Comprehensive analysis of Classical Machine Learning models and Ensemble methods for predicting Crime in urban society

S.R Divyasri (sr.divyasriragu@gmail.com)
R Saranya (saranya@cutn.ac.in)
Central University of Tamil Nadu https://orcid.org/0000-0003-2308-772X
P.Kathiravan (kathiravan.pa@gmail.com)
Central University of Tamil Nadu

Research Article

Keywords: Crime prediction, Dynamic ensemble algorithms, Machine Learning, Supervised Learning , KNORA.

DOI: https://doi.org/10.21203/rs.3.rs-2550707/v1

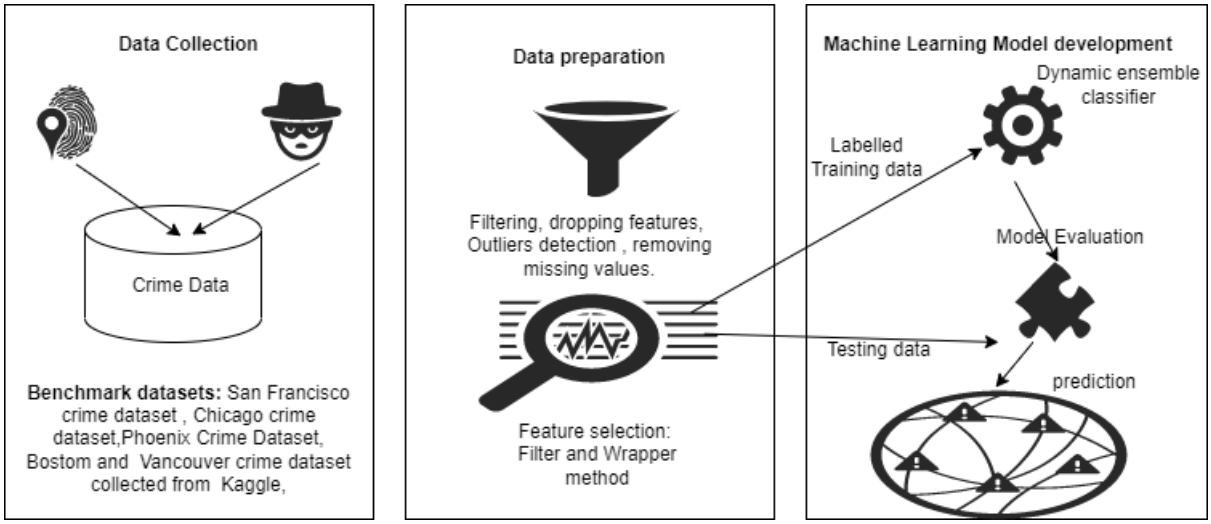
Comprehensive analysis of Classical Machine Learning models and Ensemble methods for predicting Crime in urban society

Abstract

Crimes are a social issue that affects not only an individual but also humanity. Crime classification techniques for crime forecasting are an emerging research area. generally, Crime data are centrally organized with regular maintenance of the criminal registers that can aid officers in sharing observations and improve early alert approaches to keep the citizens secure within their towns. Hence, the aim of this study is to compare the performance of the state-of-the-art Dynamic Ensemble Selection of Classifier algorithms for predicting crime. We used five different benchmark crime datasets (Chicago, San Francisco, Pheonix, Boston, and Vancouver) for this experimental research work. The performance of the state-of-the-art dynamic ensemble selection of classifiers algorithms was evaluated and compared using various performance evaluation metrics such as accuracy, F1-score, precision, and recall. The KNORA Dynamic ensemble algorithms, which select the subset of ensemble members before the forecasting, outperformed the typical machine learning algorithms, and also the traditional ensemble algorithm techniques in terms of accuracy showed that the dynamic ensemble algorithms are more powerful. This ability to predict crimes within urban societies can help citizens, and law enforcement makes precise informed conclusions and preserves the neighborhoods more unassailably to improve the quality of life for humans.

Key words:

Crime prediction, Dynamic ensemble algorithms, Machine Learning, Supervised Learning , KNORA.



Graphical Abstract

1. Introduction

Crime tremendously impacts people's minds, concerns, and spirits, not just its actual effects on society. As a result, law enforcement agencies continue to monitor controlled areas to notice suspicious activity, become more vigilant, and improve their ability to prevent potential criminal activity (Abouelnaga, 2016; VijayaKumar & Alhadidi, 2014).

Machine learning in this era of Artificial Intelligence (AI) is a gravitating topic for effectively conducting analysis and prediction (Alasadi & Bhaya, 2017; Qazi & Wong, 2019). Lately, many research works have been carried out on Crime analysis and forecast using various prediction models and methodologies to peek at the trends and patterns of past crimes, which could further help to indicate and control the expected upcoming crime that may happen in advance (Ahmed et al., 2022). This research paper organizes some of the major topics to be investigated in the crime detection and prediction techniques in machine learning and summarizes the superior methods like dynamic selection and dynamic ensemble selection algorithms for better accuracy. We will also present the future challenges and research gaps that will help scholars redefine the problems in crime analysis and prediction with various machine learning algorithms that are universally suitable for all datasets.

A few years ago, AI algorithms were limited to just the field of work for which they were processed. Nevertheless, computers could move beyond doing what they were programmed for and developing with every iteration of machine learning.

Machine learning also has an adequate flow of mixed and organized data required for a powerful ML solution. Many companies in the modern online world have access to a tremendous amount of data regarding their clients, usually millions of data. This data, which is immense in the number of data points and fields, is comprehended as big data because of its sheer amount of information, which is time-consuming and challenging to process by mortal means (Rácz et al., 2021; Yuki et al., 2019).

The more pure, usable, and machine-readable data in an expansive dataset, the better practical the training of the ML algorithm will be. Currently, ML algorithms are trained by employing three main ways Supervised learning, Unsupervised learning, and Reinforcement learning.

The objective of this experimental research work is to address the following research question:

RQ 1: What is the state-of-the-art dynamic ensemble of selection classifier algorithm for crime prediction?

RQ 2: Can we use a common ML classifier for different datasets (crime dataset)?

Our proposed work uses supervised learning, which is acquainted with labeled data. In supervised learning, the ML algorithm is presented with a small training dataset that is part of the larger dataset to operate with and helps provide the algorithm with a fundamental concept of the problem, the solution to that problem, and which data points to be encountered in the future. The trained dataset is significantly similar to the final dataset with its features and delivers the algorithm with the labeled parameters needed for the problem. Supervised algorithms find the relationships between the given parameters, effectively demonstrating a reason and outcome association between the variables in the dataset, thus explaining how the data functions and the affinity between the intake and the outcome of the algorithm.

Further, we moved one step forward to find the solution for lesser accuracy produced by the linear machine learning model that paved the way to experiment with several other algorithms. We came across many improved data pre-processing methods, such as feature engineering, which helps increase the model's efficiency. Different cross-validation techniques were applied to the datasets to check up on the algorithms working (Albahli et al., 2020).

Then we explored Ensemble supervised machine learning classifiers (Britto et al., 2014; Cruz et al., 2018; Hajela et al., 2021; Vassallo et al., 2021). The basic algorithms that return only a single hypothesis tend to suffer from three main problems. The problems include statistical problems due to high variance, computational friction, and representational issues that are highly biased, some of which can be overcome by using the ensemble method (Aldossari et al., 2020).

We have then introduced the dynamic ensemble algorithms to the pre-processed data, which dynamically selects one of the multiple trained models to make the forecasting based on specific input criteria. This field of dynamic selection has met with great success in many problems. These algorithms typically divide the input feature space and assign particular models to predict each partition. There are many different DCS algorithms, and our research focuses on ameliorated efficiency for accurately classifying crimes compared to previously achieved algorithms (Ko et al., 2008).

Results from each part of the research are compared to choose the best-performing algorithm that gives a better outcome. The rest of our study is arranged in the following paper. Section 2 has the sum up of the related works. Section 3 has the methodology used, i.e., mainly the dynamic ensemble learning algorithms. Section 4 presents the result and discussion, while section 6 contains the conclusion of our study.

2. Related works

In this paper, we have proposed dynamic selection and dynamic ensemble selection architectures for predicting crime test data. We have used the Dynamic Algorithms OLA,

KNORA-E, and KNORA-U. Apart from this, we have explored various methods and techniques to improve the accuracy of basic machine-learning algorithms through Ensembles, Cross-validation, and data pre-processing. Dynamic Ensemble selection classifiers modeling is one of the recently buzzed research fields. Using DES methods for crime prediction can help society and law enforcement to be more accurate and avoids confusion about which model to select and apply to the recently updated dataset. But unfortunately, it hasn't been considered for crime prediction and analysis. Our paper has implemented a few research gaps, such as the possibilities of recognizing different algorithms, using multiple algorithms on different datasets to ensure efficient working, directions of topics to increase accuracy, and whether the result obtained is consistent on all the databases.

In order to address the problem of locating adequate human trafficking data to permit machine learning solutions to analyze human trafficking data, the authors have provided a dataset and generalized dataset creation framework. For the state of Kentucky, this solution aggregates crime datasets from many sources to enable researchers to find patterns and information that would not be visible otherwise (Ahmed et al., 2022).

Authors have (Jangra et al., 2019) presented machine learning data mining techniques in Crime prediction. They evaluated that the Naïve Bayes algorithm performs better than the KNN, which is considered the best with respect to the base paper in terms of accuracy. In (Yuki et al., 2019), authors used the comparative analysis using accuracy with algorithms like Random Forest, Decision Tree, and ensemble algorithms such as Extra Trees, Bagging, and AdaBoost on the Chicago dataset, where the bagging algorithm shows higher accuracy.

Authors (Aldossari et al., 2020), compared the crime data prediction CC, accuracy, precision, recall and ROC of Chicago data with Naïve bayes and Decision tree algorithms. The Decision tree algorithm proved to work better on forecasting the selected features in test data.

Authors use text mining to extract logical relationships from unstructured crime data. In order to uncover multi-level linkages across crime entities, they specifically provide an associative questioning-based searching approach. They used this approach with partition clustering to

create a collaborative, human-assisted data mining and knowledge discovery process (Qazi & Wong, 2019).

In (Wibowo & Oesman, 2020), the authors proved that Naïve bayes shows highest accuracy among k-NN (all optimal value of k), Naive Bayes, and Decision Tree for the dataset of Sleman Regency. In (Albahli et al., 2020), researchers proposed naïve Bayes with feature selection methods FAMD has shown more accuracy than the PCA method on Saudi Arabia crime data. (Sri et al., 2020) did FBI crime analysis with the Chicago dataset using Decision Tree, Gaussian Naïve Bayes, k-NN, and Logistic Regression by predicting crimes classifying, pattern detection, and visualization.

(H. Wang & Ma, 2021) suggested the crime prediction model based on SVM and random forest algorithm can forecast the incidence of crime, and the trend of its forecasted data is consistent with the direction of actual data; this model that is established can effectively predict criminal behaviors that endanger public health and provide reliable data for prevention.

(Safat et al., 2021) applied various machine learning techniques to predict more than 35 crime types in Chicago and Los Angeles, such as logistic regression, SVM, Naïve Bayes, KNN, decision tree, MLP, random forest, and XGBoost, and time series analysis evaluated with RMSE and MAE by LSTM and ARIMA model to fit the crime data better. (Kshatri et al., 2021) revealed that the assemble-stacking-based crime prediction method (SBCPM) based on the SVM algorithm achieves domain-specific configurations compared with another machine learning model, J48, SMO Naïve byes bagging, and the Random Forest. They also proved that any empirical data on crime is compatible with criminological theories and suggested that the prediction accuracy of the stacking ensemble model is higher than that of the individual.

(Khatun et al., 2021) gave an idea of how crime investigation agencies can utilize data mining to discover relevant precautionary measures from prediction rates using some supervised classification algorithms, namely decision trees, KNN, and random forest algorithms. It focused on forecasting the crime for frequently occurring crimes like robbery, assault, and theft, and test data showed random forest gives the highest accuracy. (Tembusai et al.,

2021)Succeeded in analyzing the performance of the KNN method with the k-Fold of fold-3 Cross Validation algorithm as an evaluation model and the Analytic Hierarchy Process method as feature selection for the data classification process to obtain the best level of accuracy and machine learning model.

(Tamir et al., 2021) explored machine learning models like the Random Forest, KNN, AdaBoost, and Neural Network on the Chicago Police Department's CLEAR (Citizen Law Enforcement Analysis and Reporting) system. Among all four models, the Neural Network has the best outcome. (Khan et al., 2022) presented a crime prediction model by analyzing and comparing Naive Bayes, Random Forest, and Gradient Boosting Decision Tree algorithms. Exploratory data analysis is also performed for identified the patterns and understand the trends of crimes using a crime dataset. The Gradient Boosting Decision Tree prediction model is better than the other two techniques for predicting criminality, based on historical data from San Francisco city.

(da Silveira et al., 2022) alleviated the issues of Chronic kidney disease(CKD) and the necessity of early prediction. They used data from medical records of Brazilians with or without a diagnosis of CKD and presented an oversampling approach based on manual and automated augmentation. They experimented with the SMOTE, Borderline-SMOTE, and Borderline-SMOTE SVM and implemented models based on the algorithms: decision tree, random forest, and multi-class Ada Boosted Decision Trees. They applied the overall local accuracy and local class accuracy methods for dynamic classifier selection; and the KNORA U, KNORA E, and META-DES for dynamic ensemble selection. They also analyzed the models' performances using hold-out validation, multiple stratified cross-validations (CV) and nested CV.

Table 1: summarizes the recent related works mentioned above practically and ideologically.

Reference paper	Key Concept	Observation	Suggestions
Jangra M. et al. 2019[6]	Naïve Bayes Approach for the Crime Prediction in Data Mining.[6]	Naïve bayes technique is applied on Data Mining. The performance was better than KNN.[6]	Other algorithms can be explored as only selective algorithm mentioned in the observation is used on a particular dataset.
Yuki J. et al. 2019 [7]	Predicting crime using time and location data [7]	Several ensemble methods such as Random Forest, Bagging, AdaBoost and Extra Tree Classifier are used on Chicago dataset. Bagging proved to be better.[7]	More variation in the results can occur when implemented with other classifying algorithms in the future.
Bshayer S. et al. 2020[8]	A comparative study of decision tree and naive bayes machine learning model for crime category prediction in Chicago.[8]	Decision Tree and Naive Bayes are applied on a dataset, which was extracted from the Chicago Police Department's Citizen Law Enforcement Analysis and Reporting. However, comparing the two algorithms, Decision Tree performed better than Naïve Bayes algorithm. [8]	Many algorithms can be explored as only selective algorithm mentioned in the observation is used on a particular dataset.
Wibowo A. et al. 2020 [9]	The comparative analysis on the accuracy of k-NN, Naive Bayes, and Decision Tree Algorithms in predicting crimes and criminal actions in Sleman Regency.[9]	The machine learning algorithms such as KNN (with different optimal values of k), Decision tree and Naïve Bayes are applied. Naïve Bayes out performed all the other algorithms mentioned in this paper.[9]	The improvement can be made in terms of algorithm's accuracy as the highest accuracy obtained was 65.59%.
Albahli S. et al. 2020 [10]	Predicting the type of crime: Intelligence gathering and crime analysis.[10]	The influencing factors that impact crime rates in Saudi Arabia is predicted. Various machine learning algorithms are applied with feature selection techniques such as PCA and FAMD are applied to improve the accuracy.[10]	Many advanced machine learning techniques can be used with the combination of multiple feature selection techniques.
Sri L. et al. 2020 [11]	FBI crime analysis and prediction using machine learning. [11]	Used random forest which actually performed well, over the Chicago dataset and compared with other basic machine learning techniques.[11]	Different ensemble techniques stacking, bagging, voting, etc., can be applied to find the better results.
Wang H. et al. 2021[12]	Preventing crimes against public health with artificial intelligence and machine learning capabilities. [12]	The data mining technology of support vector machines and the data classification and prediction capabilities of the random forest algorithm are combinedly applied for predictive modelling.[12]	This paper suggests an idea but the implementation results may be different on various datasets.
Safat W. et al. 2021[13]	Empirical Analysis for Crime Prediction and Forecasting Using Machine Learning and Deep Learning Techniques.[13]	Eight different machine learning algorithms and Deep learning algorithms on Chicago (XGBoost is best performer) and Los Angeles (KNN as best performer) datasets. ARIMA model is applied for time series forecasting.[13]	More Data transformation and Classification techniques can be applied to obtain better accuracy.
Kshatri S. et al. 2021[14]	An Empirical Analysis of Machine Learning Algorithms for Crime Prediction Using	Algorithms like J48, Naïve Bayes, SMO have been chosen for the analysis and	Just the stack generalization method is used, other

	Stacked Generalization: An Ensemble Approach. [14]	together stacking learning Bagging classifier was applied. [14]	ensemble methods can be used to justify the result.
Khatun M. et al. 2021[15]	Data mining technique to analyse and predict crime using crime categories and arrest records.[15]	Various techniques namely KNN, Decision tree, and Random Forest are applied. The Decision tree gives slightly better performance than the Random Forest but it creates an overfitting problem. Hence RF algorithm is better model than the rest of the algorithm.[15]	To achieve better results, we have to come across more crime features instead of fixing some characteristics
Tembusai Z. et al. 2021[16]	K-Nearest Neighbor with K-Fold Cross Validation and Analytic Hierarchy Process on Data Classification.[16]	k-NN method with the k-Fold Cross Validation algorithm as an evaluation model performs really well on the dataset.[16]	Only K-fold CV has been applied but other Cross validation methods can be applied on different algorithms and dataset for deriving a better conclusion.
Tamir A. et al. 2021[17]	Crime Prediction and Forecasting using Machine Learning Algorithms.[17]	Extensive model parameter tuning was applied to some algorithms to increase the accuracy using Chicago dataset. However few models like KNN doesn't show significant improvement.[17]	Only 3 models have been considered for a particular dataset but no proof was shown in the paper that it will work with other algorithms and datasets.
Khan M. et al. 2022 [1]	Predicting and Preventing Crime: A Crime Prediction Model Using San Francisco Crime Data by Classification Techniques. [1]	The Naïve Bayes, Random Forest, and Gradient Boosting Decision Tree are used for predicting the crime category attribute labelled "violent" and "nonviolent" for the San Francisco dataset.[1]	More temporal analysis can be performed and feature engineering can be used to improve the performance.
da Silveira A. et al. 2022 [18]	Exploring Early Prediction of Chronic Kidney Disease Using Machine Learning Algorithms for Small and Imbalanced Datasets.[18]	This paper applied the overall local accuracy and local class accuracy methods for dynamic classifier selection; and the k-nearest oracles-union, k-nearest oracles-eliminate, and META-DES for dynamic ensemble selection. Also analyzed the models performances using the hold-out validation, multiple stratified cross-validation (CV), and nested CV.[18]	The same techniques can be used for other prediction problem due to its versatility to work on different data.

