps to plot the crime intensity in different police districts and neighborhoods of Baltimore city. And leaflet's MarkCluster plugin is used for plotting the crime hot spots.

2) Turf: Turf is JavaScript library for spatial analysis. It gives helper functions for data classification and statistics tools. We used Transformation module in this library to find the intersection area between Police Districts and Neighborhoods of Baltimore city.

IV. Evaluation Methodology

We are using Baltimore's dataset since it contains actual crime data geolocations[12]. It also provides some additional information such as the number of CCTV cameras present in the location, the latitude and longitude, arrest age, race,sex and police districts. Thus, Baltimore's dataset allows us the ability to form a good understanding on how well our crime prediction model will perform.

A. Mean Squared Error

Since we are dealing with a regression problem statement, our primary evaluation metrics will be Mean Squared Error (MSE) which is average squared deviation between observation and prediction. It is be represented as,

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{10}$$

B. Mean Absolute Error

The second evaluation metric is Mean Absolute Error (MAE), which is the average absolute deviation between observation and prediction. We are using MAE as it is not sensitive to outliers like MSE. It is be represented as,

$$MSE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{11}$$

where, y_i and $\hat{y}_i$ are the actual value and the predicted value respectively.

V. Results

This section contains the results from the baseline and expanding window forecasting. The graphs below represents the crimes that occurred in the different police districts of Baltimore city from 2014-2020.

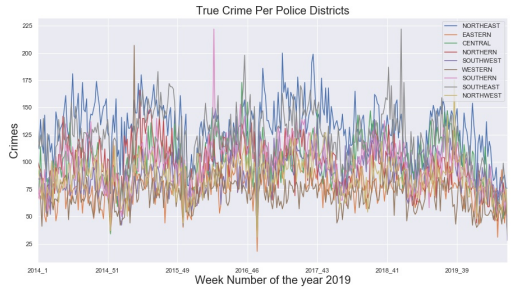

Fig. 6. Crimes in Different Police Districts over 5 years

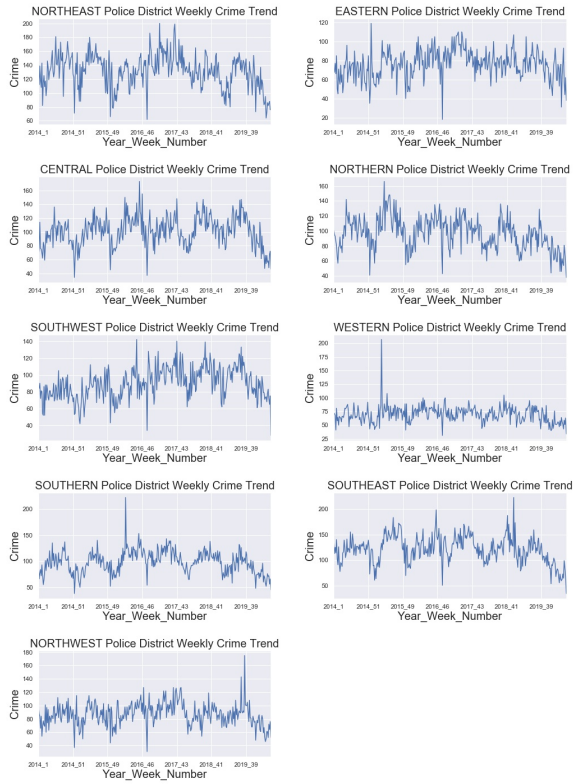

Fig. 7. Crimes in Different Police Districts over 5 years

A. Baseline Results

Our Support Vector Regressor (SVR) and Multi-Layer Perceptron Regressor (MLPR) outperformed the baseline research paper "Time Series Features for Predictive Policing" but our Random Forest Regressor (RFR) model under performed compared to baseline performance. We belive our model performance increased for SVR and MLPR where as decreased for RFR maybe because of the following reasons:

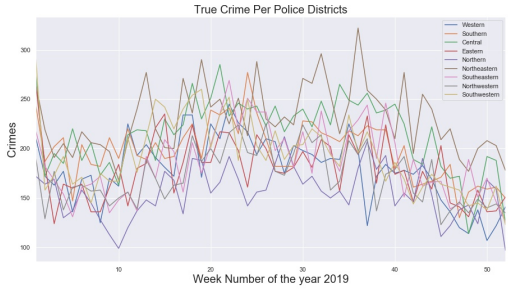

Fig. 8. Crimes in Different Police Districts over 1 year

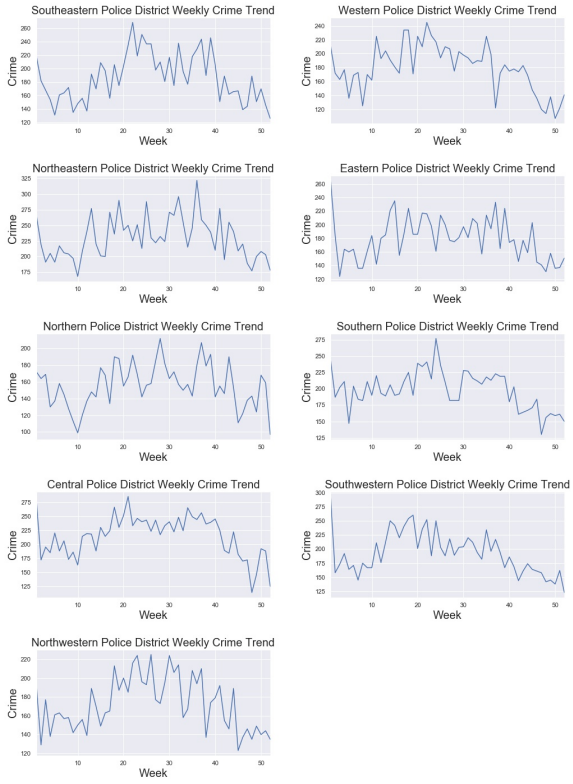

Fig. 9. Crimes in Different Police Districts over 1 year

1) Baseline research paper divided the city based on clustering. We divided the city based on police districts and have 9 sub-units unlike 20 that is mentioned in the baseline paper.
2) The authors considered only crimes with highest priority. We on the other hand considered every type of crime i.e. crime with highest priority as well as crime as low priority crime.

