RAJEEV GANDHI MEMORIAL COLLEGE OF ENGINEERING & TECHNOLOGY (AUTONOMOUS)

EDUCATION FOR PEACE

(ESTD-1995)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CRIME PREDICTION USING A MACHINE LEARNING ALGORITHM

EXISTING SYSTEM-ALGORITHMS

Related work- 1

Title: "A Meta-Analysis of ML Models in Crime Forecasting"

Authors: P. Karthik, P. Jayanth, K. Tharun Nayak, and K. Anil Kumar

1.DECISION TREE

The decision tree works by creating a flowchart-like structure where each internal node tests an attribute (feature), each branch represents the outcome of the test, and each leaf node indicates the class label (crime type). This model predicts crime categories based on features such as location, time, and type.

RESULTS: The system achieved high accuracy, with a reported 98% precision for training data and 95% for testing, demonstrating its effectiveness in crime classification.

DRAWBACKS: While transparent and interpretable, decision trees can overfit the training data, leading to poor generalization on unseen data, especially with complex and noisy datasets.

2.BAGGING CLASSIFIER (BOOTSTRAP AGGREGATING)

The bagging classifier creates multiple bootstrap samples of the dataset and trains separate base classifiers (e.g., decision trees) on these samples. The predictions are then aggregated (via voting) to improve stability and accuracy.

RESULTS: Increases prediction accuracy and robustness by reducing variance. It enhances model stability and mitigates overfitting, complementing decision trees' interpretability.

DRAWBACKS: Although more robust, the ensemble can be computationally intensive, especially with many base classifiers, and less interpretable than single models.

3.RANDOM FOREST CLASSIFIER

An ensemble of decision trees trained on different subsets of data, with feature randomness introduced at each split. It outputs the mode of classifications across the trees.

RESULTS: High accuracy and robustness, capable of handling high-dimensional data and complex interactions, making it effective for large-scale crime datasets.

DRAWBACKS: Computationally demanding and less transparent ("black-box") in explaining individual predictions.

4.ENSEMBLE LEARNING TECHNIQUES (e.g., SVM, Naïve Bayes, J48, SMO)

The study compares various classifiers, such as Support Vector Machines (SVM), Naïve Bayes, J48 decision trees, and SMO (Sequential Minimal Optimization for SVM). These are combined using ensemble strategies like stacking.

RESULTS: The ensemble approach achieved a near-perfect accuracy of 99.5%, outperforming individual classifiers.

DRAWBACKS: Complexity in model tuning, higher computational requirements, and potential

difficulty in interpretability.

5.DEEP NEURAL NETWORKS (Fully Connected Neural Networks)

Employed for complex pattern recognition in crime data, addressing non-linear relationships and limited domain knowledge.

RESULTS: Capable of capturing intricate patterns, offering improved predictive accuracy in complex scenarios.

DRAWBACKS: Require large datasets, intensive computation, and can be difficult to interpret.

6.TIME SERIES MODDELS (LSTM, ARIMA)

Used for crime trend forecasting over time; LSTM captures long-term dependencies, while ARIMA models short-term patterns.

RESULTS: High predictive performance, with ARIMA indicating possible future decreases or increases in crime.

DRAWBACKS: Sensitive to data quality, parameter tuning complexity, and assumptions (e.g., stationarity in ARIMA).

7.CLASSIFICATION ALGORITHMS (Logistic Regression, KNN, XGBoost)

Various models tested for crime classification; KNN classifies based on proximity, while XGBoost enhances trees with boosting.

RESULTS: Improved accuracy and prediction efficiency, with models like XGBoost performing well.

DRAWBACKS: KNN can be slow with large datasets; boosting models can overfit if not properly regularized.

Related work- 2

Title: "Machine Learning for Smarter Crime Prevention Strategies"

Authors: Sridharan S. et al

1. K-Nearest Neighbors (KNN)

KNN was used to classify crimes based on similar cases nearby in data. For crime prediction, it helped identify whether an area is likely to have crimes based on its similarity to other crime-prone areas.

RESULTS: It provided high accuracy (over 75%) in identifying crime hotspots and predicting future crimes.

DRAWBACKS: KNN can be slow with large datasets because it compares a new case with many existing data points. It also needs to choose the right number of neighbors, which can be tricky.

2. Naive Bayes

Naive Bayes predicted the likelihood of crimes occurring, such as repeat offenses, using probabilities based on past data.

RESULTS: It achieved an accuracy of around 78%. It was faster and effective for crime type classification.

DRAWBACKS: Assumes features are independent, which is often not true in real crime data, leading to less accurate results in some cases.

3. Linear Regression

Linear regression was used to estimate the age of victims or offenders and to forecast crime trends over time.

RESULTS: It helped understand relationships between variables like age, gender, and crime occurrence.

DRAWBACKS: It only works well if data has a linear relationship. Non-linear patterns can lead to inaccurate predictions.

4. Clustering (K-means and others)

Clustering grouped similar crimes and areas. This helped identify crime hotspots and patterns.

RESULTS: Over 75% of crime data was successfully grouped, highlighting high-crime zones.

DRAWBACKS: Choosing the number of clusters can be difficult. Poorly chosen clusters may miss important patterns.

5. Deep Learning (CNN)

Convolutional Neural Networks (CNN) analyzed images related to crimes, such as photos of crime scenes or suspects.

RESULTS: It achieved about 87% accuracy in recognizing images related to crimes.

DRAWBACKS: CNNs require lots of data and computational power. They can be complex to tune and interpret.

6. Random Forest

Random Forests combined many decision trees to predict the likelihood of crimes or factors influencing crime rates.

RESULTS: It performed better than some other methods, with around 81.35% accuracy.

DRAWBACKS: Can be computationally intensive and less transparent in showing how decisions are made.

Related work- 3

Title: "Simplified ML Models for Crime Data Analysis"

Authors: Prof. R. Hinduja, Ms. T. Tejasree, Ms. Harini Ramesh Babu

1. Random Forest Algorithm

The main algorithm used in the system. It builds many decision trees using different data samples and combines their results to predict crimes.

The system trains the Random Forest on crime data, selecting the best features like location, time, and crime type to predict the likelihood of different crimes.

RESULT: It provides accurate predictions with high reliability, handling large datasets well.

DRAWBACKS: It can be slow when training on very big data, and it may still produce biased results if data is imbalanced.

2. Decision Tree

Mentioned as a comparison model to evaluate performance.

It classifies crime data based on features by splitting data into branches.

RESULTS: Offers decent accuracy but less reliable than Random Forest because it can overfit (become too tailored to training data).

DRAWBACKS: Less accurate with complex data and prone to overfitting.

3. Support Vector Machine (SVM)

Used for crime classification, particularly cybercrime detection.

Finds the best boundary (hyperplane) that separates different crime categories.

RESULTS: Works well with clear data patterns, but performance drops with noisy or large datasets.

DRAWBACKS: Computationally intensive and less efficient with very large datasets.

4. Naïve Bayes

For initial classification tasks in crime data.

Applies probability models assuming features are independent to classify crimes.

RESULTS: Fast and handles big datasets easily, but accuracy can be low if data features are correlated.

DRAWBACKS: Oversimplifies real data assumptions, which can lead to mistakes.

5. Neural Networks

Mentioned as a deep learning method for analyzing temporal crime patterns.

Mimics brain functions to find complex patterns in crime data.

RESULTS: Can improve prediction accuracy but requires high computation power.

DRAWBACKS: Difficult to interpret (black-box nature) and compute-intensive.