From all these substantial related works research, we came to know that the dynamic classifiers and dynamic ensemble selection of classifier algorithms are not researched for crime analysis and prediction exhaustively. Hence, we explored algorithms like OLA, KNORA U, and KNORA E in dynamic algorithms. Further, this implementation covered many related studies, which will be briefly explained in the following topics.

3. Methodology

In relation to this research, our primary goal is to produce a model that could be useful to the law enforcement unit and to our civilians (More et al., 2021). Our objective is to train our model to accurately classify and forecast the crime category using the test data by using a dynamic ensemble classification algorithm (Keerthi R et al., 2020; Kshatri et al., 2021).This, in turn, could help in planning the deployment of the police force in the area with a high probability of crime occurrences so that it can be prevented prior. The block diagram of our proposed framework is shown in the figure Block diagram (Zhihui Wang & Wang, 2021).

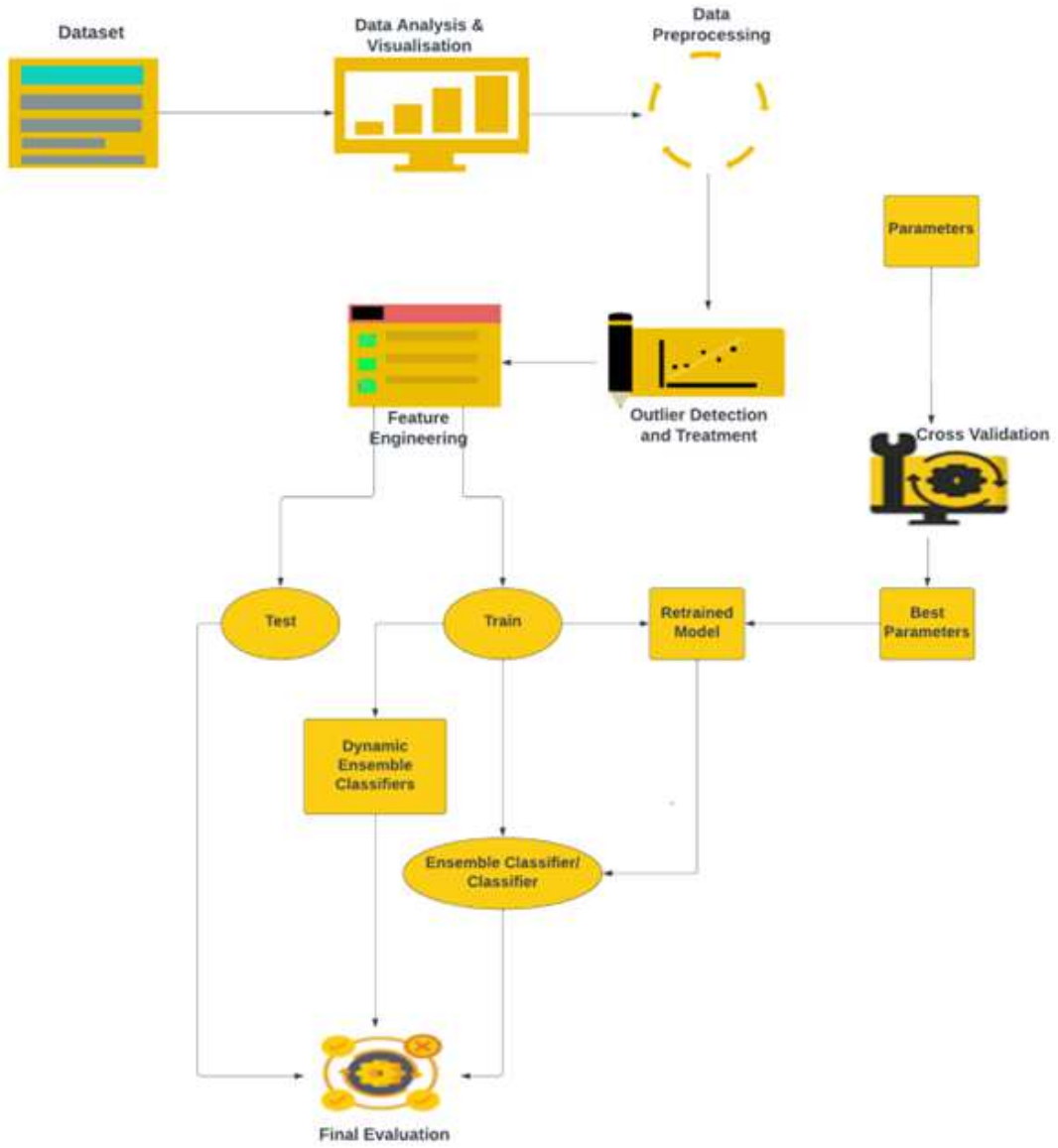


Fig 1 : Block Diagram of the proposed methodology

Figure 1 gives a brief view of the methodology that we have used. The Dataset is analyzed and visualized to understand the data. Then, we pre-processed the data with methods for handling the missing values by data imputation, data type conversions using various encoders, and removal of unclean data. The pre-processed data is used to detect and eradicate the outliers. Then feature engineering properties are applied to select the efficient features for model building. The dataset is split into train and test data. The train data is used for model building using a few classification algorithms first and then ensemble learning techniques. To attain better results, we have used cross-validation to achieve the best parameters and apply them to the same classification algorithm (*Machine Learning and the Internet of Medical Things in Healthcare - Google Books*, n.d.). The dynamic ensemble classifier models are then applied to see the results. A detailed description of the algorithms is explained later in the forthcoming paper.

3.1.Data Collection and Study Area

Data collection is the approach that involves collecting and estimating data from numerous distinct sources. Gathering data permits us to grasp a record of past occurrences to analyze that data to discover systematic patterns. With the help of those patterns and machine learning algorithms, we can build predictive models that peek at tendencies and predict future changes. Proper data collection techniques are essential to design high-performing models. The data should be without errors and include pertinent information for the assignment (Albahli et al., 2020; Zhihui Wang & Wang, 2021).

The data used for this work include the following datasets.

1) The open public San Francisco crime dataset (2003-2015) with 878049 rows in the training set and 884262 rows in the testing set, which has long registered a consistently high rate of crimes from Kaggle, comes in handy as a testing and training dataset (Abouelnaga, 2016; Hale & Liu, n.d.; Hossain et al., 2020).

2) The Kaggle dataset of Crime in Vancouver (2003-2017) contains 624,038 instances of violent crimes.

Table 2 (a): Benchmark Dataset description

Crime Dataset	Crime Record Period	# Instances	# Attributes	# Crime Types/Category
San Francisco	2003-2015	878049	9	38
Chicago	2021-2022	215969	17	31
Vancouver	2003-2017	624038	10	11
Boston	2015-2018	319073	17	34
Pheonix	2015-2021	427843	8	9

Table 2(b): Detailed attributes and labels list of the benchmark dataset

Crime Dataset	Attributes	Labels (crime types/category)
San Francisco	Dates, Category, Descript, ayOfWeek, PdDistrict, Resolution, Address,X,Y	Larceny/theft, Other offenses, Non-criminal, Assault, Drug/narcotic, Missing person, Suspicious occ, Vehicle Theft, Prostitution, Secondary codes, Trespass, Weapon laws, Vandalism, Warrants, Burglary, Forgery/counterfeiting, Robbery, Fraud, Driving under the influence, Sex offenses forcible, Kidnapping, Stolen property, Recovered vehicle, Drunkenness, Disorderly conduct
Chicago	Case#, date of occurrence, block, IUCR, primary description, secondary description, location description, arrest, domestic, beat, ward, FBI cd, x coordinate, y coordinate, latitude, longitude, location	Theft, Battery, Criminal damage, Assault, Other offense, Deceptive practice, Motor vehicle theft, Weapons violation, Robbery, Burglary, Narcotics, Criminal trespass, Homicide, Others, Offenses involving children, Sex offense, Criminal sexual assault, Public peace violation
Vancouver	Type, Year, Month, Day, Hour, Minute,Hundred_block, Neighbourhood, X, Y	Theft from Vehicle, Mischief, Break and Enter Residential/Other, Other Theft, Offence Against a Person, Theft of Vehicle, Break and Enter Commercial, Theft of Bicycle, Vehicle Collision or Pedestrian Struck (with Injury), Vehicle Collision or Pedestrian Struck (with Fatality), Homicide

Boston	Incident_number,Offense_code,Offense_code_group, Offense_description,District,Reporting_area,Shooting, Occurred_on_date, Year, Month, Day_of_week Hour, Ucr_part, Street, Lat, Long, Location	Motor Vehicle Accident Response, Larceny, Medical Assistance, Investigate Person, Other, Vandalism, Simple AssaultVerbal Disputes, OTHERS Towed, Drug Violation, Investigate Property, Larceny From Motor Vehicle Property Lost, Aggravated AssaultFraud, Residential Burglary, Auto Theft, Warrant ArrestsHarassment, Robbery, Property Found, Violations, Missing Person Located, Confidence Games, Missing Person Reported, Police Service Incidents, Fire Related Reports, Restraining Order Violations, Disorderly Conduct, License Violation Counterfeiting, Commercial Burglary, Firearm Violations
Pheonix	Inc number, Occurred on, Occurred to UCR crime category,100 block addr , Zip Premise type, Grid	Larceny-theft, Burglary, Motor vehicle theft Drug offense, Aggravated assault, Robbery Rape, Murder and non-negligent manslaughter, Arson

3) The Chicago crime dataset from Chicago Data Portal (2021-present) has 215969 records (dated June 13, 2022) (Aldossari et al., 2020; Safat et al., 2021; Sri et al., 2020).

4) The Kaggle dataset of Crime in Boston, from June 14, 2015, to September 3, 2018, contains 319073 instances of violent crimes.

5) The Phoenix Crime Dataset is a criminal record file in CSV format given by the City of Ph eonix Open Data that is updated daily and contains crime data from November 2015 until the year 2021 with 427843. Tables 2 (a) and 2 (b) illustrate the descriptions of the benchmark dat asets.

3.2.Data Analysis

Data analysis is transforming, cleaning, and processing primary raw data and pulling out valid, pertinent data that enables the model to make informed decisions. The data analysis approach helps reduce the hazards inherent in decision-making by delivering valuable understandings and statistical figures like charts, images, tables, and graphs (Pradhan et al., 2019).

We have used the classification technique for this work as we have decided to work on a particular target label (Category/Type of crime).

We have uncovered some information from the data analysis step, such as the total number of records or the rows. The San Francisco dataset has 878049 rows and 9 features, Vancouver dataset has 624038 rows and 10 features and Chicago dataset is a live dataset which gets updated every week with 17 features. Pheonix city crime data consists of 427843 rows and 8 features, and Boston dataset contains 319073 rows and 17 features to it. Analysation of the distribution of different types of crime gives a clear picture that Larceny is the crime with the highest frequency in San Francisco. The highest crime rate in Vancouver is theft from vehicles, and the Battery (a kind of theft) seems to be high in Chicago. Motor Vehicle accident response is shown as the highest occurred crime in Boston, whereas the Pheonix recorded Larceny-Theft as the city's frequently occurred crime.

To carry on with more data analysing, we need to pre-process the data, i.e., attribute splitting on the Date feature to split the timestamp into the date, month, year, hour, and minute. The frequency of crime seems to be high during 2013, and as of the month, October has the most elevated rate in San Francisco. In case of time of crime occurrences, at San Francisco most of the offenses likely occurred around six in the evening, and the crime rate touched its peak on Fridays. The highest rate of these crimes happened in the southern police department district of San Francisco.

In Vancouver, the month of August and the year 2003 got recorded as the peak of crime. By midnight, many crimes had occurred in Vancouver, Pheonix and Chicago. Around evening five there are frequent crime in Boston. In Chicago and Boston, the crime frequently happened in September and October, whereas in Pheonix the crime rate has peaked in January and mostly in 2019.

With Python's built-in data analytics mechanisms, we have easily penetrated patterns, correlated information in extensive sets, and got better insights into complementing other critical matrices in estimating performance.

3.3.Data Pre-processing

Data pre-processing is the method of converting plain data into a discernible format. This step is crucial in machine learning because we cannot work with raw data. The data quality should be maintained before applying machine learning algorithms to the data (Alasadi & Bhaya, 2017).

In Python, the libraries are predefined to perform specific tasks. Importing all the essential libraries is one of the mandatory steps in data pre-processing in machine learning. Some of the core Python libraries used for this project are NumPy, Pandas, Matplotlib, Plotly, Folium, Seaborn, SKlearn, Statsmodels, and DESlib (Cruz et al., 2020). All the datasets in .csv file format were imported using Python by the read_csv() function inside the code.

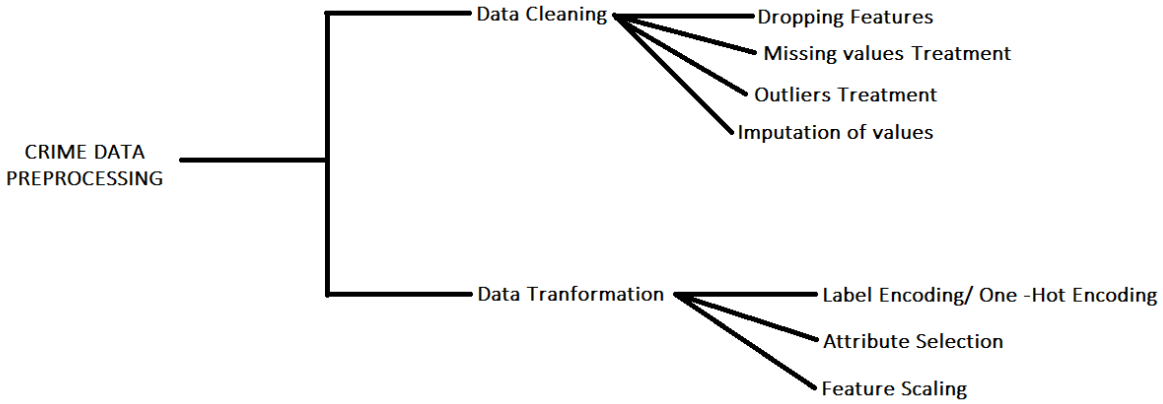


Fig2: Various data pre-processsing activities

3.3.1. Data Cleaning

Data cleaning is improving or extracting wrong, deteriorated, poorly formatted, replicas, or insufficient data in the dataset**.** Data cleansing is a vital component of the prevalent data management method and one of the essential parts of data preparation work that trains data sets for benefit in machine learning. When integrating numerous data origins, there are multiple possibilities for data to be replicated or mislabeled (Calabrese, 2018; J. Wang et al., 2020; X. Wang & Wang, 2020).

Handling the missing values in the dataset:

It is mandatory to conduct detailed analysis steps with good data visualizations besides data pre-processing to understand the data in a better way (Doshi, 2011). There are no null or missing values found in the San Francisco dataset. Chicago, Vancouver, Pheonix, and Bosten datasets had null values in features illustrated in fig 3 (a),(b),(c), and (d).

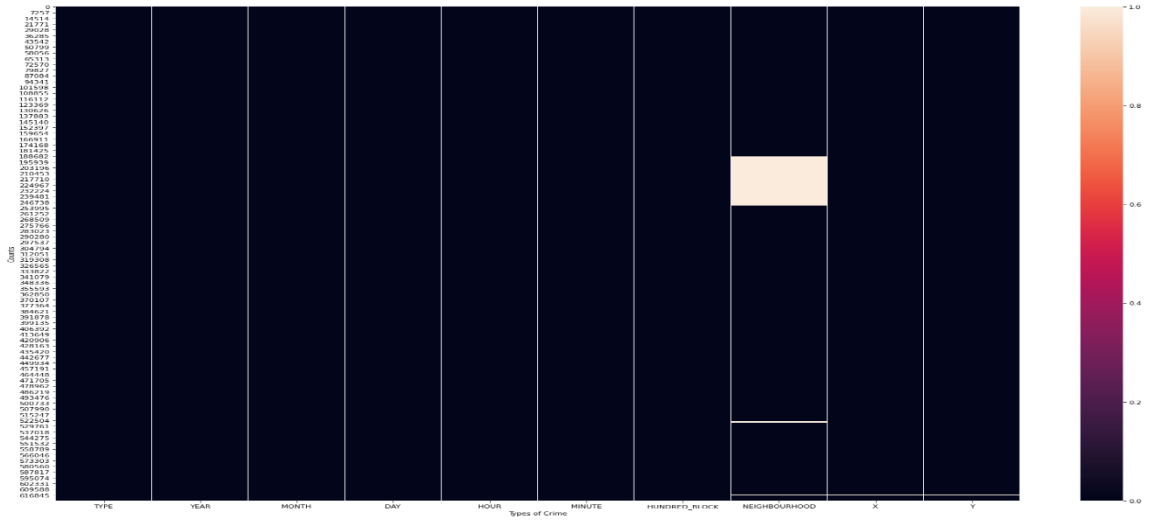


Fig 3: (a) Missing values in Dataset (Vancouver)

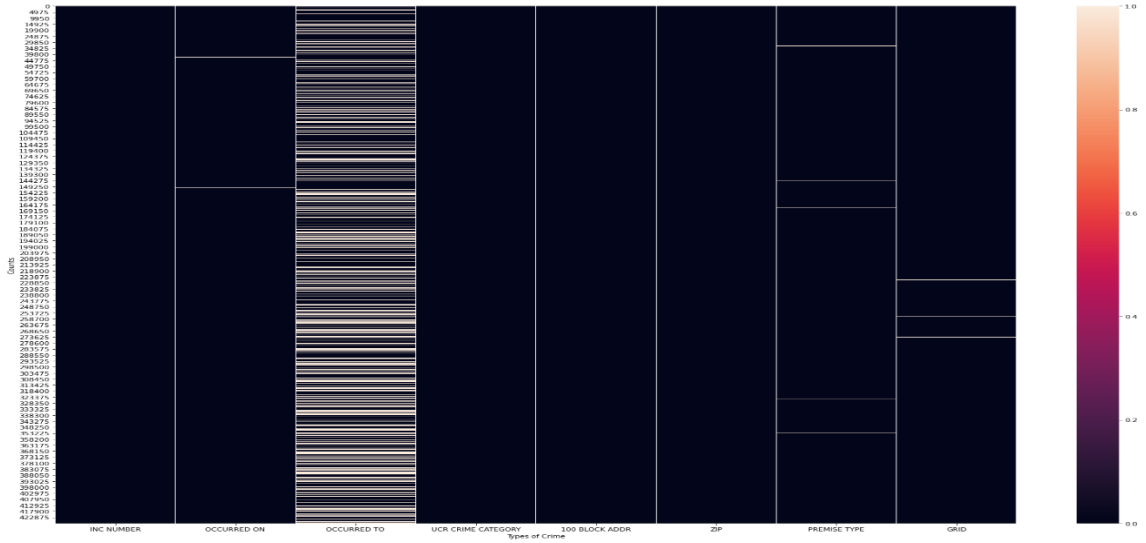


Fig 3: (b) Missing values in Dataset (Pheonix)