Below table gives the performance comparison of baseline implementation between our approach and research paper

Below graphs shows the performance of our baseline models on 5 years of data.

B. Expanding window forecast results

1) Forecasting without census data: Our approach of expanding window forecasting from forecasting frames out-

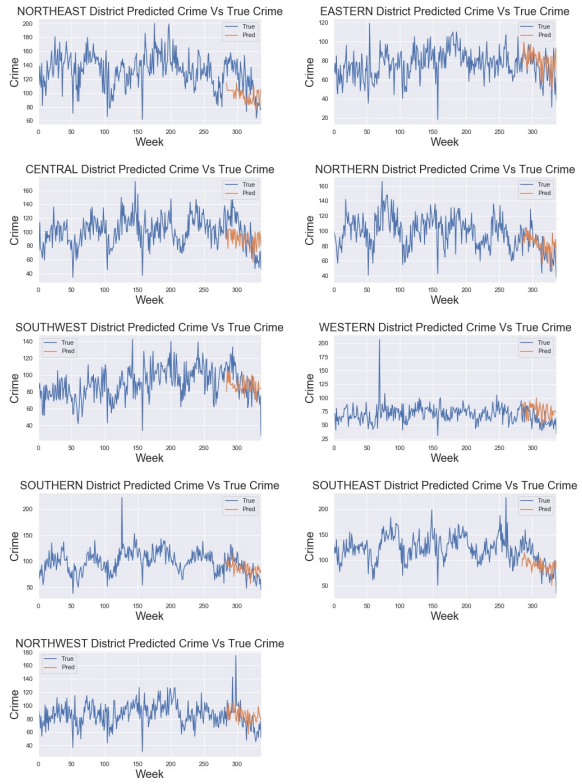

Fig. 10. Baseline Prediction Using Multilayer Perceptron Regressor

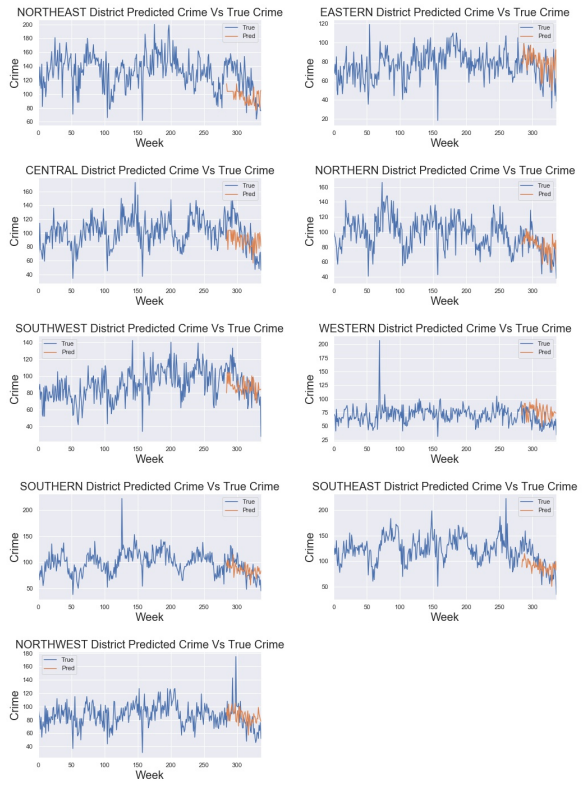

Fig. 11. Baseline Prediction Using Support Vector Regressor

Evaluation Metrics		
Model Name	Our MSE	Research Paper's MSE
MLPR	736.438	1718.571
SVR	736.438	961.326
RFR	2253.103	897.149

TABLE I
PERFORMANCE COMPARISON OF BASELINE IMPLEMENTATION BETWEEN OUR APPROACH AND RESEARCH PAPER.

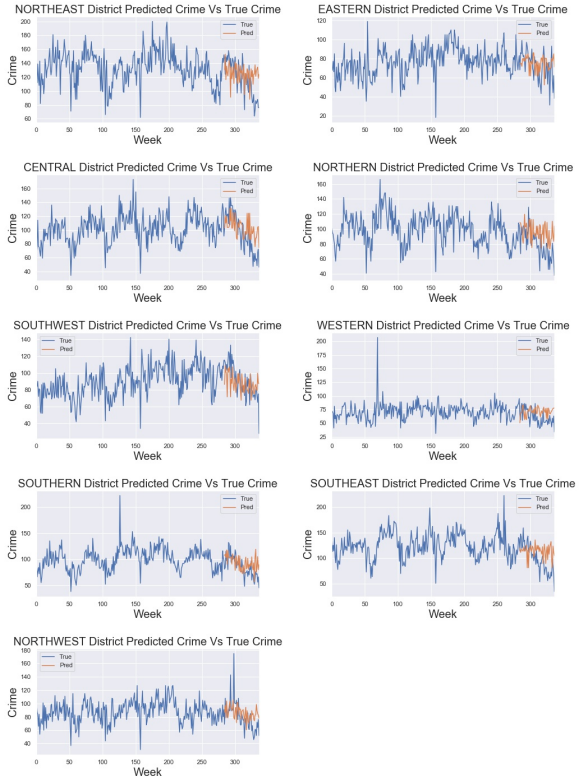

Fig. 12. Baseline Prediction Using Random Forest Regressor

performed the our baseline by 87.5% for all crime types by giving us an average MSE of 273.097 and average MAE of 12.608 for all the districts.

The below table represents the performance of Regressor (RFR), Multi-Layer Perceptron Regressor (MLPR) and Support Vector Regressor (SVR) on 1 year of time series data. In which RFR outperformed all other models. This out performance of RFR helped us to focus on one model for expanding window forecasting approach.

Evaluation Metrics		
Model Name	MSE Value	MAE
MLPR	22448716335.04	73574.90
SVR	1285.96	29.14
RFR	824.18	22.84

TABLE II
PERFORMANCE OF DIFFERENT REGRESSION MODEL ON 1 YEAR OF TIME SERIES DATA USING EXPANDING WINDOW FORECASTING.

The below table depicts the performance of Random Forest Regressor(RFR) on 5 year of data for all crime types.

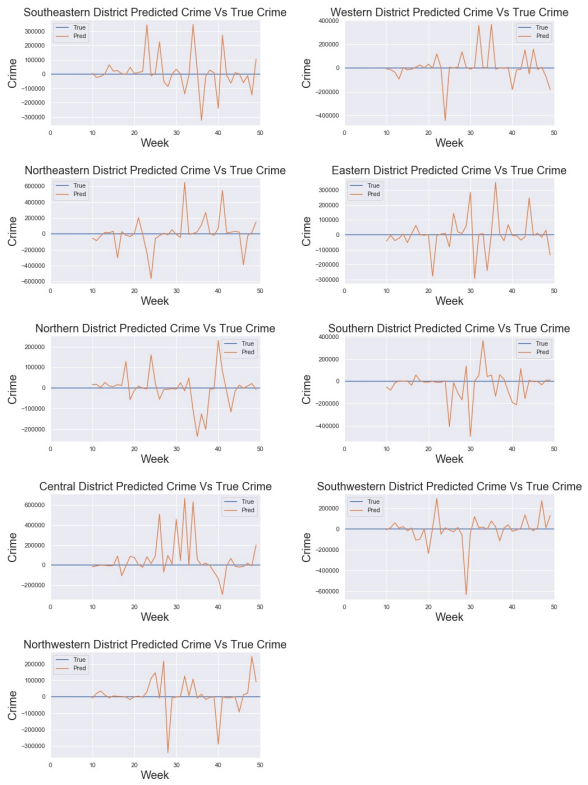

Fig. 13. Predicted Crime Vs True Crime Using Multilayer Perceptron Regressor on 1 year data.

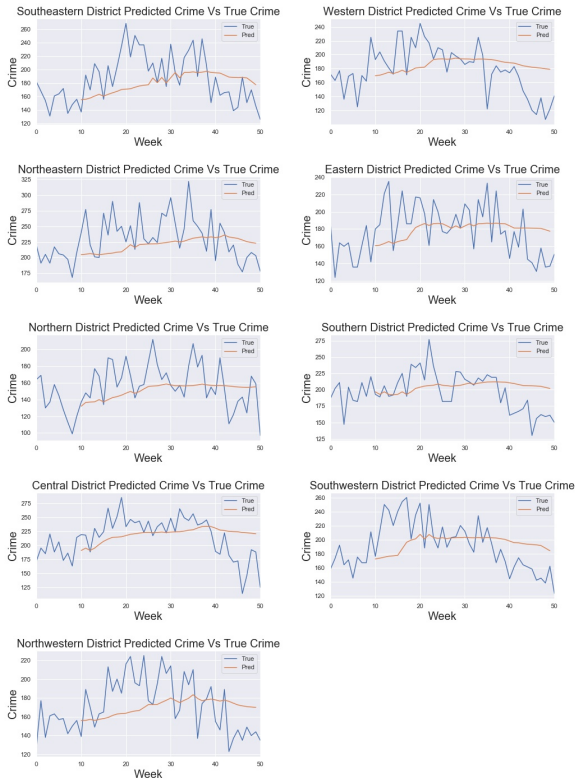

Fig. 14. Predicted Crime Vs True Crime Using Support Vector Regressor on 1 year data.

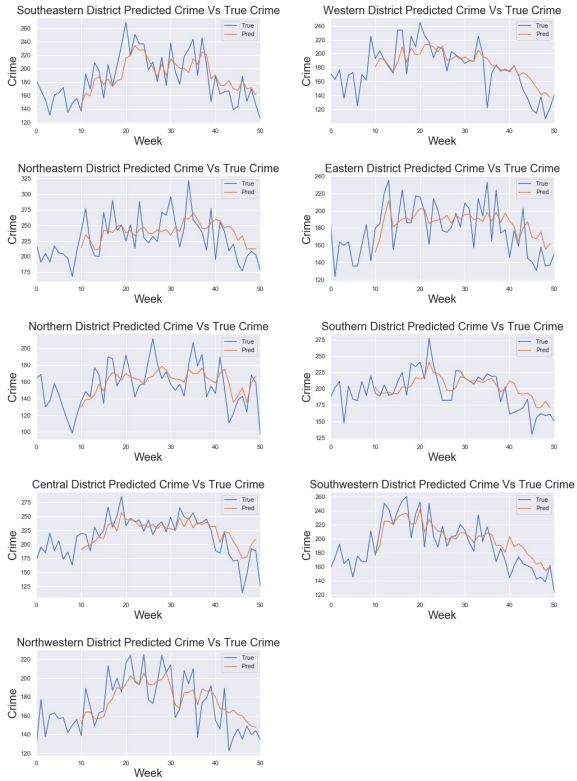

Fig. 15. Predicted Crime Vs True Crime Using Random Forest Regressor on 1 year data

Evaluation Metrics		
Crime Types	MSE Value	MAE
All Crimes	273.0974	12.608
Assault	53.488	5.958
Homicide	0.856	0.727
Rape	0.579	0.631
Robbery	194.416	10.671
Shooting	3.161	1.349

TABLE III
PERFORMANCE OF PERFORMANCE OF RANDOM FOREST REGRESSOR ON 5 YEARS OF TIME SERIES DATA USING EXPANDING WINDOW FORECASTING.