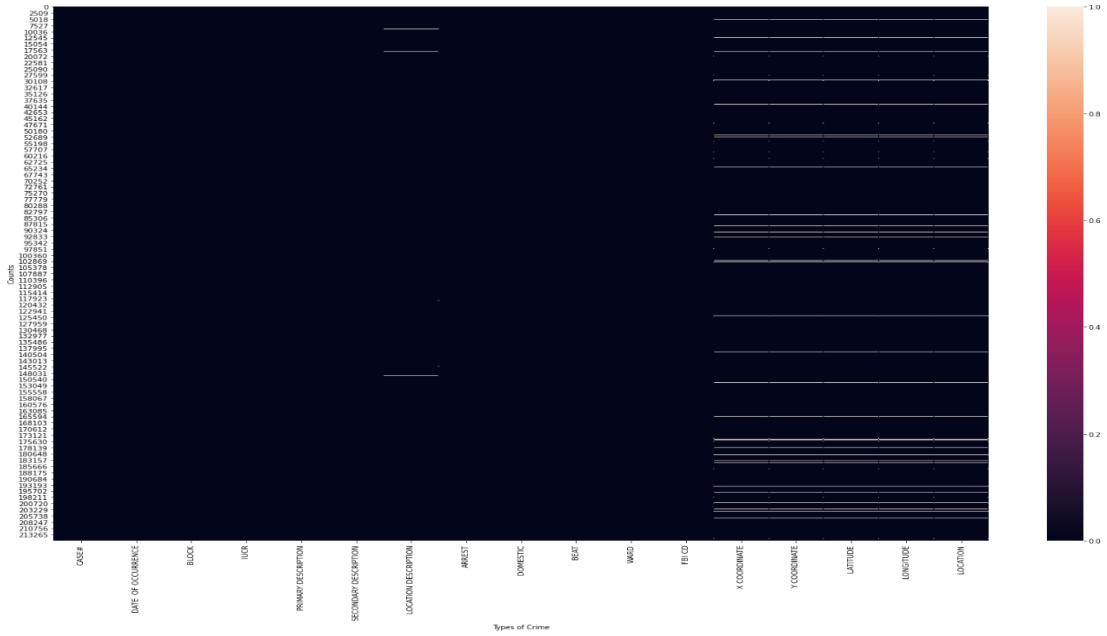


Fig 3: (c) Missing values in Dataset (Chicago)

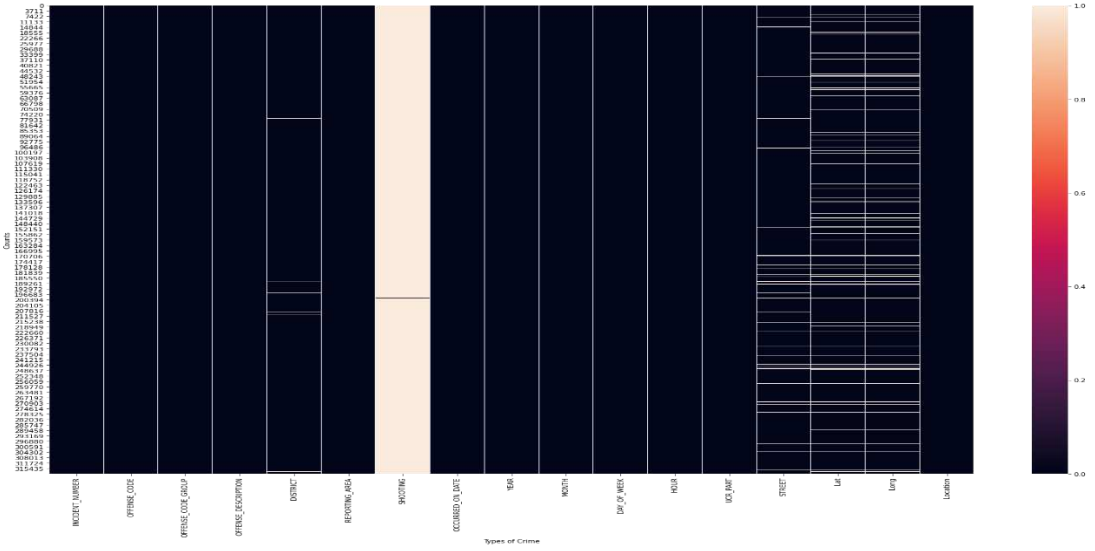


Fig 3: (a) Missing values in Dataset (Boston)

We used both data imputations using the standard deviation and deleted the unwanted columns and rows with missing data for this dataset. But removing missing values rows gave better accuracy for all the datasets (Johnson et al., 2021; Josse & Husson, 2012; Maddileti et al., 2020).

Encoding the categorical data:

The Machine is trained chiefly with numerical values. Hence it is essential to convert the character data types to numerical data that the device can understand. So, we have used label encoding[30] and one-hot encoding with dummy variables to convert the char data typed feature to categorical variables and nominal variables—especially the encoding of target labels to categorical variables and then to numerical values by factorizing it (Chiou et al., 2014; *Machine Learning and the Internet of Medical Things in Healthcare - Google Books*, n.d.; Vink et al., 2020).Fig 4. Encoding shows how the features are converted into label encoding and one-hot encoding.

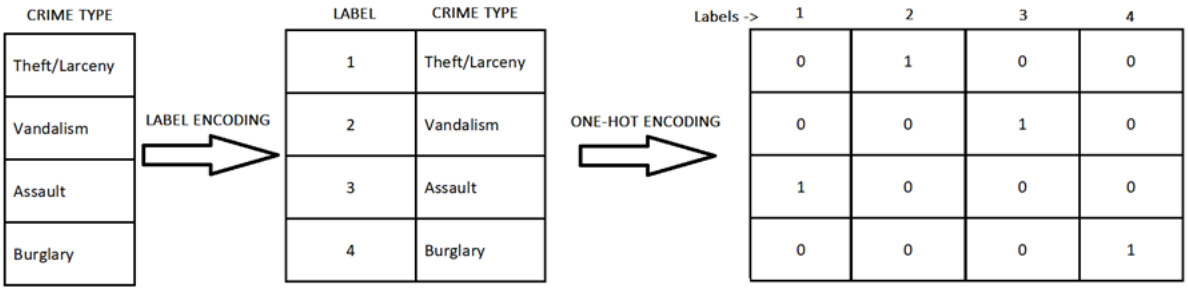


Fig 4: Example for encoding the categorical data

The problem that arises during the conversion of categorical value is that the variable may show the multicollinear property, that is, a robust correlation of independent variables to each other. Multicollinearity is a notion in statistics where multiple variables in a model are associated with each other, i.e., correlation**.** When the correlation coefficient is negative or positive, the variable is collinear in nature (Ghorbanzadeh et al., 2020).This led to the consequence of less dependable statistical hypotheses. So, to check that, we have used the Chi-square test and VIF.

A random variable (χ) follows chi-square distribution as a sum of squared standard normal variables (Cuneen & Tobar, 2021). The Chi-square test is utilized to test the Correlation between two variables. Let us take the data of two variables, with the number of observations as 'o' and expected observation as 'e.' Chi-Square estimates how e and o deviate from each other (Franke et al., 2012).

$$x^2 = \sum \frac{(O_i - E_i)^2}{E_i} \qquad \text{equation (1)}$$

Where O_i is the observed value, and E_i is the expected value

VIF score is a score given to each independent variable to show how other independent variables explain the variables (Budilaksana et al., 2021).

R^2 represents VIF. The higher the R^2 value, the higher variable is correlated with the other variables. VIF is statistically denoted as:

$$VIF = \frac{1}{1-R^2}$$ equation (2)

$$= \frac{1}{Tolerence}$$

Outliers Detection and Removal:

Outlier is a data entity that varies enormously from the remnant of the data entities and acts differently**.** This won't fail the model we build (Singh & Upadhyaya, 2012). We tried plotting a crime awareness street map but found outliers in the latitude and longitude in some of our datasets. To confirm whether the data has outliers, we have applied various visualization techniques like Boxplot, violin plot, and scatter plot on the features, a few of which are shown in Fig 5 Outliers scatterplot, Fig 6 Violinplot1, Fig 7 Violinplot1, Fig 8 Outliers2, and Fig 9 Outliers 2.

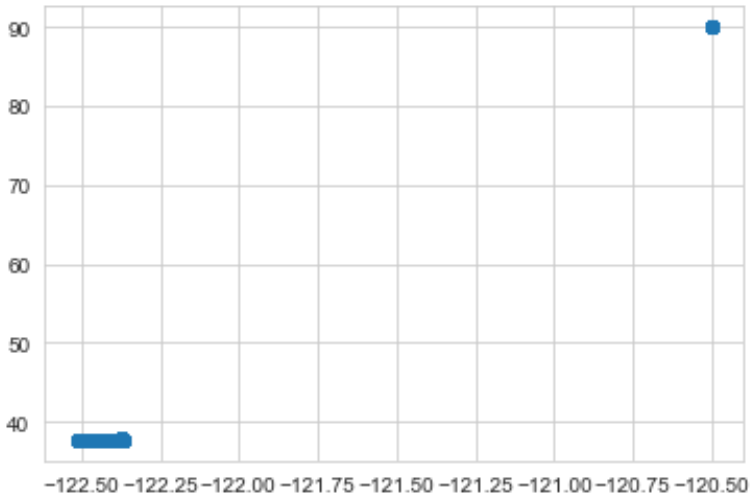


Fig 5: Outliers ScatterPlot

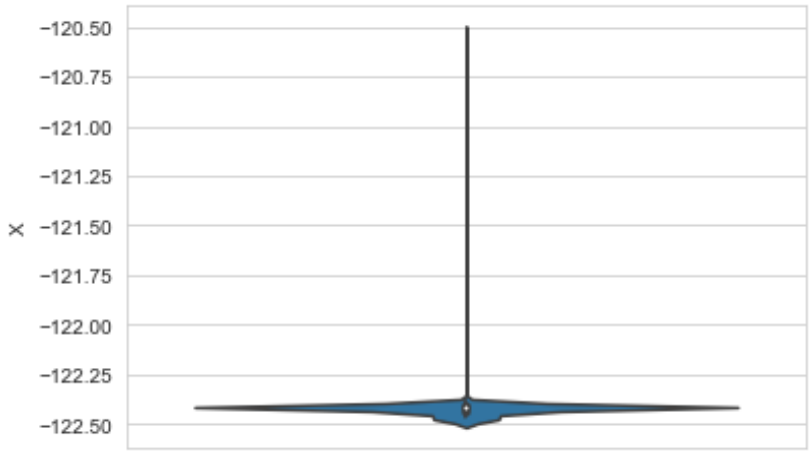


Fig 6: Outliers Violinplot1

Fig 7: Outliers Violinplot1

Fig 8: Outliers 1

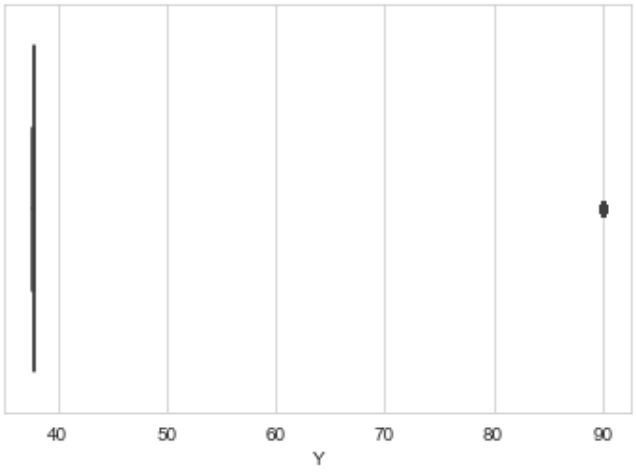


Fig 9: Outliers 2

Outliers are unusual data points that are distant from the other observation. Thus, it needs to be detected and treated before using the Machine learning algorithms as they are sensitive to the distributive range of the values of the features (Hamdi et al., 2022; Singh & Upadhyaya, 2012). So, to see the outliers, various techniques like

- **Z-score:** Z-score or the standard score enables a more straightforward way to comprehend whether the given attribute data value is more or less significant than the mean of that feature and at how much distance it lies from the norm with the standard deviation (Anusha et al., 2019; Bae & Ji, 2019).

  $z = \frac{(x-\mu)}{\sigma}$ equation (3)

  where μ= mean, σ = standard deviation and x = score (Shiffler, 1988).

- **Isolation forest:** Isolation Forest uses the unsupervised decision tree kind of model, which takes a random sub-sampled model and forms a tree with some features (Alsini et al., 2021). The samples with deeper branches are not considered anomalies and are eliminated or eliminated by isolating them. On the other hand, those with less depth show the outlier property. This method gets executed repeatedly to verify the complete dataset. For this work, we used 50 estimators with automatic maximum samples.

- **IQR Outlier detection:** Interquartile range is a measure of statistical distribution, which is the spread of the data (Sunitha et al., 2014). After finding the data median, IQR is the difference between Quartile 1 and Quartile 3. To detect the outlier, multiply IQR by 1.5, add it to Quartile 3 and subtract it from Quartile 1 separately. The points or the data that are greater and smaller than those calculated limits are outliers.

$$IQR = Q_3 - Q_1 \qquad \text{equation (4)}$$

Outliers advance the variability in the data, which lowers the statistical control. Therefore, excluding outliers would make our results statistically influential (S. Xu et al., 2015). After confirming the outliers and finding out where those are, we need to treat the anomalies. We adopted two methods to treat them (Chiou et al., 2014). The first one is to remove those rows with the outliers altogether because the outlier's percentage is less. The second method is data imputation by random sampling from a normal distribution with the standard deviation from the mean (Srivastava et al., 2022).

Feature scaling

Feature scaling is a step of Data Pre-Processing applied to independent variables or features of data. This is a part of data transformation which is utilized to transform the primary raw data into an appropriate format that efficiently relieves data mining and regains strategic data (Thara et al., 2019)**.** The columns' scaling helps standardize the data within a specific range. Sometimes, it also helps in racing up the estimations in an algorithm (Wan, 2019). Feature scaling is applied for the dataset to remove points outside the bounding box, points with wrong coordinates, and drop duplicate rows. Some of the most common feature scaling methods we have used include:

- Standardization: Feature standardization drives the values of individual features in the data to have zero mean and zero-unit variance (Liu, 2020). The known computation

technique is to decide the mean distribution and each feature's standard deviation and estimate the latest data point by the given formula (Rockett, 2022):

$$Standardization = \frac{X-\mu}{\sigma} \quad \text{equation (5)}$$

X = Observation , μ= mean, σ = standard deviation

- Normalization: Feature Normalization is the most straightforward approach and consists of rescaling the spectrum of features to scale the degree in [0, 1]. Hence, it is also known as min-max scaling or min-max normalization (Liu, 2020). The general formula for normalization is shown as

$$X_{new} = \frac{X - X_{min}}{X_{max} - X_{min}} \quad \text{equation (6)}$$

Here, max(x) and Min(x) are the feature's highest and lowest values, respectively. We can even do the normalization method over distinct gaps, e.g., selecting to retain the variable spreading in any [a, b] interval, where a and b are real numbers. The formula to rescale a range between an arbitrary set of values [a, b]:

$$x_{normalized} = (b-a)\frac{x - min(x)}{max(x) - min(x)} + a \quad \text{equation (7)}$$

3.3.2. Feature selection

Feature selection is the method of decreasing the number of infusion variables when creating a predictive model (Khaire & Dhanalakshmi, 2022; Li et al., 2017) essential because, with fewer features, the models that are yet to be built become additionally interpretable, and training of the model becomes faster, which in turn reduces the space required by the model (Li et al., 2017). To improve the predictive accuracy of our test data, we used feature selection algorithms in our dataset by choosing more relevant features and leaving the irrelevant and replicated features (Pilnenskiy & Smetannikov, 2020). For our datasets, we have applied the following:

- **Filter Methods:** Features are fixed based on their scores in various statistical ordeals for their Correlation with the output variable. Our work uses Pearson's Correlation to quantify linear dependency between two continuous variables. Pearson's Correlation is a statistic that estimates the linear Correlation between an input A feature and the outcome B feature. Its value ranges from a negative to a positive one, where one indicates an absolute positive correlation between the features, and -1 indicates an entirely negative correlation between the components. Consequently, zero means that there is no linear correlation. To compute the Pearson correlation coefficient, carry the covariance of the input feature A and result in feature Y and divide it by the product of the two features' standard deviation. The other method is Chi-Square, a statistical examination applied to the groups of categorical attributes to estimate the likelihood of Correlation or association between them utilizing their frequency distribution. The critical point to recall in the filter method is that it won't remove multicollinearity. Thus, we must clear those before training the model (Bommert et al., 2020).
- **Wrapper Method:** In Wrapper methods, we attempt to utilize a subset of features and prepare a model using them. We choose to add or remove elements from our subset based on the assumptions drawn from the earlier model. These wrapper techniques are usually computationally costly (Kohavi & John, 1997). One such method is backward elimination, which first considers all the features. Then the least significant component at each iteration is extracted to improve the model's performance. This process iterates until observation of no progress on removing features (Ali, 2017).

Fig 10 Feature Selection shows how the selection happens in the filter and wrapper method with a clear understanding.

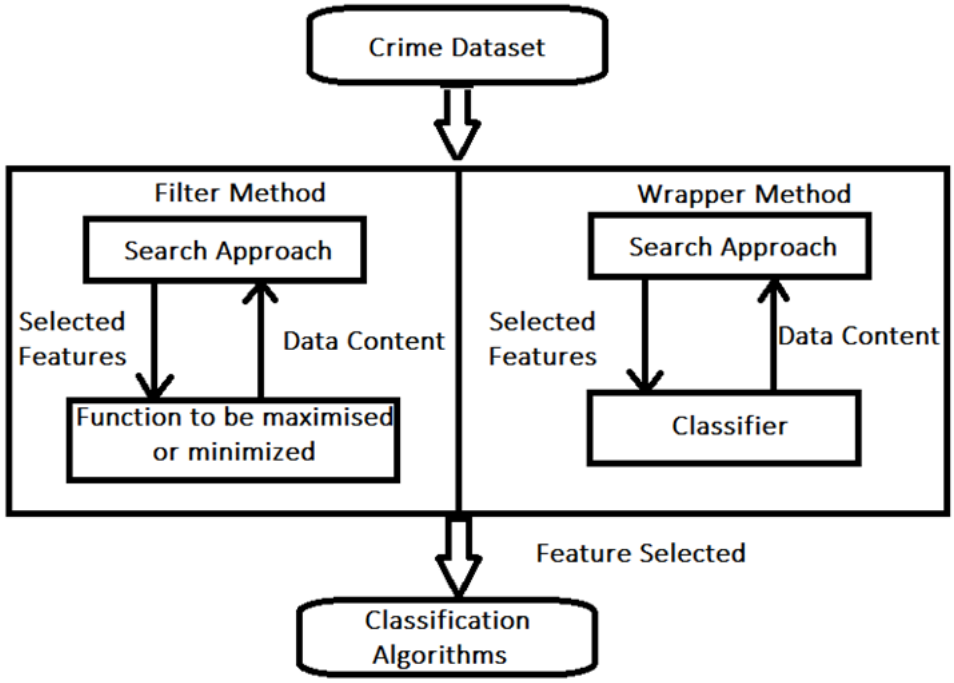


Fig 10: Various Feature Selection techniques

3.4.Classification Algorithm

The supervised classification machine learning model is built by understanding and generalizing the training data, then applying that earned knowledge to new data it has never witnessed before to predict the output label and fulfil its goal (Abouelnaga, 2016; Hajela et al., 2021; Larios, 2016). Once the dataset is tidied, we can train the model to learn from the data we've equipped by applying a range of techniques and algorithms. Classification is an approach for deciding which class the dependent belongs to, based on one or more independent variables that predict discrete answers (Abouelnaga, 2016; Hale & Liu, n.d.; Padmanabhan & D, 2019).