The below graph depicts the performance of our model for every crime type.

The performance of our model on sub crime type can be seen from the below graphs.

2) Forecasting with census data: We implemented two approaches with census or population data to make forecasting. We tried to use population as another feature to the time series feature and the other approach was to predict number crimes per 10,000 people in a police district.

The prediction obtained from first approach i.e. population as a feature column had similar MSE and MAE output. This might be since the values in the population column never changed for any row.

The predictions obtained in the second approach i.e. crimes per 10,000 were first normalized the number of crimes per week to 10,000 people. The normalization was done by

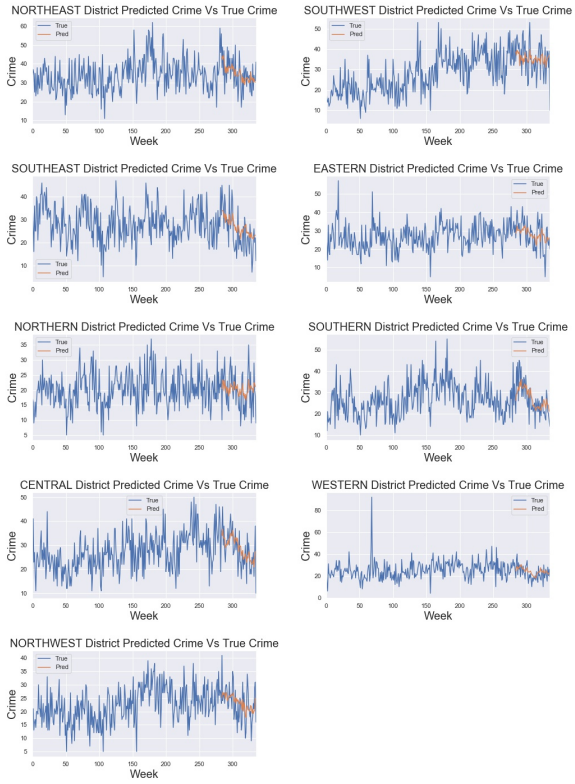

Fig. 16. Predicted Crime Vs True Crime Using Random Forest Regressor for all crime types.

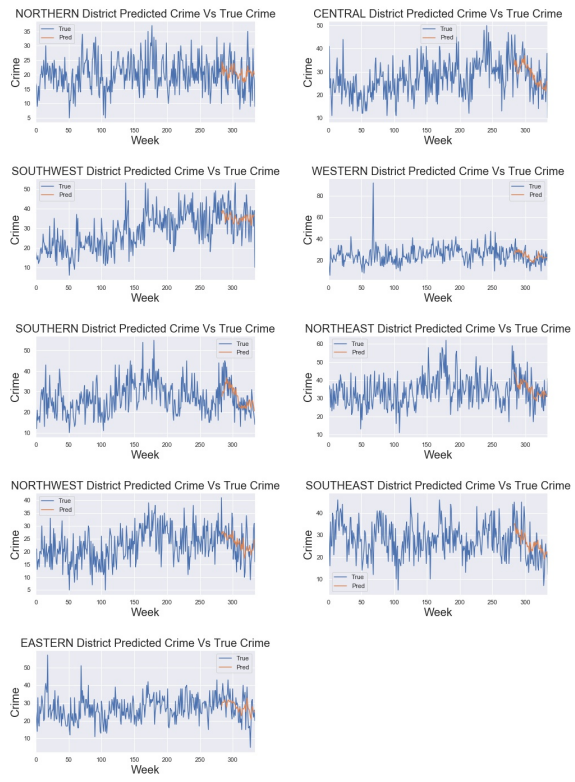

Fig. 17. Predicted Crime Vs True Crime Using Random Forest Regressor for Assault crimes.

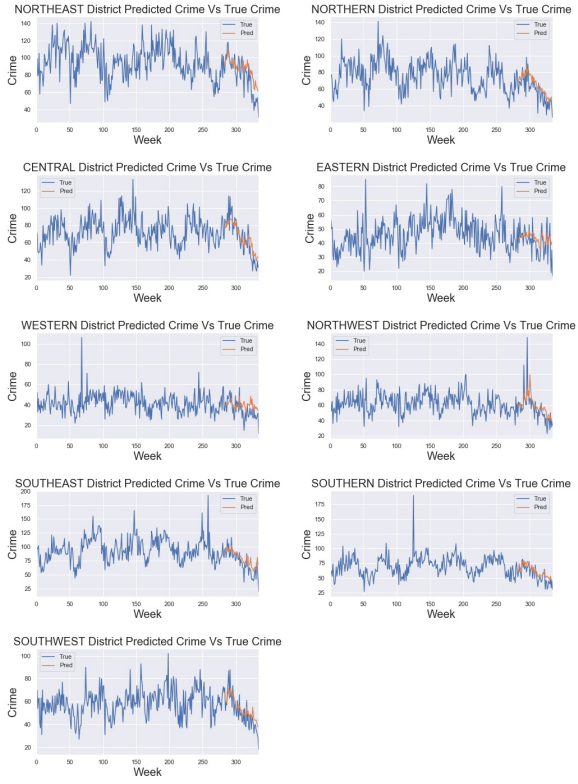

Fig. 18. Predicted Crime Vs True Crime Using Random Forest Regressor for Robbery crimes.