Decision tree Classifier

Decision trees can be employed for both regression and classification. The name itself indicates that it utilizes a sorted tree structure that flows from top to bottom like a flowchart with a sequence of feature-based separations in the classification technique. It begins

splitting from the root node and concludes with a decision made by leaves (Aldossari et al., 2020; Wibowo & Oesman, 2020).

Individually every node in the tree serves as an ordeal case for some particular. Each edge is plunging from that node coordinates to one of the probable solutions to the test case. This process happens repetitively and is replicated for every subtree embedded at the latest nodes. It finds the most reasonable attribute in the dataset by utilizing the Attribute Selection Measure (ASM) (Song & Lu, 2015). The execution of ASM using the Information gain technique measures shifts in entropy post the dataset segmentation based on an attribute. Entropy is a metric to calculate the contaminant in a given feature. It defines randomness in data.

Logistic Regression

Logistic regression is used to express the data and the relationship between one dependent variable and one or more independent variables. The independent variables are of any type, such as interval, nominal, or ordinal. The logistic regression model building data should have any multicollinearity; the independent variables must be distinct. This algorithm works well with large sample size. Equation 8 defined the logistic function (Connelly, 2020).

$$g(z) = \frac{1}{1+e^{-x}} \qquad \text{equation (8)}$$

This algorithm models the likelihood of the default class. Logistic regression uses the logistic function or the sigmoid function, whose value lies between zero and one.

The Logistic Regression classifier is parametrized by a weight matrix and a bias vector w,b the projecting data points onto a set of hyper-planes used to classify the data point and the distance to which is used to determine a class membership probability.

$$P(Y = i|x, W, b) = \frac{e^{W_{ix} + bi}}{\sum_j e^{W_{jx} + b_j}} \qquad \text{equation (9)}$$

In equation 9 the corresponding to each class y_i logistic classifier is characterized by a set of parameters W_i ,b_i (Connelly, 2020; Rotarou, 2018).

K-Nearest Neighbours

K-NN is a non-parametric and lazy learning algorithm that takes the likeness between the new data and available data and puts the new data into the category that is most similar to the general categories. It never learns from the training set instantly; instead, it keeps the dataset, and during the time of classification, it perpetuates an action on the dataset (Zhang et al., 2017).

The hyperparameter K's value is fine-tuned for accurate classification/prediction. Since we don't have a precise method for choosing the proper value of K, we tried to test the model's accuracy for different K values. Selecting an odd value of K is picked because such a state of equality between the two classes would never occur here. The value of K is determined as odd because one out of the two groups would always be in the majority. In contrast, we are dealing with an imbalanced data set. [30]The value of K is taken such that K produces the most satisfactory accuracy for both the training and testing data selected. For all the datasets taken, we have applied KNN with K values of 1, 5, 10, 15, 20, and 25 (Darapureddy et al., 2019; Janardhan & Kumaresh, 2022; More et al., 2021).

Naive Baye's

Naive Bayes is a machine learning model for large volumes of data with millions of data records. It is fast and gives excellent results and an uncomplicated classification algorithm (Jangra et al., 2019). It is a classifier that uses conditional probability based on the Bayes theorem. Prediction of associateship chances is constructed for every class, such as the probability of data points associated with a particular category (Aldossari et al., 2020; Berrar, 2018).

Equation 10 is described the formulas for the Bayes theorem.

$$P(A|B) = \frac{P(A \cap B)}{P(B)} = \frac{P(A) \cdot P(B|A)}{P(B)} \qquad \text{equation (10)}$$

P(A) = The probability of A occurring
P(B) = The probability of B occurring
P(A|B) = The probability of A given B
P(B|A) = The probability of B given A
P(A∩B) = The probability of both A and B occurring

The class that attains the highest probability is decided as the most suitable class, known as Maximum A Posteriori (MAP).

$$P(\theta|X) = P(X|\theta) * P(\theta) \qquad \text{equation (11)}$$

We have chosen Gaussian Naïve Bayes for Chicago Dataset and Boston Dataset because the characteristic values are persistent. Because the above two datasets have continuous features, we have utilized Gaussian Naïve Bayes. The assumption is created such that the values associated with each class are distributed according to Normal Distribution. The formula of Gaussian distribution is as follows (Ismail et al., 2020):

$$P(x_i \mid y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} exp\left(-\frac{(x_i - \mu_y)^2}{2\sigma_y^2}\right) \qquad \text{equation (12)}$$

For the Vancouver, we have used Categorical Naïve Bayes, and, for both the Pheonix and San Francisco datasets, Bernoulli Naïve Bayes are used. Formula 1 represents the working of Categorical Naïve Bayes, and formula 2 means the working of Bernoulli Naïve Bayes .

$$\text{P(class | data)} = \frac{P(data \mid class) * P(class)}{P(data)} \qquad \text{equation (13)}$$

$$p(x) = P[X = x] = \begin{cases} q = 1 - p, & x = 0 \\ p, & x = 1 \end{cases}$$ equation (14)

Random forest

A random forest algorithm is a method used to solve classification problems with the help of an ensemble of numerous decision trees as it eradicates the limitations of overfitting issues in the decision tree algorithm (Khatun et al., 2021). This algorithm demonstrates the result based on the mean of the results produced by various decision trees. The number of trees is directly proportional to the precision of the outcome (Aljamaan & Alazba, 2020; Maddileti et al., 2020; Vassallo et al., 2021).

At first need to choose the number of decision trees to be built. And then repetitively set random data points from the training set and build the decision trees associated with those subsets. For current data points, find the predictions of the individual decision tree, and set the new data points to the class that wins the plurality votes (Alves et al., 2018).

$$Gini = 1 - \sum_{i=1}^{c} (p_i)^2$$ equation (15)

$$Entropy = \sum_{i=1}^{c} -p_{i} * log_2 (p_i)^2$$ equation (16)

Gini index and entropy is the measure for estimating the information gain. Decision tree algorithms utilize data acquired to divide a node. Both Gini and entropy estimate the impurity of a node**.** A node with multiple classes is impure, while the node with solely one type is pure.

Splitting the dataset

The given data is split as train data and test data using the train-test split, which is a process for assessing the performance of all the machine learning algorithms, which takes up the whole dataset and separates it into two parts (Brownlee, 2020). The model is fit by the first subset,

the training dataset. The second subset, called the test dataset, doesn't train the model; instead, this dataset is given as the input element to the model, then predictions are made and compared to the expected values (Rácz et al., 2021). For our work, the test-train split is 80% of the data as train data, and 20% is declared as test data. There are two contending concern situations during the division of a dataset: If training data is less, the parameter calculations will have more significant variance. And If testing data is less, the implementation statistic will have a more substantial conflict.

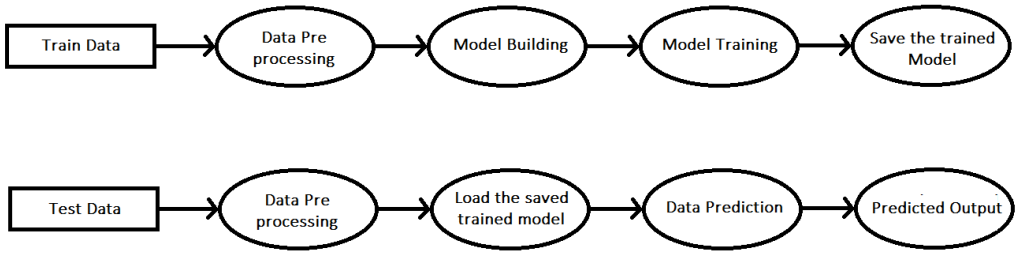


Fig 11: train and test steps.

Cross-Validation Methods

Cross-validation is a process for estimating the machine learning Classifiers by training many different classifiers on the batch of the available data given as the input and assessing those on the complementary data set. For our work, we used cross-validation to find and see if data is overfitting (Zihao Wang et al., 2021).

Overfitting occurs when there is high variance in the dataset instances and has lower friction. Underfitting occurs in the Vice versa case. In simple words model that performed well on training duration but not during the test, the cycle is said to have Overfitting. The model performed worse on test and train sessions than it is under-fitted (Tripathi, 2020). The Y axis in the figure represents the Types or categories of the crime we are classifying, and the X axis shows the predictions made with the test data.

Our data (considering all the datasets) has shown different fits in various algorithms. Hence, we have applied various cross-validation techniques such as Stratified K-Fold, K-Fold, Repeated K-Fold, and Shuffle split.

K-Fold:

K-Fold cross-validation is a method employed to evaluate the mastery of the model on new data. The approach has one parameter (k), which refers to splitting a given data sample into various groups (Nurhopipah & Hasanah, 2020). Since K-Fold ensures that every observation from the initial dataset can occur in the training and test set, it works best when we have limited input data. The value for k is selected so that every train/test batch of data samples is adequately enormous to be statistically suggestive of the more comprehensive dataset. For our implementation, the k value for the k-fold methodology is taken as five, and the whole sample dataset was randomly split into five equally sized disjoint folds, every time giving a varied folding of the whole sample. For every i value, four of the folds were utilized for validating the model, and the rest one-fold was utilized for testing.

The figure K-Fold CV clearly explains how the K-Fold cross-validation works for our dataset. The whole crime dataset is divided into five sub-datasets equally. For every i value, a different fold is used as the test data, and the accuracy is calculated. The accuracy of every fold is summated to give the final predicted accuracy (Tamilarasi & Rani, 2020).

Stratified K-Fold:

Stratified K-Fold is a variation of k-fold, which produces layered folds. In Stratified K-Fold CV, every Individual set holds the same ratio of samples of separate target classes as the whole set. From our observation of the results, we can choose Stratified K-Fold over K-Fold while working with classifiers with highly variated class distributions (Bardhi & Zapirain, 2021).For our implementation of the Stratified k-fold, the k value is chosen as five, and the full sample

dataset with the same ratio of samples of separate target classes was split into five equally sized disjoint folds, every time giving a varied folding of the whole sample. For every i value, four of the folds were utilized for validating the model, and the rest one-fold was used for testing.

Repeated K-Fold:

Repeated K-Fold cross-validation delivers a method to enhance the calculated implementation of a machine learning model. This procedure implies merely reiterating the cross-validation process numerous times and noting the mean development across all folds from all execution. So, this takes a high computation cost to execute this technique, and it fits smaller-sized datasets. Repeated K-Fold is an efficient approach to estimating the forecast fallacy and the precision of a model (Tuson et al., 2021). For our work of the Repeated k-fold, the k value is chosen as five, the whole sample dataset was split into five equally sized disjoint folds, every time giving a varied folding of the whole sample. For every value of i, four of the folds were utilized for validating the model, and the rest one-fold is utilized for testing. This process is repeated 3 times.

Shuffle Split/Monte Carlo:

Shuffle Split/Monte Carlo cross-validation uses the Reprised arbitrary subsampling validation mechanism that divides the dataset haphazardly into training and testing sets (Q. S. Xu & Liang, 2001). The traditional k-fold cross-validation splits the dataset into groups or folds, but shuffle split cross-validation uses the random split method. For every dataset that we have taken, we chose five as the number of splits, and the dataset is split into ten equal parts. Random fifty percent of data is used as a train set; thirty percent is used as a test set, and the rest is left unused. This procedure follows a shuffled pattern.

Ensemble methods

An ensemble method is obtained by blending diverse models to get a more optimal predictive model (Almaw & Kadam, 2018). Instead of just counting on one model and expecting we

earned the right decision at each split, ensemble methods permit us to take a sampling of various models into account, compute which features to utilize or queries to ask at each partition, and make a final predictor based on the aggregated results of the sampled models (Almaw & Kadam, 2018). As of why the ensemble works better than the primary machine learning model due to its performance because it can predict better than the linear model does. The ensemble model's reliability is higher as there is a reduction in the distribution of the model performance and prediction. We have used Boosting, Bagging, Voting, and Stacking in our work. Bagging failed to give more accuracy than the primary machine learning models due to the dataset's highly biased distribution of categories. In voting, we went for hard voting over soft voting as some of the classifiers in our problems can't predict the probabilities. The stack generalization method is used for each dataset separately by stacking three classifiers together in such a way that

- The three best performed linear algorithms
- A combination of the best performers in the linear, ensemble, and neural network
- The three weak performers
- A random combination

Ensemble classifiers generate diverse base classifiers from which a unique classifier is derived, which functions more promising than any associated classifier**.**

There are different types of ensemble classifiers that we have used, including

- **Boosting:** Boosting is a technique that tries to build a robust classifier from the numerous feeble classifiers in a series (Lu & Li, 2020). A model is built at first using the training data, and then the next model is built that attempts to fix the errors present in the model created earlier (Lamari et al., 2020). Until the complete dataset is predicted correctly, this procedure is repeated with a maximum number of models (Almaw & Kadam, 2018). The figure boosting below gives a clear view of how the boosting technique works.

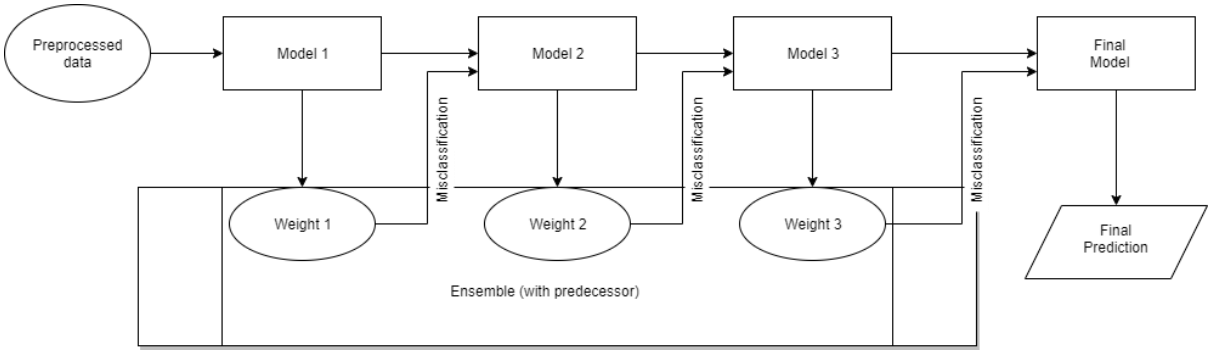


Fig 12: Working process of Boosting technique

The pre-processed data is trained using a particular classifier model named Model1 which gives an output with Weight 1 that has the higher weightage of data points with misclassification by reducing the weight of the correctly predicted data points.

Misclassification here represents the classification of instances with identical features but different output labels. The machine doesn't have a fixed mechanism to work upon this scenario hence it classifies repeatedly as any one of them. These setbacks are fed as an input with the pre-processed new data again into model 2 for training to obtain weight 2. This process continues until we get a reliable final prediction with the last machine learning model.

We have chosen Adaptive Boosting, a.k.a. AdaBoost algorithm, which is adaptive in that successive weak learners are tweaked in the rage of those samples misclassified earlier. The most standard algorithm utilized with AdaBoost is decision trees with a single level, which signifies the Decision trees with solely one split (Aljamaan & Alazba, 2020). These trees are likewise named Decision Stumps.

Gradient boosting falls in a greedy algorithm that penalizes various parts of the algorithm and can quickly improve performance by overfitting a training dataset (Vassallo et al., 2021). It provides a forecast model in forming an ensemble of weaker predictive models, generally

decision trees. In our work, the AdaBoost increases the performance of our machine learning classification algorithm as it is not prone to overfitting.

- **Voting Classifiers:** A voting classifier is an ensemble model that trains various models and forecasts based on aggregating the conclusions of each model are taken. In classification, the outcomes are produced by the preponderance vote of contributing models. The aggregating standards are an integrated conclusion of voting for every model outcome (Atallah & Al-Mousa, 2019). Either of the two methods performs the voting methods. Hard Voting: Voting on the expected output class and Soft Voting on the expected probability of the output class. We have used hard voting since it is purely based on each classifier's class labels and weights (Malikhah et al., 2021).

  $V = mode[M_1(x), M_2(x), \ldots, M_n(x)]$ equation (17)

  For this work we have taken three best performing classifiers because three is an odd number and can produce results without bias. Assuming the ensemble of three classifiers that classify a training sample as follows:

  Model 1 - Class 0

  Model 2 - Class 1

  Model 3 - Class 0

  V =mode[0;1;0]=1 equation (18)

  With the help of majority voting, the classification output of the sample is "Class 0."

- **Stacking:** Stacking or Stack Generalization is one of the best ways of improving the accuracy of the predictive model. In figure Stacking, the training dataset is taken and given to the classifiers (Model 1, Model 2, Model 3) parallelly to get the predicted outcome of new stacked dataset. The newly built dataset is now fed into Level 1 in which the same models are applied once again to get the better trained dataset. This process of training is repeated till we get a better prediction with lesser loss. At the final

level a single model is applied to the dataset that was built in the previous set. The final prediction is then made out of it (Kshatri et al., 2021).

Algorithm 1: STACKING

INPUT:
Cleansed Data set df;
Base Algorithms $X_r(r = 1;2;3)$;
Meta Algorithm X;
Number of Instances n;

PROCESS:
Step 1: Train all the base algorithms X_r for learning with df.
for(r=1 to 3):

$$Lr = X(df_r)$$

StepStep 2:3: EndTake a new dataset df^0 and classify it.
for(N=1 to n):
for(r=1 to 3):

$$C_{ir} = L_r(Xi)$$

Step 4: New Data obtained is, $df^0 = df^0 U((z_{ir}; z_{ir}; :::z_{ir}); y_i)$
Step 5: End
Step 6: Use Meta Classifier to train the new data.

$$L^0 = L(df^0)$$

OUTCOME : return

$$L^0(L1(x);L2(x);L3(x))$$

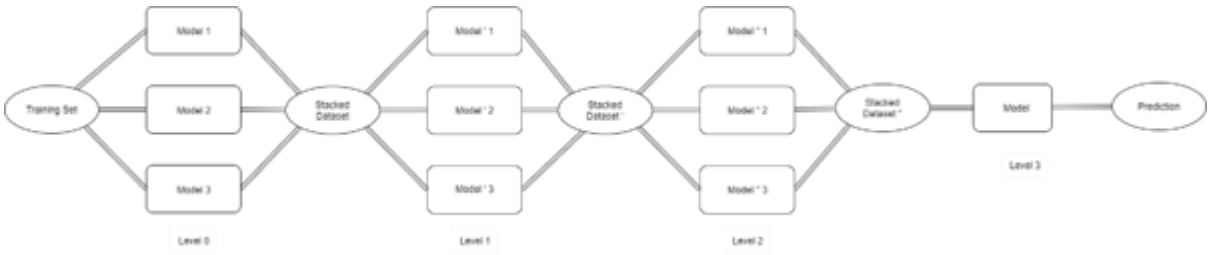


Fig 13: Stacking

We have taken three weakly correlated models every time in our work and performed the stacking with four-folds. So the best of all, worst of all, and combination of linear and ensemble learning models were used to understand and compare how the stacking prediction works (Kshatri et al., 2021; Malikhah et al., 2021; Puurula et al., 2014).

DYNAMIC CLASSIFICATION METHODS

The dynamic ensemble is an ensemble learning approach that automatically selects a subset of ensemble algorithms during the classification period. Many machine learning prototypes were fitted to the training dataset in this process. Then the best model to predict a unique new instance is selected based on the expected components of the sample (Ali, 2017; Britto et al., 2014)**.**

The DES method can be performed using the k-nearest neighbor model to find the instance in the training dataset farthest from the expected new sample. Evaluating all the models in that particular neighborhood pool and using the model with the best performance in the neighborhood predicts the current criteria (classification classifies a collection of data into categories or classes) (Wibowo & Oesman, 2020).

For our dataset we have applied K-Nearest Oracle Union (KNORA-U), k-Nearest Oracle Eliminate (KNORA-E), and DCS using OLA models.

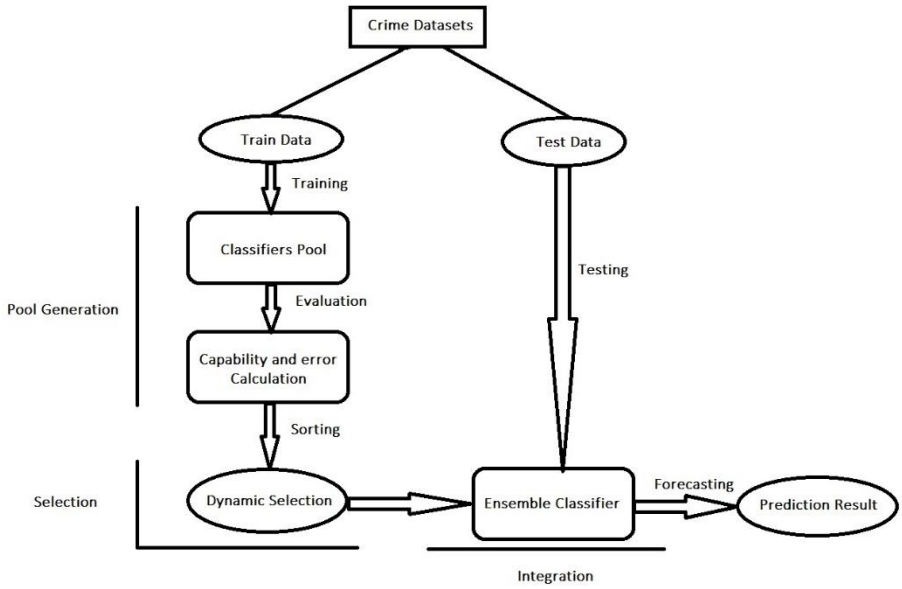


Fig14: Dynamic Ensemble Classifiers

The figure 14 Dynamic ensemble classifiers represent how Dynamic Ensemble selection for classification works for Machine Learning algorithms (Britto et al., 2014). Pre-processed training data is given for training to various models i.e., the pool of classifiers. The test data is used to make the classification. The classified label is then compared with the original label using KNN methodology and its accuracy is noted. This process is repeated foe each classifier in the pool and the model with high accuracy is selected. At the end the final prediction of the test data is made.

DCS With Overall Local Accuracy (OLA)

DCS-LA model is estimated using overall local accuracy on the artificial dataset. For our work, we will use default model hyperparameters, including bagged decision trees, as the collection of classifier models and the neighbor value is seven to select the local neighborhood during classification forecasting (Q. S. Xu & Liang, 2001). For the evaluation of the model, we did repeat stratified k-fold cross-validation with ten folds and three times repetitions. The results were taken using metrics like the mean and standard deviation of the accurateness of the model across all repeats and folds.

Algorithm 2: DYNAMIC CLASSIFIER(OLA)

INPUT:
Cleansed Data sets df1 and df2;
Base Algorithms X;
KNN of size k;

PROCESS:
for t testing samples in df2
do
train t with all Xi
if(predicted label l == original label in all algorithms)
return l;
else

Ψ=k(t);in(df1)

for every Xi do
Calculate OLA
end for
select

X = argmaxi(OLAi)

X is used to classify the data end if
end for

OUTCOME : return Xi

the best classifier for every testing sample t in df2

Rather than discovering the most appropriate classifier, we pick the most appropriate ensemble for the individual sample. The idea of the K-nearest-oracles (KNORA) is identical to the concepts of OLA, LCA, and the A Priori & A Posteriori techniques, considering the neighborhood of test patterns, while it can be differentiated from the rest by the immediate use of its possessions of holding training samples in the region with which to locate the most suitable ensemble for a given sample. For a given test data point, KNORA just discovers its closest K neighbors in the validation set, figures out which classifiers accurately classify those neighbors in that particular set, and utilizes them as the ensemble for classifying the provided pattern in that test set (S. Xu et al., 2015).

KNORA-U

K-Nearest Oracle Union (KNORA-U), the process determines all classifiers that perfectly categorize at most small one sample belonging to the region of competence of the query sample. Every classifier chosen has several votes equal to the number of samples in the region of competence that predicts the accurate label. The votes acquired by all ground classifiers are aggregated to obtain the last ensemble decision (Janardhan & Kumaresh, 2022).

Algorithm 3:KNORA-U

INPUT:
Cleansed Data set df splitted into:
*test data - dft with testing sample
Ts *Validation data - dftr
Pool of Classifiers C;
KNN of size k;

PROCESS:
for Ts in dft
do

while k > 0
do
Find ϕ as the k of Ts in validation set dftr
for every sample Ci in C do
for every classifier Ci in C do
if($C^0C=^i$ classifyC^0 [C_i ϕ i correctly) then
end if
end for
end for
end while
end for
OUTCOME : return C^0;

KNORA-E

The KNORA-E process probes for a provincial Oracle, a ground classifier that accurately classifies all samplings belonging to the area of competence of the test data. All classifiers with ideal performance in the region of competence are selected (local Oracles). Suppose no classifier performs to perfect accuracy. In that case, the size of the competence region is lowered (by dragging the most distant neighbor), and the performance of the classifiers is re-evaluated. The outcomes of the selected ensemble of classifiers are integrated using the majority voting procedure. The entire pool is employed for classification if no base classifier is selected (Oliveira et al., n.d.).

Algorithm 4: KNORA-E

INPUT:
Cleansed Data set df splitted into:
*test data - dft with testing sample Ts
*Validation data - dftr
Pool of Classifiers C;
KNN of size k;

PROCESS:
for Ts in dft do
while k > 0 do
Find ϕ as the k of T_s in validation in set df_{tr}
For each classifier C_i in C do
If (C_i identifies all sample correctly in ϕ) then
C' = C' U C_i;
End if
End for
If (C' == ϕ) then
Reduce k by 1
Else
Break;

End if
End while
If(C' == ϕ) then
Ci with correct classification:
Choose the Ci to construct the ensemble C':
End if
End for
Outcome: return C'

The fig 15 Difference between KNORA-E and KNORA-U, below shows the difference between KNORA-E and KNORA-U. The KNORA-E on the left only employs classifiers that accurately classify all the K-nearest patterns whereas the KNORA-U employs classifiers that precisely classify any of the K-nearest patterns. The test pattern is shown as a pentagon on the feature side, validation data points are shown as circles and the five nearest validation points are darkened. The used classifiers i.e., the intersection of accurate classifiers is shaded on the right side. The figure Difference between KNORA-E and KNORA-U, clearly depicts in KNORA-E, the result is obtained by intersection of the pool of classifiers from the features that are selected whereas in KNORA U, union of the pool of classifiers serves as the result.

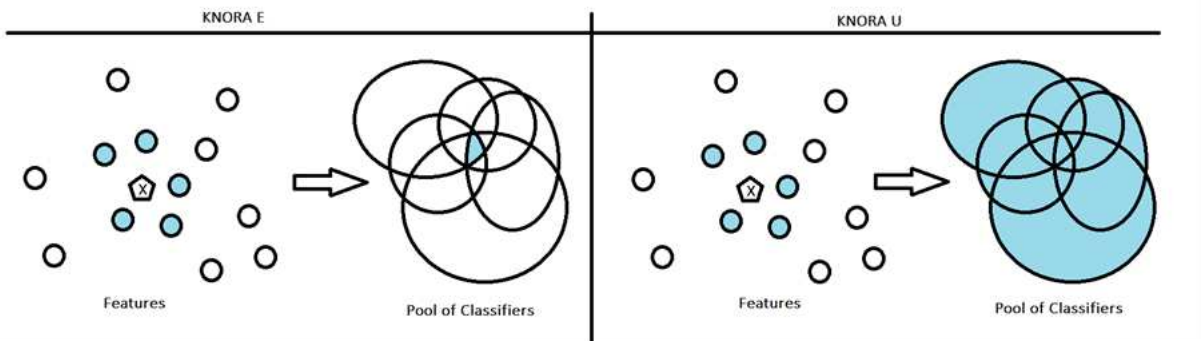


Fig 15: Difference between KNORA-E and KNORA-U

4. Result

Evaluating the performance of an algorithm using metrics is part of all ML pipelines. These indicators indicate whether or not progress is indicated by displaying them in a numerical format. All machine learning models require a measured value to estimate their performance, from basic linear models to complex models (Yu et al., 2022).

Metrics are utilized to observe and estimate the performance of the model (both in the training dataset and test dataset). Every machine learning task is either a Regression or Classification task. Many metrics are available for both problems.

Metrics and loss function are two different terms. Loss functions deliver a skeletal measure of the model's performance, mainly employed to train a machine learning model (with the help of Optimization algorithms. For example, Gradient Descent). The metrics for each algorithm are usually differentiable in the model's parameters (Yu et al., 2022).

Since we went with classification models in this work, we will focus on classification metrics. Classification models will likely have a discrete outcome; thus, we need a metric that compares discrete classes. Classification Metrics assess the model's performance and give a result of how adequate or inadequate the classification is, but each assumes it in a distinguishable way. The primary metrics lie in the confusion matrix to measure performance. The outcome is two or more classes in the tabular format with expected and actual value combinations (Yuki et al., 2019). The confusion matrix has:

True Positive states that the prediction is optimistic and it's true.
True Negative is for the prediction that is negative but true.
False Positive is predicted positive, but it is false.
False Negative is a prediction that is both negative and false.

Some of the important metrics used in our work include:

- **Accuracy:** Accuracy entirely calculates how frequently the classifier's predictions are correct. Accuracy, in short, is defined as the percentage of the number of accurate predictions and the total number of predictions. When a particular model shows an accuracy of a higher rate, like 99 or 100 percent, we might think that prototype we created is functioning very well. But this is not consistently correct and can be deceiving in a few situations, so it is better to check with other metrics.

$$Accuracy = \frac{TP + TN}{TP+TN+FP+FN}$$ equation (19)

[90]

- **Recall:** The ratio of the number of Favorable samples correctly classified as Favorable to the total number of Favorable samples is called the Recall. The *Recall estimates the ML model's capability to catch the positive samples*. The more elevated the Recall, the better positive samples detected.

$$Recall = \frac{TP}{TP+FN}$$ equation (20)[90]

It is autonomous of the number of negative sample types. Additionally, if the model classifies all positive samples exactly as positive, the value of Recall will be 1.

- **Precision:** Precision is the ratio of predicted favorable observations to the total predicted positive observations. Precision denotes the percentage of your relevant outcomes. In different words, it can be expressed as the ratio of precisely classified complimentary samples (which is True Positive) to the total number of classified positive samples (the samples can be either correct or incorrect). Precision permits visualizing the dependability of the machine learning model in classifying the model as positive.

$$Precision = \frac{TP}{TP+FP}$$ equation (21)[90]

The precision of the machine learning model will be high when the Value of True positive in the numerator is greater than the denominator with the sum of true positive and False Positive.

- **F1 Score:** The F1 score incorporates precision and Recall relative to a typical positive class. The F1 score is also known as F-measure, which shows the balance between

precision and Recall. The F1 score analyzes the harmonic mean of precision and Recall. F1 score gets its best value at one and worst at zero.

$$F1 - Score = \frac{TP}{(TP+(1/2)*(FP+FN)}$$

$$= 2 * [\frac{precision * recall}{precidion + recall}]$$ equation (22)[90]

Throughout the result section we have mentioned the datasets Chicago, San Francisco, Vancouver, Pheonix and Boston as Dataset 1, Dataset 2, Dataset 3, Dataset 4, and Dataset 5 respectively.

Result of Basic Classification algorithms:

As stated in the methodology section, five basic machine learning algorithms were used and the results were compared using the above metrics. Table 3 below briefly summarizes the 5 machine learning algorithms used on our five different crime datasets.

Table 3: brief summary about the result of all 5 bench mark datasets

DATASET	MODEL	ALGORITHM(S)	ACCURACY	RECALL	PRECISION	F1 SCORE
Dataset 1	Tree	Decision Tree	99.81%	99.81%	99.81%	99.81%
Dataset 1	Linear Model	Logistic Regression	26.08%	26.08%	13.98%	26.08%
Dataset 1	Neighbours	KNN(1)	67.59%	67.59%	67.39%	67.59%
Dataset 1	Neighbours	KNN(5)	67.43%	67.43%	66.46%	67.4281%
Dataset 1	Neighbours	KNN(10)	66.50%	66.50%	65.42%	66.50%
Dataset 1	Neighbours	KNN(15)	65.37%	65.37%	64.26%	65.37%
Dataset 1	Neighbours	KNN(20)	64.37%	64.37%	63.25%	64.37%
Dataset 1	Neighbours	KNN(25)	63.49%	63.49%	62.42%	63.49%
Dataset 1	Naïve Bayes	Gaussian NB	37.05%	37.05%	32.17%	37.05%
Dataset 1	Ensemble Tree	Random Forest	90.02%	90.02%	90.58%	90.02%

Dataset 2	Tree	Decision Tree	95.33%	95.33%	99.32%	96.55%
Dataset 2	Linear Model	Logistic Regression	81.35%	81.35%	81.35%	81.35%
Dataset 2	Neighbours	KNN(1)	80.72%	80.72%	81.20%	80.73%
Dataset 2	Neighbours	KNN(5)	70.63%	70.63%	91.89%	77.45%
Dataset 2	Neighbours	KNN(10)	79.43%	79.43%	86.27%	91.52%
Dataset 2	Neighbours	KNN(15)	79.68%	79.68%	87.84%	89.32%
Dataset 2	Neighbours	KNN(20)	74.92%	74.92%	89.53%	78.27%
Dataset 2	Neighbours	KNN(25)	76.35%	76.35%	82.68%	76.35%
Dataset 2	Naïve Bayes	Bernoulli NB	90.28%	90.28%	90.28%	90.28%
Dataset 2	Ensemble Tree	Random Forest	84.96%	84.96%	84.96%	84.96%
Dataset 3	Tree	Decision Tree	38.66%	38.66%	38.06%	38.34%
Dataset 3	Linear Model	Logistic Regression	37.26%	37.26%	37.26%	37.26%
Dataset 3	Neighbours	KNN(1)	44.59%	44.59%	44.59%	44.59%
Dataset 3	Neighbours	KNN(5)	48.85%	48.85%	53.51%	50.72%
Dataset 3	Neighbours	KNN(10)	49.92%	49.92%	58.11%	52.98%
Dataset 3	Neighbours	KNN(15)	49.83%	49.83%	60.60%	53.67%
Dataset 3	Neighbours	KNN(20)	49.58%	49.58%	62.47%	54.03%
Dataset 3	Neighbours	KNN(25)	49.38%	49.38%	64.01%	54.32%
Dataset 3	Naïve Bayes	Categorical NB	47.70%	47.70%	47.70%	47.70%
Dataset 3	Ensemble Tree	Random Forest	37.26%	37.26%	37.26%	37.26%
Dataset 4	Tree	Decision Tree	46.14%	46.14%	46.81%	46.14%
Dataset 4	Linear Model	Logistic Regression	51.73%	51.73%	26.76%	51.73%
Dataset 4	Neighbours	KNN(1)	37.03%	37.03%	36.75%	37.03%
Dataset 4	Neighbours	KNN(5)	46.33%	46.33%	35.71%	46.33%
Dataset 4	Neighbours	KNN(10)	48.99%	48.99%	35.45%	48.99%

Dataset 4	Neighbours	KNN(15)	50.14%	50.14%	35.41%	50.14%
Dataset 4	Neighbours	KNN(20)	50.64%	50.64%	35.36%	50.64%
Dataset 4	Neighbours	KNN(25)	50.92%	50.92%	35.04%	50.92%
Dataset 4	Naïve Bayes	Bernoulli NB	52.81%	52.81%	33.95%	52.81%
Dataset 4	Ensemble Tree	Random Forest	55.59%	55.59%	49.47%	55.59%
Dataset 5	Tree	Decision Tree	49.26%	49.26%	49.34%	49.26%
Dataset 5	Linear Model	Logistic Regression	45.16%	45.16%	45.16%	45.16%
Dataset 5	Neighbours	KNN(1)	42.69%	42.69%	42.53%	42.69%
Dataset 5	Neighbours	KNN(5)	49.16%	49.16%	43.88%	49.16%
Dataset 5	Neighbours	KNN(10)	50.72%	50.72%	43.15%	50.72%
Dataset 5	Neighbours	KNN(15)	52.52%	52.52%	44.52%	52.52%
Dataset 5	Neighbours	KNN(20)	53.33%	53.33%	44.95%	53.33%
Dataset 5	Neighbours	KNN(25)	53.77%	53.77%	45.22%	53.77%
Dataset 5	Naïve Bayes	Gaussian NB	16.38%	16.38%	16.28%	16.38%
Dataset 5	Ensemble Tree	Random Forest	77.54%	77.54%	82.15%	77.54%