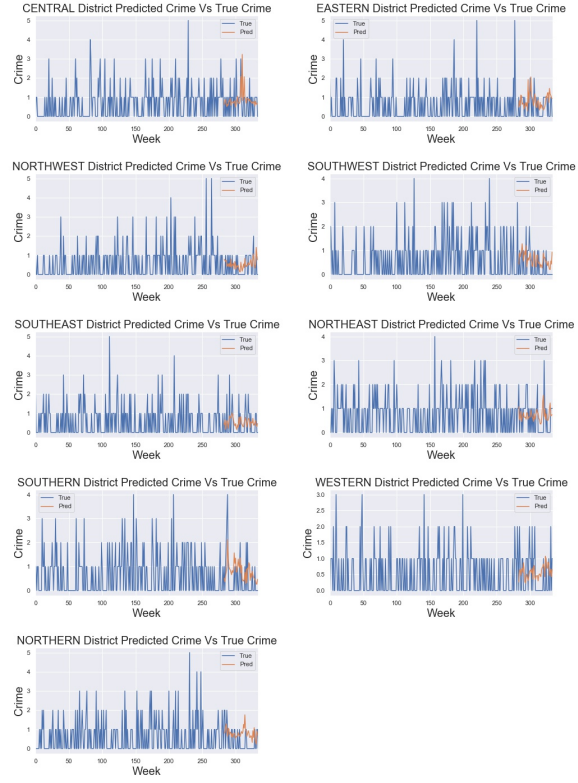

Fig. 19. Predicted Crime Vs True Crime Using Random Forest Regressor for Rape crimes.

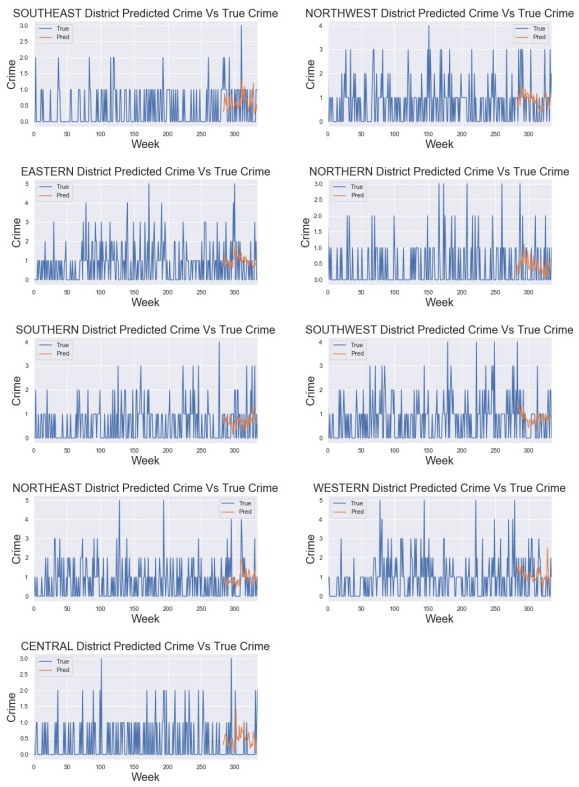

Fig. 20. Predicted Crime Vs True Crime Using Random Forest Regressor for Homicide crimes.

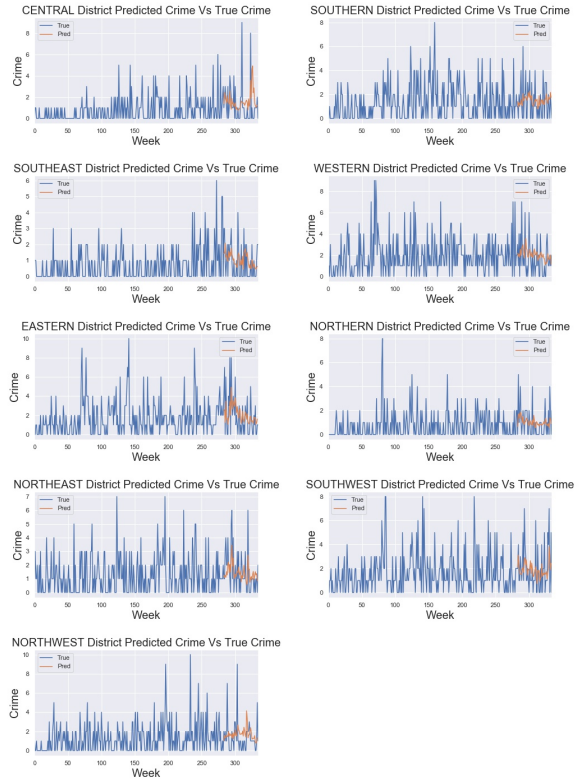

Fig. 21. Predicted Crime Vs True Crime Using Random Forest Regressor for Shooting crimes.

applying unitary method which can represented as follows:

$$C10k = \frac{CPW}{TPPD} * 10000 \tag{12}$$

where, *C10k* is Crimes per 10,000 people, *CPW* Crimes per Week and *TPPD* is Total Population per 10,000 people.

Once the total crimes per week was normalized to 10,000 people we again followed the same expanding window forecasting pipeline. The evaluation metrics obtained from this approach were very promising prediction for frequent total crimes and crimes which are frequent. The below table depicts the performance of Random Forest Regressor(RFR) for 10,000 people.

Evaluation Metrics		
Crime Types	MSE Value	MAE
All Crimes	9.029	2.165
Assault	1.866	1.048
Homicide	0.043	0.076
Rape	0.026	0.053
Robbery	5.387	1.723
Shooting	0.185	0.268

TABLE IV
PERFORMANCE OF PERFORMANCE OF RANDOM FOREST REGRESSOR ON 5 YEARS OF TIME SERIES DATA PER 10K PEOPLE USING EXPANDING WINDOW FORECASTING.

The below graph depicts the performance of our model for every crime type.