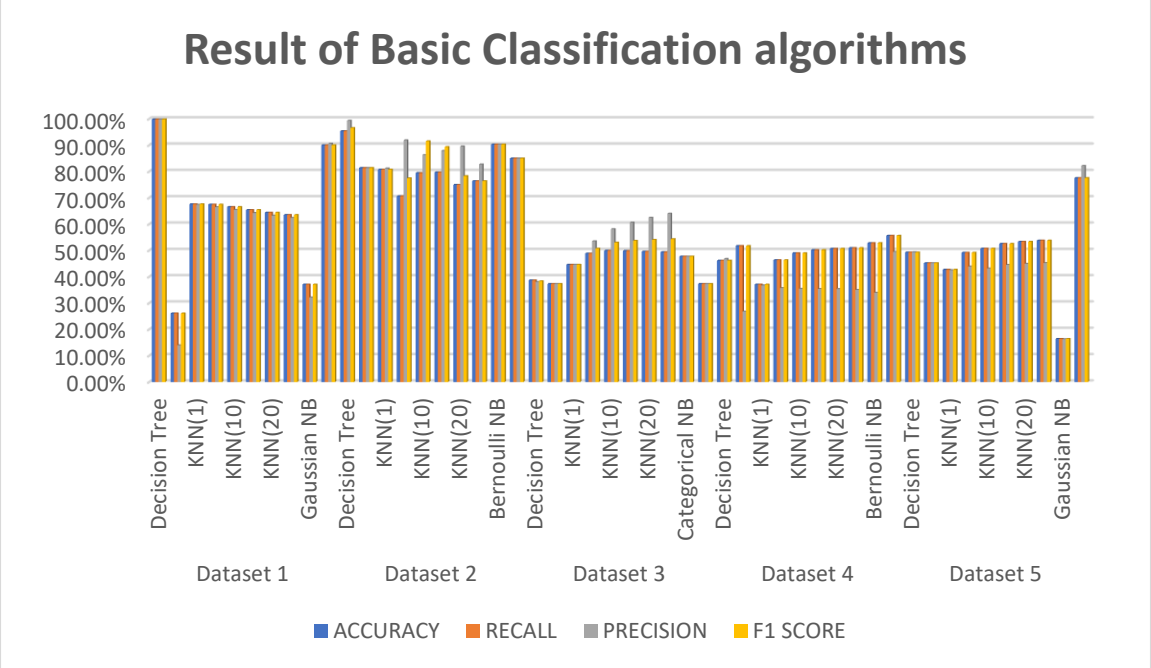


Fig 16: Visualization of the result of basic classification algorithms

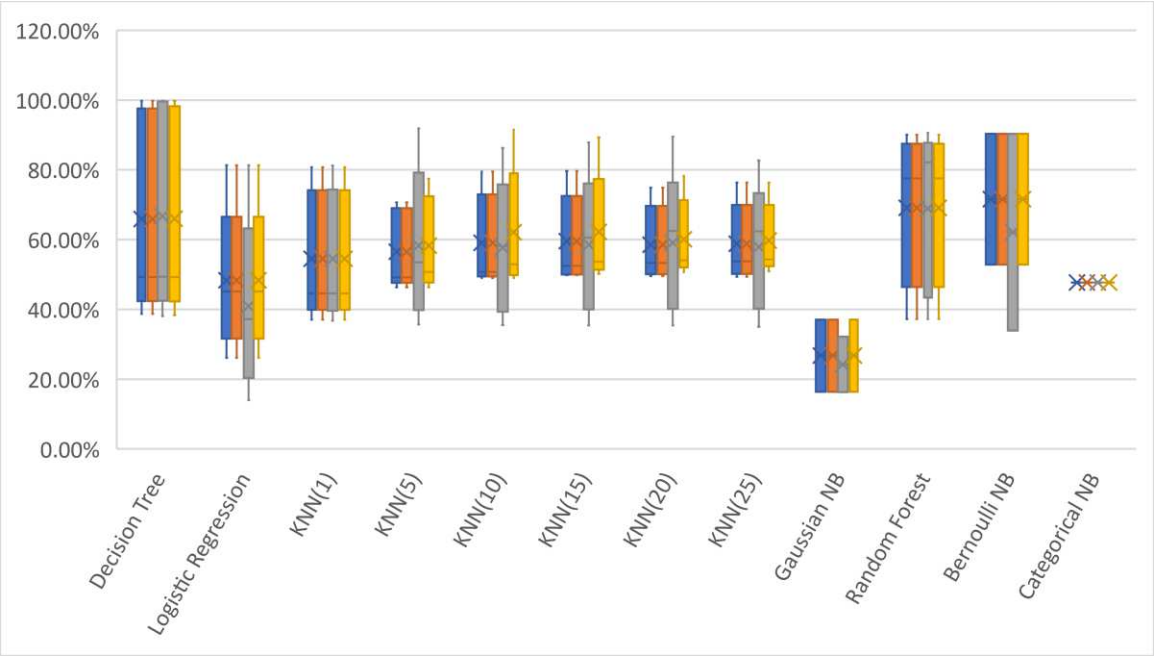


Fig 17: Visualization of the result of basic classification algorithms

Performance results post Cross-validation:

Further, the hyper parameter adjustments were made with different cross-validation methods. The following results are obtained, which are shown in table 4, table 5, table 6, and table 7.

Performance results with K-Fold CV:

K-fold Cross-validation helps in reducing overfitting. As for the accuracy improvement, a minimal increase was observed after K-fold Cross-validation.

Table 4: Experimental algorithms with various CVs and their accuracy

DATASET	ALGORITHM	WITHOUT CV (ACCURACY)	K-FOLD	No of splits
Dataset 1	Decision Tree	99.7669 %	99.7426 %	5
	Logistic Regression	26.0838 %	36.0868 %	5
	KNN(1)	67.5889 %	66.2719 %	5
	Naïve Bayes	37.0481 %	36.9000 %	5
	Random Forest	90.0209 %	88.7833 %	5
Dataset 2	Decision Tree	99.3317 %	90.1749 %	5
	Logistic Regression	81.3462 %	90.9824 %	5
	KNN(25)	83.1342 %	84.2829 %	5
	Naïve Bayes	90.2841 %	90.2393 %	5
	Random Forest	84.9633 %	83.0355 %	5
Dataset 3	Decision Tree	38.6556 %	38.2793 %	5
	Logistic Regression	37.2604 %	37.1099 %	5
	KNN(1)	48.8449 %	43.9997 %	5
	Random Forest	37.2604 %	46.6102 %	5
Dataset 4	Decision Tree	46.1417%	45.9404%	5
	Logistic Regression	51.7334%	51.8499%	5
	KNN(25)	50.9153%	51.0480%	5
	Naïve Bayes	52.8070%	52.9335%	5
	Random Forest	55.5888%	55.2171%	5
Dataset 5	Decision Tree	49.2648%	49.2351%	5
	Logistic Regression	45.1613%	44.6889%	5
	KNN(25)	53.7710%	52.5882%	5
	Naïve Bayes	16.3760%	16.2652%	5
	Random Forest	77.5409%	76.8182%	5

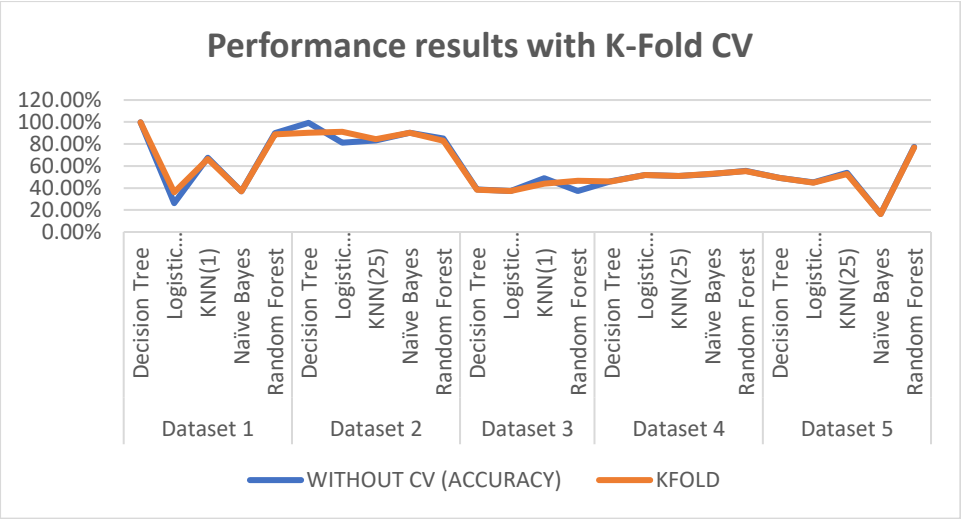


Fig 18: Visualization of the result of different classification algorithms with K-fold CV

Performance results with Stratified K-Fold CV:

Stratified K-Fold Cross Validation returns the stratified sampling folds and is the variant of K-fold. To overcome the random sampling issue, this Stratified k-fold is used, and we have observed mixed results, i.e., in some cases, the accuracy has increased, but in a few, it has depreciated.

Table 5: Experimental result of various algorithms with SKCV

DATASET	ALGORITHM	WITHOUT CV (ACCURACY)	STRATIFIED KFOLD	No of splits
Dataset 1	Decision Tree	99.7669 %	99.7468 %	5
	Logistic Regression	26.0838 %	36.1322 %	5
	KNN(1)	67.5889 %	66.2979 %	5
	Naïve Bayes	37.0481 %	36.9109 %	5
	Random Forest	90.0209 %	89.0557 %	5
Dataset 2	Decision Tree	99.3317 %	90.2125 %	5
	Logistic Regression	81.3462 %	90.9972 %	5
	KNN(25)	83.1342 %	84.3244 %	5
	Naïve Bayes	90.2841 %	90.1516 %	5
	Random Forest	84.9633 %	82.9972 %	5
Dataset 3	Decision Tree	38.6556 %	38.4288 %	5
	Logistic Regression	37.2604 %	37.1099 %	5
	KNN(1)	48.8449 %	43.9988 %	5
	Random Forest	37.2604 %	46.5523 %	5
Dataset 4	Decision Tree	46.1417%	45.8936%	5
	Logistic Regression	51.7334%	51.8499%	5
	KNN(25)	50.9153%	51.0436%	5
	Naïve Bayes	52.8070%	52.9492%	5
	Random Forest	55.5888%	55.1983%	5
Dataset 5	Decision Tree	49.2648%	49.6568%	5
	Logistic Regression	45.1613%	45.0441%	5
	KNN(25)	53.7710%	52.5255%	5
	Naïve Bayes	16.3760%	16.2053%	5
	Random Forest	77.5409%	77.0353%	5

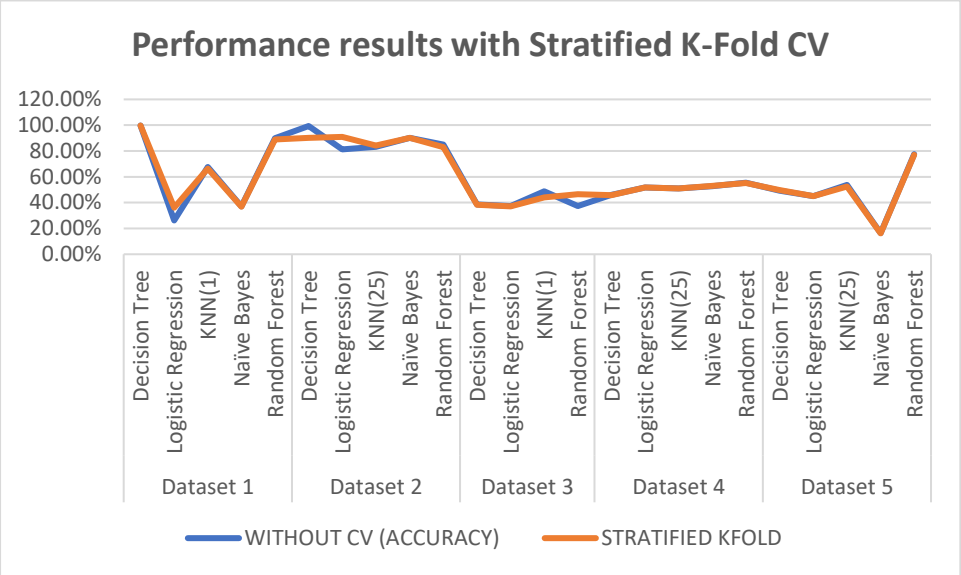


Fig 19: Visualization of the result of different classification algorithms with S K-fold CV

Performance results with Repeated K-Fold CV:

Repeatedly applying the K-folds, which select different folds per each repeat, on the datasets has shown similar kinds of results as other k-fold.

Table 6: Experimental result of various ML algorithms with RK-fold

DATASET	ALGORITHM	WITHOUT CV (ACCURACY)	REPEATED KFOLD	No of splits	No of Repeats
Dataset 1	Decision Tree	99.7669 %	99.7436 %	5	3
	Logistic Regression	26.0838 %	36.1051 %	5	3
	KNN(1)	67.5889 %	66.9345 %	5	3
	Naïve Bayes	37.0481 %	36.8922 %	5	3
	Random Forest	90.0209 %	89.2344 %	5	3
Dataset 2	Decision Tree	99.3317 %	92.9612 %	5	3
	Logistic Regression	81.3462 %	90.9849 %	5	3
	KNN(25)	83.1342 %	85.1103 %	5	3
	Naïve Bayes	90.2841 %	90.1977 %	5	3
	Random Forest	84.9633 %	83.4399 %	5	3
Dataset 3	Decision Tree	38.6556 %	38.7062 %	5	3
	Logistic Regression	37.2604 %	37.1099 %	5	3
	KNN(1)	48.8449 %	44.2014 %	5	3
	Random Forest	37.2604 %	46.7735 %	5	3
Dataset 4	Decision Tree	46.1417%	46.0904%	5	3
	Logistic Regression	51.7334%	51.8500%	5	3
	KNN(25)	50.9153%	51.0975%	5	3

	Naïve Bayes	52.8070%	52.9493%	5	3
	Random Forest	55.5888%	55.8168%	5	3
Dataset 5	Decision Tree	49.2648%	49.9095%	5	3
	Logistic Regression	45.1613%	44.8592%	5	3
	KNN(25)	53.7710%	53.2139%	5	3
	Naïve Bayes	16.3760%	16.1825%	5	3
	Random Forest	77.5409%	77.8618%	5	3

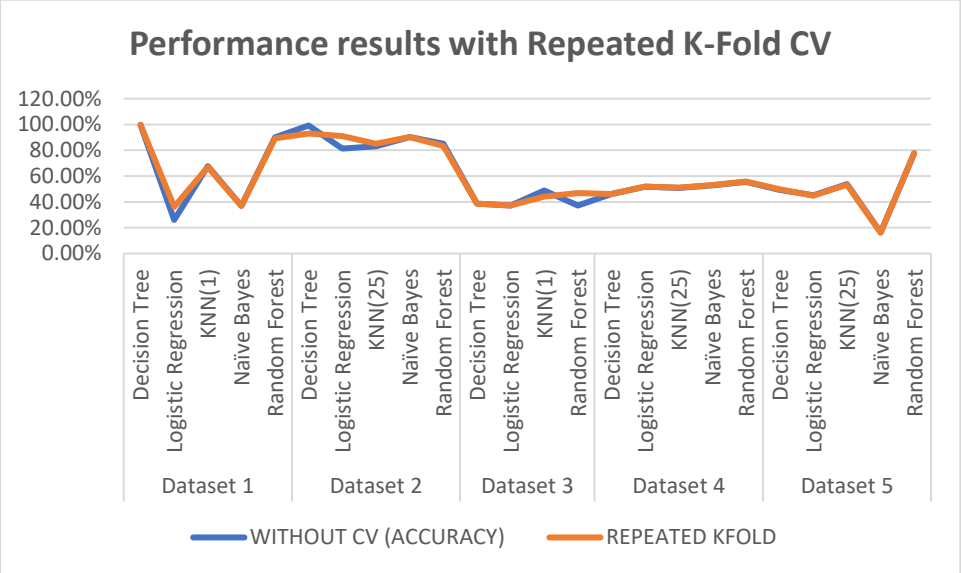


Fig 20: Visualization of the result of different classification algorithms with R K-fold CV

Performance results with shuffle split CV:

Shuffle Split brings out various indices each time. The result is relatively better than other cross-validation methods.

Table 7: Experimental result of various ML algorithms with shuffle split

DATASET	ALGORITHM	WITHOUT CV (ACCURACY)	SHUFFLE SPLIT	Train size	Test size	No of splits
Dataset 1	Decision Tree	99.7669 %	99.5962 %	50 %	30 %	10
	Logistic Regression	26.0838 %	35.9979 %	50 %	30 %	10
	KNN(1)	67.5889 %	63.4134 %	50 %	30 %	10
	Naïve Bayes	37.0481 %	36.9749 %	50 %	30 %	10

	Random Forest	90.0209 %	87.9633 %	50 %	30 %	10
Dataset 2	Decision Tree	99.3317 %	81.7428 %	50 %	30 %	10
	Logistic Regression	81.3462 %	90.9686 %	50 %	30 %	10
	KNN(25)	83.1342 %	82.4188 %	50 %	30 %	10
	Naïve Bayes	90.2841 %	90.3474 %	50 %	30 %	10
	Random Forest	84.9633 %	81.6109 %	50 %	30 %	10
Dataset 3	Decision Tree	38.6556 %	37.3134 %	50 %	30 %	10
	Logistic Regression	37.2604 %	37.1117 %	50 %	30 %	10
	KNN(1)	48.8449 %	43.1021 %	50 %	30 %	10
	Random Forest	37.2604 %	45.8099 %	50 %	30 %	10
Dataset 4	Decision Tree	46.1417%	45.6019%	50 %	30 %	10
	Logistic Regression	51.7334%	51.8137%	50 %	30 %	10
	KNN(25)	50.9153%	51.1267%	50 %	30 %	10
	Naïve Bayes	52.8070%	52.9293%	50 %	30 %	10
	Random Forest	55.5888%	54.8771%	50 %	30 %	10
Dataset 5	Decision Tree	49.2648%	46.0970%	50 %	30 %	10
	Logistic Regression	45.1613%	44.9350%	50 %	30 %	10
	KNN(25)	53.7710%	49.9125%	50 %	30 %	10
	Naïve Bayes	16.3760%	19.9452%	50 %	30 %	10
	Random Forest	77.5409%	74.2253%	50 %	30 %	10

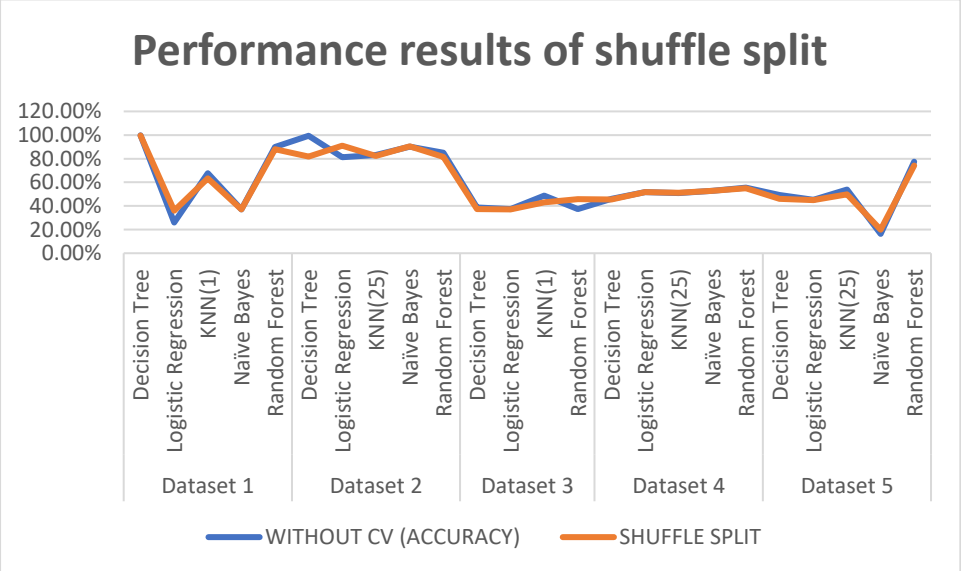


Fig 21: Visualization of the result of different classification algorithms with shuffle split

Performance analysis of Ensemble Classifiers:

Stacking:

Table 8 represents the results obtained from each dataset applying stack generalization. From the following table, we observed that stacking an ensemble of the three best classifier models combined with Best among all meta-models yields a better result. When weak learners have stacked up with the best learner meta-model, it has improved the accuracy at a great rate. The combinations hence prove to be more powerful than the single classification model.

The stacking is traditionally performed on stacking the weak learners and using the strong meta models to attain better results. But here to explore the behavior of the stacking process. Whether it shows the same improvement while stacking the best performers.

Observation shows that in all aspects of combination, the stacking has performed pretty well compared to basic models.

Table 8: performance analysis of dynamic ensemble classifier

Dataset	**Model1**	**Model2**	**Model3**	**Meta Model**	**Acc**	**Recall**	**Precision**	**F1 score**
Dataset 1	KNN	DT	RF	DT	99.69%	99.69%	99.69%	99.69%
Dataset 1	KNN	DT	GB	DT	99.94%	99.94%	99.94%	99.94%
Dataset 1	KNN	LR	NB	KNN	64.21%	64.21%	64.46%	64.21%
Dataset 3	MLP	RF	LR	LR	84.70%	84.70%	84.70%	84.70%
Dataset 3	DT	RF	NB	NB	96.12%	96.12%	96.12%	96.12%
Dataset 3	KNN	LR	NB	NB	86.39%	86.39%	86.39%	86.39%
Dataset 2	MLP	RF	LR	LR	76.74%	76.74%	76.74%	76.74%
Dataset 2	DT	RF	GB	DT	98.88%	98.88%	98.88%	98.88%
Dataset 2	KNN	DT	MLP	NB	92.92%	92.92%	92.92%	92.92%
Dataset 4	KNN	LR	NB	RF	51.73%	51.73%	51.73%	51.73%
Dataset 4	GB	RF	LR	RF	56.14%	56.14%	48.76%	56.14%
Dataset 4	NB	DT	KNN	RF	53.80%	53.80%	38.60%	53.80%
Dataset 5	DT	RF	KNN	RF	74.80%	74.80%	75.48%	74.80%
Dataset 5	DT	RF	GB	RF	64.85%	64.85%	60.26%	64.85%
Dataset 5	LR	KNN	NB	DT	56.06%	56.06%	46.09%	56.06%

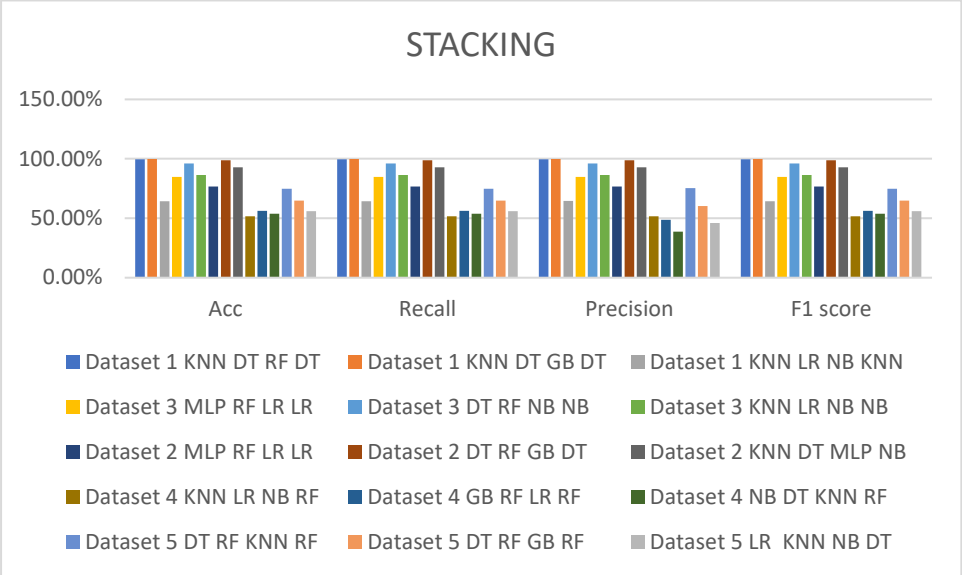


Fig 22: Visualization of the performance of different benchmark datasets with Stacking

Voting:

Table 9 below shows the observation of the Hard Voting technique applied in our work for the selected datasets. After validating the models using hard voting, we can observe a little increase in decimal values of accuracy percentage. The further hard vote is selected over Soft voting because not all the models are suitable or work well with the probability distribution.

Table 9: Result with Hard voting techniques

Dataset	**Model 1**	**Model 2**	**Model 3**	**Acc**	**Recall**	**Precision**	**F1 score**
Dataset 1	KNN	DT	RF	96.22%	96.22%	96.20%	96.22%
Dataset 3	KNN	DT	NB	96.81%	96.81%	96.81%	96.81%
Dataset 2	KNN	DT	NB	91.24%	91.24%	91.24%	91.24%
Dataset 4	NB	LR	RF	53.52%	53.52%	35.01%	53.52%
Dataset 5	KNN	DT	RF	67.69%	67.69%	77.48%	67.69%

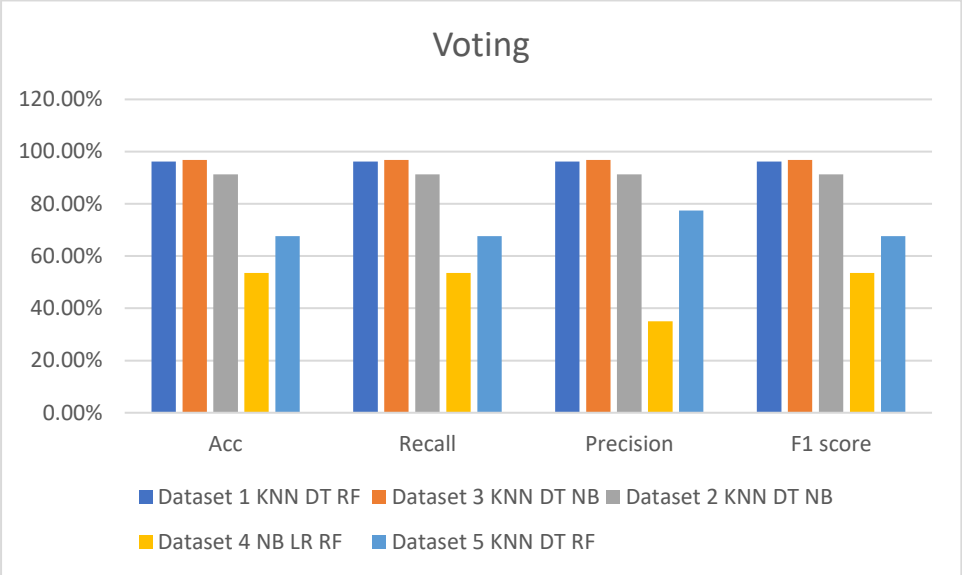


Fig 23: Visualization of the performance of different benchmark datasets with Voting

Boosting:

In boosting, adaptive boosting and gradient boosting were applied to all the datasets, and the following results shown in table 10 is obtained. For Dataset 2, the gradient boosting technique didn't work due to the overfitting because of adding too many trees.

Table 10: Comparison of Adaptive boosting and Gradient boosting techniques.

Dataset	**Adaptive Boosting**				**Gradient Boosting**			
	Acc	Recall	Precision	F1 score	Acc	Recall	Precision	F1 score
Dataset 1	33.68%	33.68%	19.52%	30.68%	99.88%	99.88%	99.88%	99.88%
Dataset 3	78.60%	78.60%	78.60%	78.60%	90.23%	90.23%	90.23%	90.23%
Dataset 2	87.36%	87.36%	87.36%	87.36%	-	-	-	-
Dataset 4	53.36%	53.36%	41.89%	53.36%	55.91%	55.91%	48.88%	55.91%
Dataset 5	54.54%	54.54%	44.50%	54.54%	68.49%	68.49%	73.87%	68.49%

Fig 24, depicts the performance of Adaptive and Gradient boosting in various benchmark datasets.

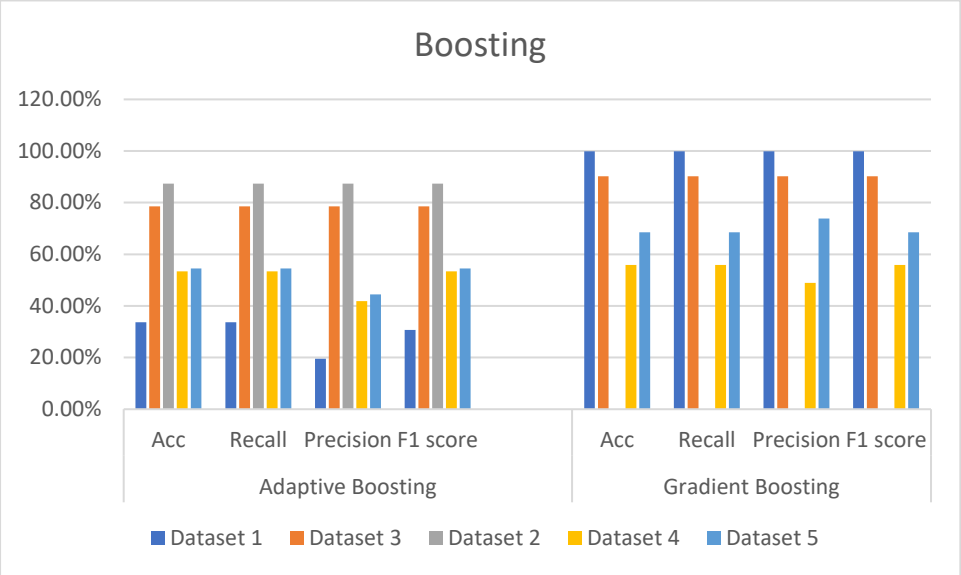


Fig 24: Visualization of the performance of different benchmark datasets with Boosting (Adaptive and Gradient boosting)

Result of Dynamic Classifier and Dynamic Ensemble Classifier algorithms:

The dynamic ensemble algorithm works better than the other algorithms discussed in this research because members are selected just in time, depending on the specific input pattern that requires forecasting.

Table 11 and Fig 25 show the result of various dynamic classifiers (OLA) and dynamic ensemble algorithms used for the different datasets that we have taken:

Table 11: Results of various dynamic classifiers

DATASET	OLA	KNORA-E	KNORA-U
Dataset 1	99.5%	99.8%	93.6%
Dataset 2	95.7%	96.3%	96.1%
Dataset 3	78.1%	85.7%	82.6%
Dataset 4	46.3%	48.5%	93.6%
Dataset 5	49.1%	52.6%	93.5%

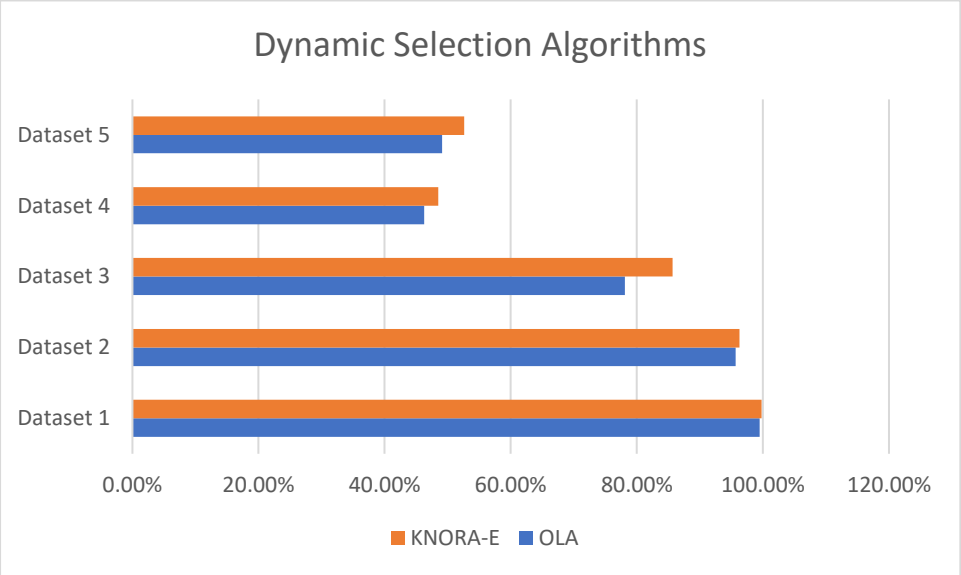


Fig 25: Visualization of the performance of different benchmark datasets with dynamic selection algorithms

The DES is promising because it relies entirely on selecting the best competent ensemble classifier for forecasting each split of sample data. Moreover, it performs all the integration, pooling, and selection independently. The result further shows that dynamic ensemble selection can perform better than any single model in the pool and is more beneficial than averaging all the static ensemble selections.

5. Conclusion

We experimented with many machine learning models for classifying crime. The evaluation of these ML classifiers was done mainly in terms of accuracy. Overall, our results reveal a powerful impact of the dynamic algorithms, i.e., KNORA-U, and KNORA-E algorithms, achieved a reliable accuracy above 90% in classifying and identifying the crime types or categories. This outcome is anticipated because of the selection of the classifiers pool to predict every sample test data split. For every dataset apt classifier is chosen automatically without any pre-fixation. This research has also paved a path to discussing many more general learnings. Not all the data can be compatible with the desired algorithm, which has performed well with

other data. Hence the algorithms can create either positive or negative on their performances on a particular dataset. Data pre-processing steps like feature engineering, outliers removal, and hyperparameter tuning have played a vital role in better accurate crime classification. When applied to the data, ensemble algorithms yield better accuracy than single algorithms. After several trials, briefly translated, our findings indicate that the Stacking of the algorithms, irrespective of their performance individually, can outperform the algorithm's accuracy in forecasting the test data to a reasonable extent. Altogether, our outcomes reinforce the essence of choosing dynamic ensemble classification algorithms for the crime domain, which in turn helps save many unfortunate situations that are yet to happen. Future studies could fruitfully explore the possibility of using Deep learning algorithms further by using them dynamically, which can handle larger quantities of data. Additionally, web and mobile applications can be built to be used by law enforcement to update crime data and analyze and forecast the same.

References

Abouelnaga, Y. (2016). *San Francisco Crime Classification*. http://arxiv.org/abs/1607.03626

Ahmed, S., Gentili, M., Sierra-Sosa, D., & Elmaghraby, A. S. (2022). Multi-layer data integration technique for combining heterogeneous crime data. *Information Processing and Management*, *59*(3), 102879. https://doi.org/10.1016/j.ipm.2022.102879

Alasadi, S. A., & Bhaya, W. S. (2017). Review of data preprocessing techniques in data mining. *Journal of Engineering and Applied Sciences*, *12*(16). https://doi.org/10.3923/jeasci.2017.4102.4107

Albahli, S., Alsaqabi, A., Aldhubayi, F., Rauf, H. T., Arif, M., & Mohammed, M. A. (2020). Predicting the type of crime: Intelligence gathering and crime analysis. *Computers, Materials and Continua*, *66*(3), 2317–2341. https://doi.org/10.32604/cmc.2021.014113

Aldossari, B. S., Alqahtani, F. M., Alshahrani, N. S., Alhammam, M. M., Alzamanan, R. M., Aslam, N., & Irfanullah. (2020). A comparative study of decision tree and naive bayes

machine learning model for crime category prediction in chicago. *PervasiveHealth: Pervasive Computing Technologies for Healthcare*, 34–38. https://doi.org/10.1145/3379247.3379279

Ali, W. (2017). Phishing Website Detection based on Supervised Machine Learning with Wrapper Features Selection. *International Journal of Advanced Computer Science and Applications*, *8*(9). https://doi.org/10.14569/ijacsa.2017.080910

Aljamaan, H., & Alazba, A. (2020). Software defect prediction using tree-based ensembles. *PROMISE 2020 - Proceedings of the 16th ACM International Conference on Predictive Models and Data Analytics in Software Engineering, Co-Located with ESEC/FSE 2020*. https://doi.org/10.1145/3416508.3417114

Almaw, A., & Kadam, K. (2018). Survey Paper on Crime Prediction using Ensemble Approach. *International Journal of Pure and Applied Mathematics*, *118*(8).

Alsini, R., Almakrab, A., Ibrahim, A., & Ma, X. (2021). Improving the outlier detection method in concrete mix design by combining the isolation forest and local outlier factor. *Construction and Building Materials*, *270*. https://doi.org/10.1016/j.conbuildmat.2020.121396

Alves, L. G. A., Ribeiro, H. V., & Rodrigues, F. A. (2018). Crime prediction through urban metrics and statistical learning. *Physica A: Statistical Mechanics and Its Applications*, *505*. https://doi.org/10.1016/j.physa.2018.03.084

Anusha, P. V., Anuradha, C., Chandra Murty, P. S. R., & Kiran, C. S. (2019). Detecting outliers in high dimensional data sets using Z-score methodology. *International Journal of Innovative Technology and Exploring Engineering*, *9*(1). https://doi.org/10.35940/ijitee.A3910.119119

Atallah, R., & Al-Mousa, A. (2019). Heart Disease Detection Using Machine Learning Majority Voting Ensemble Method. *2019 2nd International Conference on New Trends*

in Computing Sciences, ICTCS 2019 - Proceedings. https://doi.org/10.1109/ICTCS.2019.8923053

Bae, I., & Ji, U. (2019). Outlier detection and smoothing process for water level data measured by ultrasonic sensor in stream flows. *Water (Switzerland)*, *11*(5). https://doi.org/10.3390/w11050951

Bardhi, O., & Zapirain, B. G. (2021). Machine learning techniques applied to electronic healthcare records to predict cancer patient survivability. *Computers, Materials and Continua*, *68*(2). https://doi.org/10.32604/cmc.2021.015326

Berrar, D. (2018). Bayes' theorem and naive bayes classifier. In *Encyclopedia of Bioinformatics and Computational Biology: ABC of Bioinformatics* (Vols. 1–3). https://doi.org/10.1016/B978-0-12-809633-8.20473-1

Bommert, A., Sun, X., Bischl, B., Rahnenführer, J., & Lang, M. (2020). Benchmark for filter methods for feature selection in high-dimensional classification data. *Computational Statistics and Data Analysis*, *143*. https://doi.org/10.1016/j.csda.2019.106839

Britto, A. S., Sabourin, R., & Oliveira, L. E. S. (2014). Dynamic selection of classifiers - A comprehensive review. *Pattern Recognition*, *47*(11), 3665–3680. https://doi.org/10.1016/j.patcog.2014.05.003

Brownlee, J. (2020). Train-Test Split for Evaluating Machine Learning Algorithms. In *Machine Learning Mastery*.