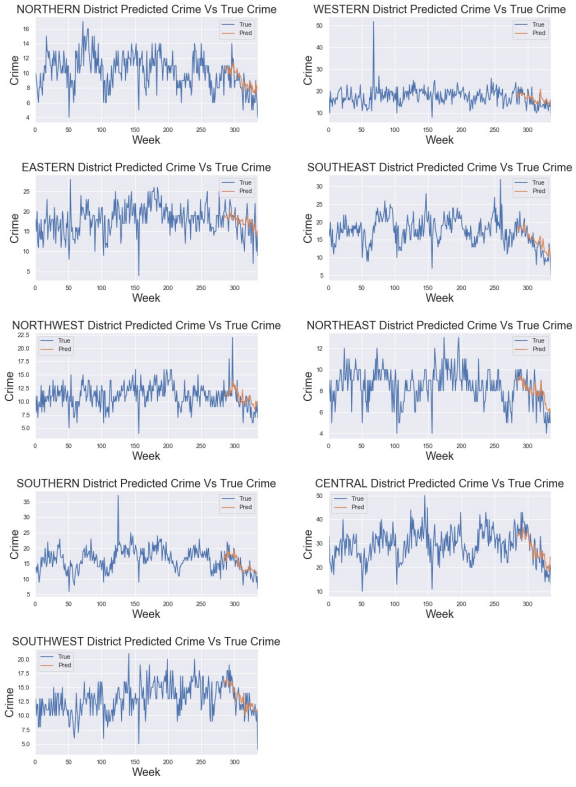

Fig. 22. Predicted Crime Vs True Crime Using Random Forest Regressor for all crime types per 10k people.

The performance of our model on sub crime type can be seen from the below graphs.

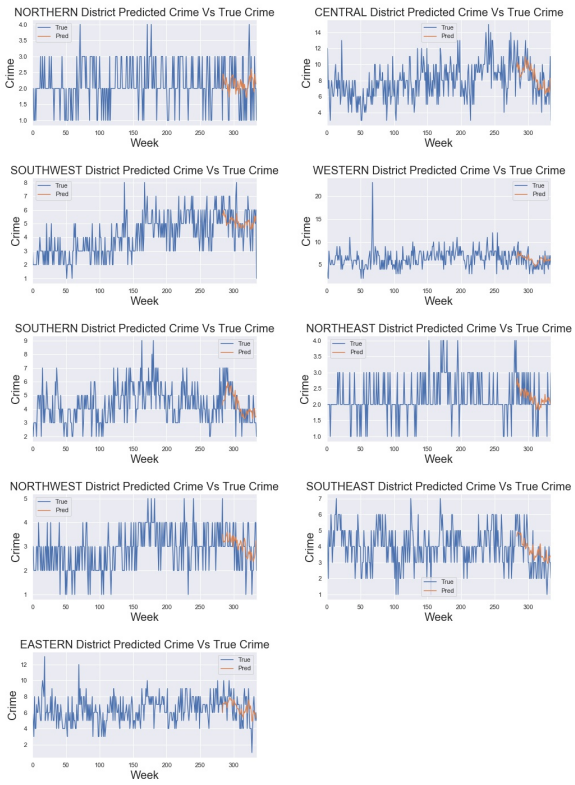

Fig. 23. Predicted Crime Vs True Crime Using Random Forest Regressor for Assault crimes per 10k people.

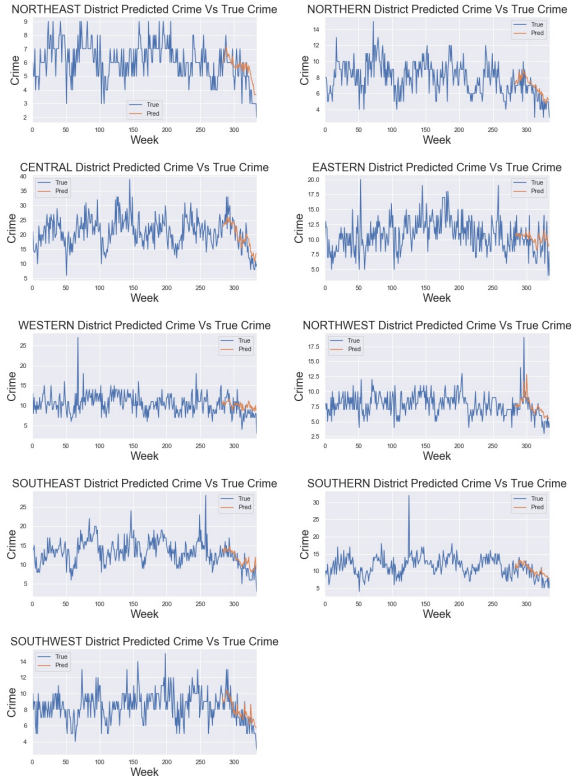

Fig. 24. Predicted Crime Vs True Crime Using Random Forest Regressor for Robbery crimes per 10k people.

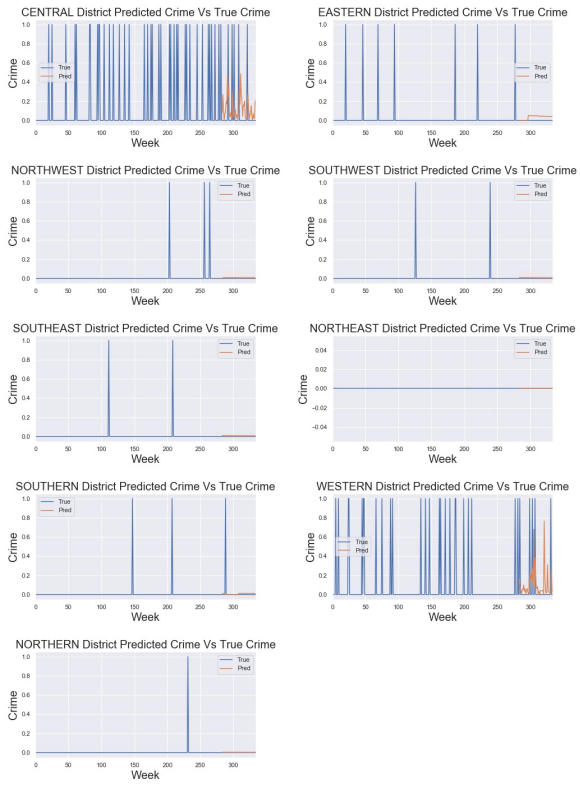

Fig. 25. Predicted Crime Vs True Crime Using Random Forest Regressor for Rape crimes per 10k people.

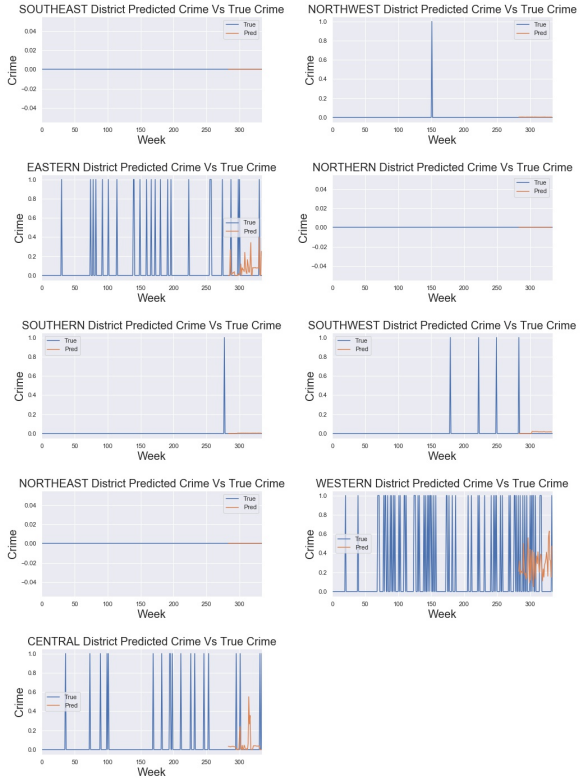

Fig. 26. Predicted Crime Vs True Crime Using Random Forest Regressor for Homicide crimes per 10k people.

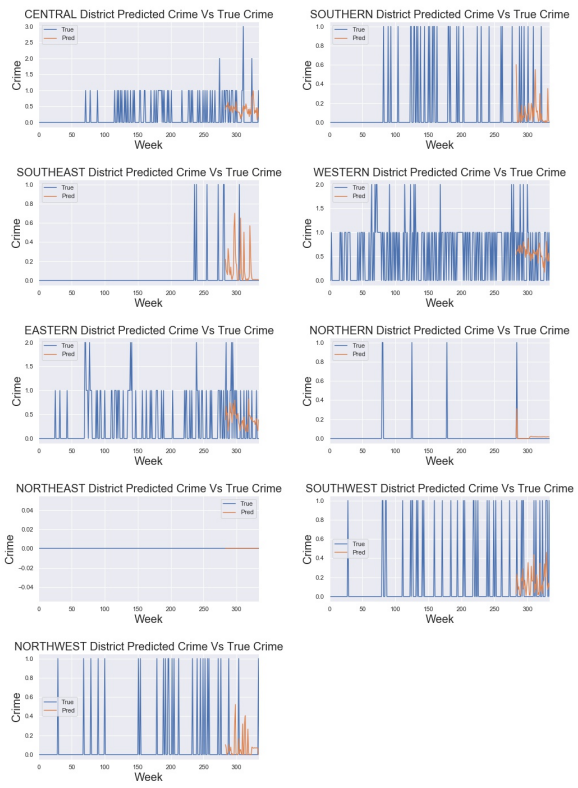

Fig. 27. Predicted Crime Vs True Crime Using Random Forest Regressor for Shooting crimes per 10k people.

Uncommon crimes like shooting, rape and homicide are not common so we get 0 crimes in certain police districts. For this police district we have to normalize our crimes per week data from 10,000 to less number of people.

VI. Discussion

Some of the challenges faced during the implementation of the project included our migration from San Jose calls for service to Baltimore City Victim based crime since the geolocations for each of the crimes committed and clear demarcation for the police districts available in the form of multipolygons was available with Baltimore Data which was beneficial for our study. A notable mention for the combination of census for Baltimore city's neighborhoods and police districts and combining the two datasets. Another challenge with San Jose Census data was the availability of Neighborhood census which is protected by data privacy laws for the State of California. Very few locations for San Jose CCTV cameras information was provided which made the data inaccurate and hence Baltimore had a dataset with CCTV data. During the architecture implementations there were two approaches to build the system. The first consisted of building a system where the different Machine learning models could be uploaded to a backend and the system be trained. The other approach is to carry out pre-processing of data using Machine learning and store the results in a database. We went with the first approach since the Census is carried out in different capacities at different locations thereby building a generic system which can be adopted provided similar dataset.

VII. Related Works

Many existing online applications and software tools are available that serve the purpose of performing crime forecasting. These applications consist of many user friendly visualizations showing their output which makes them an ideal option for crime prediction. Since our work can be divided into two categories, we are following related crime forecasting works and state of the art web applications present till date.

For creating an effective crime forecasting model, we are focusing on works that analyze crimes of a particular place at a particular point in time and estimate the probability of criminal activity in the future. One instinctive method is to divide the eligible area into smaller sub regions and treat them as independent units. This methodology removes the spatial component allowing time series prediction to be done directly on these unique sub areas [3], [4] and [5] . In article [4], the researchers used time series analysis with different demographics features like unemployment rate, population age distributions, alcohol sale, etc.