Budilaksana, D., Sukarsa, I. M., & Wiranatha, A. A. K. A. C. (2021). Implementing kNearest Neighbor Methods to Predict Car Prices. *Jurnal Ilmiah Merpati (Menara Penelitian Akademika Teknologi Informasi)*. https://doi.org/10.24843/jim.2021.v09.i01.p06

Calabrese, B. (2018). Data cleaning. In *Encyclopedia of Bioinformatics and Computational Biology: ABC of Bioinformatics* (Vols. 1–3). https://doi.org/10.1016/B978-0-12-809633-

8.20458-5

Chiou, J. M., Zhang, Y. C., Chen, W. H., & Chang, C. W. (2014). A functional data approach to missing value imputation and outlier detection for traffic flow data. *Transportmetrica B*, *2*(2). https://doi.org/10.1080/21680566.2014.892847

Connelly, L. (2020). Logistic regression. *MEDSURG Nursing*, *29*(5). https://doi.org/10.46692/9781847423399.014

Cruz, R. M. O., Hafemann, L. G., Sabourin, R., & Cavalcanti, G. D. C. (2020). DESlib: A dynamic ensemble selection library in python. *Journal of Machine Learning Research*, *21*.

Cruz, R. M. O., Sabourin, R., & Cavalcanti, G. D. C. (2018). Dynamic classifier selection: Recent advances and perspectives. *Information Fusion*, *41*. https://doi.org/10.1016/j.inffus.2017.09.010

Cuneen, J., & Tobar, D. (2021). Chi-square Tests. In *Sport Industry Research and Analysis*. https://doi.org/10.4324/9781315212944-28

da Silveira, A. C. M., Sobrinho, Á., da Silva, L. D., Costa, E. de B., Pinheiro, M. E., & Perkusich, A. (2022). Exploring Early Prediction of Chronic Kidney Disease Using Machine Learning Algorithms for Small and Imbalanced Datasets. *Applied Sciences (Switzerland)*, *12*(7). https://doi.org/10.3390/app12073673

Darapureddy, N., Karatapu, N., & Battula, T. K. (2019). Research of machine learning algorithms using k-fold cross validation. *International Journal of Engineering and Advanced Technology*, *8*(6 Special issue). https://doi.org/10.35940/ijeat.F1043.0886S19

Doshi, B. (2011). Handling Missing Values in Data Mining. *Cs.Rit.Edu*.

Franke, T. M., Ho, T., & Christie, C. A. (2012). The Chi-Square Test: Often Used and More Often Misinterpreted. *American Journal of Evaluation*, *33*(3).

https://doi.org/10.1177/1098214011426594

Ghorbanzadeh, O., Shahabi, H., Mirchooli, F., Valizadeh Kamran, K., Lim, S., Aryal, J., Jarihani, B., & Blaschke, T. (2020). Gully erosion susceptibility mapping (GESM) using machine learning methods optimized by the multi-collinearity analysis and K-fold cross-validation. *Geomatics, Natural Hazards and Risk*, *11*(1). https://doi.org/10.1080/19475705.2020.1810138

Hajela, G., Chawla, M., & Rasool, A. (2021). A multi-dimensional crime spatial pattern analysis and prediction model based on classification. *ETRI Journal*, *43*(2), 272–287. https://doi.org/10.4218/etrij.2019-0306

Hale, C., & Liu, F. (n.d.). *CS 229 Project Report: San Francisco Crime Classification*.

Hamdi, A., Shaban, K., Erradi, A., Mohamed, A., Rumi, S. K., & Salim, F. D. (2022). Spatiotemporal data mining: a survey on challenges and open problems. *Artificial Intelligence Review*, *55*(2). https://doi.org/10.1007/s10462-021-09994-y

Hossain, S., Abtahee, A., Kashem, I., Hoque, M. M., & Sarker, I. H. (2020). *Crime Prediction Using Spatio-Temporal Data*. http://arxiv.org/abs/2003.09322

Ismail, M., Hassan, N., & Bafjaish, S. S. (2020). Comparative Analysis of Naive Bayesian Techniques in Health-Related for Classification Task. *Journal of Soft Computing and Data Mining*, *1*(2).

Janardhan, N., & Kumaresh, N. (2022). Improving Depression Prediction Accuracy Using Fisher Score-Based Feature Selection and Dynamic Ensemble Selection Approach Based on Acoustic Features of Speech. *Traitement Du Signal*, *39*(1), 87–107. https://doi.org/10.18280/ts.390109

Jangra, M., Cse, M.-T., & Kalsi, S. (2019). Naïve Bayes Approach for the Crime Prediction in Data Mining. In *International Journal of Computer Applications* (Vol. 178, Issue 14).

Johnson, T. F., Isaac, N. J. B., Paviolo, A., & González-Suárez, M. (2021). Handling missing values in trait data. *Global Ecology and Biogeography*, *30*(1). https://doi.org/10.1111/geb.13185

Josse, J., & Husson, F. (2012). Handling missing values in exploratory multivariate data analysis methods. *Journal de La Société Française de …*, *153*(2).

Keerthi R, Kirthika B, Pavithraa S, & Gowri, V. (2020). PREDICTION of CRIME RATE ANALYSIS using MACHINE LEARNING APPROACH. *International Research Journal of Engineering and Technology*.

Khaire, U. M., & Dhanalakshmi, R. (2022). Stability of feature selection algorithm: A review. In *Journal of King Saud University - Computer and Information Sciences* (Vol. 34, Issue 4). https://doi.org/10.1016/j.jksuci.2019.06.012

Khan, M., Ali, A., & Alharbi, Y. (2022). Predicting and Preventing Crime: A Crime Prediction Model Using San Francisco Crime Data by Classification Techniques. *Complexity*, *2022*. https://doi.org/10.1155/2022/4830411

Khatun, M. R., Ayon, S. I., Hossain, M. R., & Alam, M. J. (2021). Data mining technique to analyse and predict crime using crime categories and arrest records. *Indonesian Journal of Electrical Engineering and Computer Science*, *22*(2). https://doi.org/10.11591/ijeecs.v22.i2.pp1052-1060

Ko, A. H. R., Sabourin, R., & Britto, A. S. (2008). From dynamic classifier selection to dynamic ensemble selection. *Pattern Recognition*, *41*(5). https://doi.org/10.1016/j.patcog.2007.10.015

Kohavi, R., & John, G. H. (1997). Wrappers for feature subset selection. *Artificial Intelligence*, *97*(1–2). https://doi.org/10.1016/s0004-3702(97)00043-x

Kshatri, S. S., Singh, D., Narain, B., Bhatia, S., Quasim, M. T., & Sinha, G. R. (2021). An

Empirical Analysis of Machine Learning Algorithms for Crime Prediction Using Stacked Generalization: An Ensemble Approach. *IEEE Access*, *9*, 67488–67500. https://doi.org/10.1109/ACCESS.2021.3075140

Lamari, Y., Freskura, B., Abdessamad, A., Eichberg, S., & de Bonviller, S. (2020). Predicting spatial crime occurrences through an efficient ensemble-learning model. *ISPRS International Journal of Geo-Information*, *9*(11). https://doi.org/10.3390/ijgi9110645

Larios, G. H. (2016). *Case Study Report San Francisco Crime Classification*.

Li, J., Cheng, K., Wang, S., Morstatter, F., Trevino, R. P., Tang, J., & Liu, H. (2017). Feature selection: A data perspective. In *ACM Computing Surveys* (Vol. 50, Issue 6). https://doi.org/10.1145/3136625

Liu, C. (2020). *Data Transformation: Standardization vs Normalization*. KDnuggets.

Lu, R., & Li, L. (2020). Application of an ensemble learning based classifier in crime prediction. *Proceedings of 2019 the 9th International Workshop on Computer Science and Engineering, WCSE 2019*. https://doi.org/10.18178/wcse.2019.06.019

Machine Learning and the Internet of Medical Things in Healthcare - Google Books. (n.d.). Retrieved January 9, 2023, from https://books.google.co.in/books?hl=en&lr=&id=wHQJEAAAQBAJ&oi=fnd&pg=PP1&dq=Saravanan,+Vijayalakshmi,+and+Singh,+Akansha.+Machine+Learning+%26+Internet+of+Things+(IoT)+for+Urban+Intelligence.+In+Machine+Learning+%26+Internet+of+Things+(IoT)+for+Urban+Intelligence.&ots=3Rm7CAxzkT&sig=fbm88AhX85ohHLig3eX3jVLOf5M&redir_esc=y#v=onepage&q&f=false

Maddileti, T., Sai Madhav, V., Sashank, S., & Rao, G. S. (2020). Crime Data Analysis Using Machine Learning Models. *International Journal of Advanced Science and Technology*, *29*(9s).

Malikhah, M., Sarno, R., & Sabilla, S. I. (2021). Ensemble Learning for Optimizing Classification of Pork Adulteration in Beef Based on Electronic Nose Dataset. *International Journal of Intelligent Engineering and Systems*, *14*(4). https://doi.org/10.22266/ijies2021.0831.05

More, S., Mench, S., Kuge, S., Bagwan, H., & Professor, A. (2021). Crime Prediction Using Machine Learning Approach. *International Journal of Advanced Research in Computer and Communication Engineering*, *10*(5). https://doi.org/10.17148/IJARCCE.2021.10537

Nurhopipah, A., & Hasanah, U. (2020). Dataset Splitting Techniques Comparison For Face Classification on CCTV Images. *IJCCS (Indonesian Journal of Computing and Cybernetics Systems)*, *14*(4). https://doi.org/10.22146/ijccs.58092

Oliveira, D. V. R., Cavalcanti, G. D. C., Porpino, T. N., Cruz, R. M. O., & Sabourin, R. (n.d.). *K-Nearest Oracles Borderline Dynamic Classifier Ensemble Selection*.

Padmanabhan, K. A., & D, L. S. (2019). Prediction of Crime Rate Analysis Using Supervised Classification Machine Learning Approach. *International Research Journal of Engineering and Technology*.

Pilnenskiy, N., & Smetannikov, I. (2020). Feature selection algorithms as one of the python data analytical tools. *Future Internet*, *12*(3). https://doi.org/10.3390/fi12030054

Pradhan, I., Eirinaki, M., Potika, K., & Potikas, P. (2019). Exploratory data analysis and crime prediction for smart cities. *ACM International Conference Proceeding Series*. https://doi.org/10.1145/3331076.3331114

Puurula, A., Read, J., & Bifet, A. (2014). *Kaggle LSHTC4 Winning Solution*. https://deepai.org/publication/k-nearest-oracles-borderline-dynamic-classifier-ensemble-selection

Qazi, N., & Wong, B. L. W. (2019). An interactive human centered data science approach

towards crime pattern analysis. *Information Processing and Management*, *56*(6). https://doi.org/10.1016/j.ipm.2019.102066

Rácz, A., Bajusz, D., & Héberger, K. (2021). Effect of dataset size and train/test split ratios in qsar/qspr multiclass classification. *Molecules*, *26*(4). https://doi.org/10.3390/molecules26041111

Rockett, P. (2022). Constant optimization and feature standardization in multiobjective genetic programming. *Genetic Programming and Evolvable Machines*, *23*(1). https://doi.org/10.1007/s10710-021-09410-y

Rotarou, E. S. (2018). Does Municipal Socioeconomic Development Affect Public Perceptions of Crime? A Multilevel Logistic Regression Analysis. *Social Indicators Research*, *138*(2). https://doi.org/10.1007/s11205-017-1669-2

Safat, W., Asghar, S., & Gillani, S. A. (2021). Empirical Analysis for Crime Prediction and Forecasting Using Machine Learning and Deep Learning Techniques. *IEEE Access*, *9*, 70080–70094. https://doi.org/10.1109/ACCESS.2021.3078117

Shiffler, R. E. (1988). Maximum z scores and outliers. *American Statistician*, *42*(1). https://doi.org/10.1080/00031305.1988.10475530

Singh, K., & Upadhyaya, S. (2012). Outlier Detection: Applications And Techniques. *International Journal of Computer …*, *9*(1).

Song, Y. Y., & Lu, Y. (2015). Decision tree methods: applications for classification and prediction. *Shanghai Archives of Psychiatry*, *27*(2). https://doi.org/10.11919/j.issn.1002-0829.215044

Sri, L. A., Manvitha, K., Amulya, G., Sanjuna, I. S., & Pavani, V. (2020). *FBI CRIME ANALYSIS AND PREDICTION USING MACHINE LEARNING*. *11*. www.jespublication.com

Srivastava, A. K., Kumar, Y., & Singh, P. K. (2022). Hybrid diabetes disease prediction framework based on data imputation and outlier detection techniques. *Expert Systems*, *39*(3). https://doi.org/10.1111/exsy.12785

Sunitha, L., Balraju, M., Sasikiran, J., & Ramana, E. V. (2014). Automatic Outlier Identification in Data Mining Using IQR in Real-Time Data. *International Journal of Advanced Research in Computer and Communication Engineering*, *3*.

Tamilarasi, P., & Rani, R. U. (2020). Diagnosis of Crime Rate against Women using k-fold Cross Validation through Machine Learning. *Proceedings of the 4th International Conference on Computing Methodologies and Communication, ICCMC 2020*. https://doi.org/10.1109/ICCMC48092.2020.ICCMC-000193

Tamir, A., Watson, E., Willett, B., Hasan, Q., & Yuan, J.-S. (2021). Crime Prediction and Forecasting using Machine Learning Algorithms. In *Article in International Journal of Computer Science and Information Technology*. https://www.researchgate.net/publication/355872171

Tembusai, Z. R., Mawengkang, H., & Zarlis, M. (2021). K-Nearest Neighbor with K-Fold Cross Validation and Analytic Hierarchy Process on Data Classification. *International Journal of Advances in Data and Information Systems*, *2*(1). https://doi.org/10.25008/ijadis.v2i1.1204

Thara, T. D. K., Prema, P. S., & Xiong, F. (2019). Auto-detection of epileptic seizure events using deep neural network with different feature scaling techniques. *Pattern Recognition Letters*, *128*. https://doi.org/10.1016/j.patrec.2019.10.029

Tripathi, M. (2020). Underfitting and Overfitting in Machine Learning. *DataScience Foundation*, *9624670*.

Tuson, M., Turlach, B., Murray, K., Kok, M. R., Vickery, A., & Whyatt, D. (2021). Predicting future geographic hotspots of potentially preventable hospitalisations using

all subset model selection and repeated K-fold cross-validation. *International Journal of Environmental Research and Public Health*, *18*(19). https://doi.org/10.3390/ijerph181910253

Vassallo, D., Vella, V., & Ellul, J. (2021). Application of Gradient Boosting Algorithms for Anti-money Laundering in Cryptocurrencies. *SN Computer Science*, *2*(3). https://doi.org/10.1007/s42979-021-00558-z

VijayaKumar, M., & Alhadidi, B. (2014). Crime Classification Algorithm for Mining Crime Hot Spot and Cold Spot. *International Journal of Computing Academic Research*, *3*(2), 58–69. http://www.meacse.org/ijcar

Vink, M., Netten, N., Bargh, M. S., Van Den Braak, S., & Choenni, S. (2020). Mapping crime descriptions to law articles using deep learning. *ACM International Conference Proceeding Series*. https://doi.org/10.1145/3428502.3428507

Wan, X. (2019). Influence of feature scaling on convergence of gradient iterative algorithm. *Journal of Physics: Conference Series*, *1213*(3). https://doi.org/10.1088/1742-6596/1213/3/032021

Wang, H., & Ma, S. (2021). Preventing crimes against public health with artificial intelligence and machine learning capabilities. *Socio-Economic Planning Sciences*. https://doi.org/10.1016/j.seps.2021.101043

Wang, J., Wang, X., Yang, Y., Zhang, H., & Fang, B. (2020). A review of data cleaning methods for web information system. In *Computers, Materials and Continua* (Vol. 62, Issue 3). https://doi.org/10.32604/cmc.2020.08675

Wang, X., & Wang, C. (2020). Time Series Data Cleaning: A Survey. *IEEE Access*, *8*. https://doi.org/10.1109/ACCESS.2019.2962152

Wang, Zhihui, & Wang, J. (2021). Applications of Machine Learning in Public Security

Information and Resource Management. *Scientific Programming*, *2021*. https://doi.org/10.1155/2021/4734187

Wang, Zihao, Song, Z., & Zhou, T. (2021). Machine learning for ionic liquid toxicity prediction. *Processes*, *9*(1). https://doi.org/10.3390/pr9010065

Wibowo, A. H., & Oesman, T. I. (2020). The comparative analysis on the accuracy of k-NN, Naive Bayes, and Decision Tree Algorithms in predicting crimes and criminal actions in Sleman Regency. *Journal of Physics: Conference Series*, *1450*(1). https://doi.org/10.1088/1742-6596/1450/1/012076

Xu, Q. S., & Liang, Y. Z. (2001). Monte Carlo cross validation. *Chemometrics and Intelligent Laboratory Systems*, *56*(1). https://doi.org/10.1016/S0169-7439(00)00122-2

Xu, S., Lu, B., Baldea, M., Edgar, T. F., Wojsznis, W., Blevins, T., & Nixon, M. (2015). Data cleaning in the process industries. *Reviews in Chemical Engineering*, *31*(5). https://doi.org/10.1515/revce-2015-0022

Yu, L., Zhou, R., Chen, R., & Lai, K. K. (2022). Missing Data Preprocessing in Credit Classification: One-Hot Encoding or Imputation? *Emerging Markets Finance and Trade*, *58*(2). https://doi.org/10.1080/1540496X.2020.1825935

Yuki, J. Q., Mahfil Quader Sakib, M., Zamal, Z., Habibullah, K. M., & Das, A. K. (2019). Predicting crime using time and location data. *PervasiveHealth: Pervasive Computing Technologies for Healthcare*, 124–128. https://doi.org/10.1145/3348445.3348483

Zhang, S., Li, X., Zong, M., Zhu, X., & Cheng, D. (2017). Learning k for kNN Classification. *ACM Transactions on Intelligent Systems and Technology*, *8*(3). https://doi.org/10.1145/2990508