For baseline of our methodology we are following a related work by J. Borges et al[6]. The approach proposed in this work used time series data from the historical crime records for crime prediction. The authors extracted different time series features from the time series data using tsfresh and used recursive feature elimination to remove features with high correlation. Doing so reduced the dimensionality of the time series data. Once the reduced dimension dataset was obtained the authors used different regression models to predict the total crime in an area.

A similar work on Crime Forecasting is Crime Hotspot Detection, i.e. spotting sub-areas with high criminal activities. Borges et al. [7], approached a related problem with classification instead of regression technique. Similarly, they are breaking down the urban area into sub-areas and then obtaining time series features from time series data and urban features from urban space to classify the sub areas as criminal hot spots or not.

For creating our web application features as per state of the art technologies we followed online articles and journal papers on these softwares. Software such as Predpol [1] developed for the Los Angeles Police Department is one example. The software uses Risk Terrain Modeling for predictive policing which locates high risk neighbourhoods as areas of higher crime rates by providing graphical crime analysis through the use of heat maps. In general terms it predicts the areas where serious crime can occur at a particular time period. Currently Predpol, is being used by 60 police departments across the US. This system is also deployed for corporate security.

Other considerations are prescriptive and descriptive analytics. Crimescan and Hunchlab [2] use recent 911 calls for service for its prediction algorithm. Since applications with these potentials are neither free nor open source, we are assuming these applications are using time series prediction as their primary prediction algorithm. The time series prediction

is done obtaining the time series data distributed over time and generating trend, seasonality, etc. Ultimately every solution has an approach toward solving predictive analysis.

Open source software which is available was built for the Crime Forecasting challenge held by National Institute of Justice which brought various methods to solve crime prediction based on spatial temporal events with Kernel methods[8]. Open source solutions use data to render maps on open street maps view and data from FourSquare to determine high risk neighbourhoods in Portland, Oregon.

VIII. Conclusions

The main goal of conducting our research on the topic of crime prediction, we were able to implement both a scalable and reliable system to be able to predict the number of crimes that have a high possibility of occurring in the future. Using the baseline prediction from the research paper and by us we were able to create a standard bar for comparison. The results obtained by our baseline prediction underperformed with Random Forest Regressor whereas our Support Vector Regressor and Multi-Layer Regressor outperformed the prediction in the research paper. The main cause for this mismatch in evaluation metrics might have arised due to dividing the city into less number of subunits and taking every type of crime in contrast to the approach given in the research paper. Our prediction using expanding windows outperformed baseline prediction's random forest by 80

Acknowledgment

The authors are deeply indebted to Professor Mahima Agumbe Suresh for her invaluable comments and assistance in the preparation of this study.We would also like to thank Professor Dan Harkey for his continued support and guidance. We would like to thank Professor Kaikai Liu for his inputs on Industry Standards.

References

[1] Perry, Walter L., Brian Mcinnis, Carter C. Price, Susan C. Smith, and John S. Hollywood. Predictive Policing. RAND Corporation, 2013. Web.

[2] Benbouzid, Bilel. "To Predict and to Manage. Predictive Policing in the United States." Big Data & Society 6.1 (2019): Big Data & Society, July 2019, Vol.6(1). Web.

[3] D. E. Brown and R. B. Oxford, "Data mining time series with applications to crime analysis," in Systems, Man, and Cybernetics, 2001 IEEE International Conference on, vol. 3. IEEE, 2001, pp. 1453–1458.

[4] A. Malik, R. Maciejewski, S. Towers, S. McCullough, and D. S. Ebert, "Proactive spatiotemporal resource allocation and predictive visual analytics for community policing and law enforcement," Visualization and Computer Graphics, IEEE Transactions on, vol. 20, no. 12, pp. 1863–1872, 2014.

[5] C.-H. Yu, M. W. Ward, M. Morabito, and W. Ding, "Crime forecasting using data mining techniques," in Data Mining Workshops (ICDMW), 2011 IEEE 11th International Conference on. IEEE, 2011, pp. 779– 786.

[6] J. Borges, D. Ziehr, M. Beigl, N. Cacho, A. Martins, A. Araujo, L.Bezerra, S. Geisler, "Time-Series Features for Predictive Policing," 2018 IEEE International Smart Cities Conference (ISC2), Kansas City, MO, USA, 2018, pp. 1-8.

[7] J. Borges, D. Ziehr, M. Beigl, N. Cacho, M. Martins, S. Sudrich, S. Abt, P. Frey, T. Knapp, M. Etter, and J. Popp, "Feature engineering for crime hotspot detection," IEEE International Conference on Smart City Innovations (IEEE SCI 2017), 2017

[8] Flaxman, Seth, Michael Chirico, Pau Pereira, and Charles Loeffler. "Scalable High-resolution Forecasting of Sparse Spatiotemporal Events with Kernel Methods: A Winning Solution to the NIJ "Real-Time Crime Forecasting Challenge"." (2018). Web.

[9] M. Christ, A. W. Kempa-Liehr, and M. Feindt, "Distributed and parallel time series feature extraction for industrial big data applications," arXiv preprint arXiv:1610.07717, 2016.

[10] J. Borges, D. Ziehr, M. Beigl, N. Cacho, A. Martins, A. Araujo, L.Bezerra, S. Geisler, "Time-Series Features for Predictive Policing," 2018 IEEE International Smart Cities Conference (ISC2), Kansas City, MO, USA, 2018, pp. 1-8.

[11] https://tsfresh.readthedocs.io/en/latest/api/tsfresh.utilities.html

[12] Districts, Baltimore Police Department, Accessed on:June 14,2020. [Online]. Available:https://www.baltimorepolice.org/districts/find-my-district

[13] CCTV Cameras, Open Baltimore, Accessed on:June 20, 2020. [Online]. Available:https://data.baltimorecity.gov/Public-Safety/CCTV-Cameras/y3f4-umna