Sameeksha Bhati, 10:5
ISSN (Online): 2348-4098
ISSN (Print): 2395-4752

Analysis and Prediction of Crime Detection Techniques Using Machine Learning Approach

Sameeksha Bhati, Assistant Professor Priyanshu Dhameniya
Department of Computer Science
Astral Institute of Technology and Research
Indore,MP,India

Abstract-The field of study known as machine learning examines how machines can learn to act autonomously. Self-driving cars, speech recognition, web search, and a deeper understanding of the human genome are just a few of the recent applications of machine learning. In addition, it has made it possible to make crime forecasts using historical data. Using nominal class labels, classification is a supervised prediction method. Weather forecasting, medical care, banking, homeland security, and corporate intelligence are just few of the numerous fields that have benefited from classification [6]. Data gathering, classification, pattern recognition, prediction, and visualization are typical steps in a machine learning-based approach to analyzing criminal behavior. Association analysis, classification and prediction, cluster analysis, and outlier analysis are examples of classic data mining methods that focus on structured data; newer methods can also extract useful insights from unstructured data.

Keywords- Weather forecasting, medical care, banking, homeland security, and corporate intelligence.

I. INTRODUCTION

To put it simply, crime is a societal and economic issue that has a negative impact on people's standard of living and the health of the economy [1]. How exactly criminal activity is carried out might vary widely from one community to the next. Factors like education, poverty, employment, and climate have been found to affect crime rates in previous studies on crime prediction.

Vancouver, British Columbia, Canada, is a major urban center with a large population and a rich cultural mosaic. Even though 2017 saw a 1.5% decline in Vancouver's overall crime rate, a persistent problem remains high rates of vehicle break-ins and theft. Residential burglaries in Vancouver fell by 27 percent when the Vancouver Police Department (VPD) used a crime-predictive algorithm to reduce such offenses.

Using information gathered from many sources and analyzed statistically, law enforcement agencies can studying this topic. Crime hotspots in London, UK were predicted using human behavioral data collected from mobile network activity, in addition to demographic information based on actual crime statistics. In [6], WEKA, an open-source data mining software, and 10-fold cross-validation were used to evaluate and contrast two different categorization algorithms: Decision Tree and Naive Bayesian.

The 1990 US Census, the 1990 US LEMAS survey, and the 1995 FBI UCR were used to create the socioeconomic, law enforcement, and crime datasets for this research. Factors such as driver, weather, vehicle, and road conditions were examined in a study of Ethiopian road accident trends [8]. A total of 18,288 accidents were employed across three different categorization algorithms (KNN, Naive Bayesian, and Decision tree). Accuracy in making predictions ranged from 79 to 81 percent across the board for the three systems.

II. PROPOSED METHODOLOGY

The suggested method's final stage is dependent on clustering with a suite of classifiers. Multiple

classifiers are used to enhance the consistency and precision of our results by compensating for each classifier's limitations and strengths to provide the best possible decision after examining all of the data in the set [17, 18].K-Nearest Neighbor (KNN), Decision Tree (DT), MLP Classifier, Extra Tree, Support Vector Machine (SVM), and XG-Boost are some of the classifiers we've used.

1. Decision tree classifier:

To classify data, a decision tree classifier is a straightforward but effective tool. A training model is constructed using the dataset using this flowchart-like supervised learning methodology. The purpose of a decision tree is to help find trends, connections, and hidden insights in data. The data in decision trees is presented in a very clear way.

These can be quickly and easily interpreted for use in real-time decision making. Other classifiers, such as the Support Vector Machine (SVM), the Nave Bayes classifier, and the Artificial Neural Network (ANN), act as black boxes where they only reveal the output, but decision trees show the conclusion through a straightforward flowchart-like structure. Because of their flexibility, decision trees are widely used in a wide range of contexts.

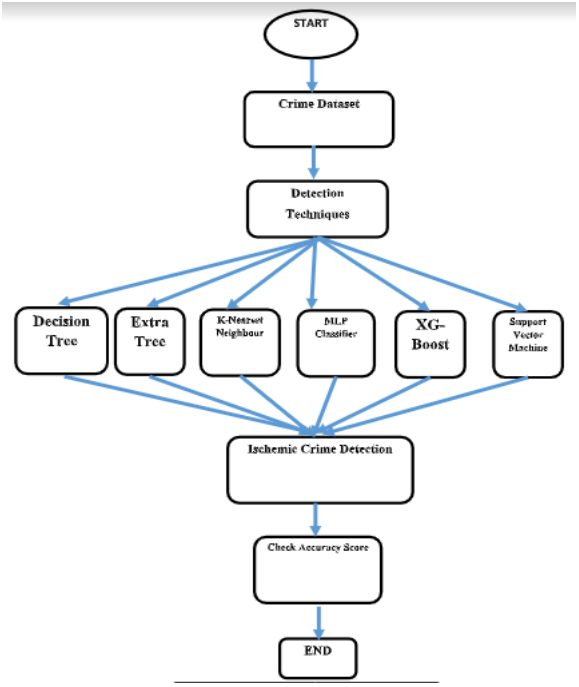


Fig 1.Flow chart of Proposed Algorithm.

If the size of the data set is growing rapidly, in terms of both the number of occurrences and the number of attributes, a decision tree is the optimal tool for analysis. Learning decision trees becomes a tedious and time-consuming process as data volumes rise.

Both training with huge data sets and building decision trees provide unique challenges. The literature is replete with the suggestions of various researchers on how to deal with these issues. In this chapter, we will examine a summary of the various methods that can be applied to decision trees. Methods for developing reliable and efficient decision trees are highlighted. To classify data in a supervised manner, a decision tree's set of nodes forms a tree structure.

There are two types of nodes: internal nodes and leaves. Decisions are made and the data set is partitioned at internal nodes, which is why they are also known as decision nodes. Predicted class, or decision class, is represented by leaves. Attributes are segmented using splitting criteria during decision tree development.

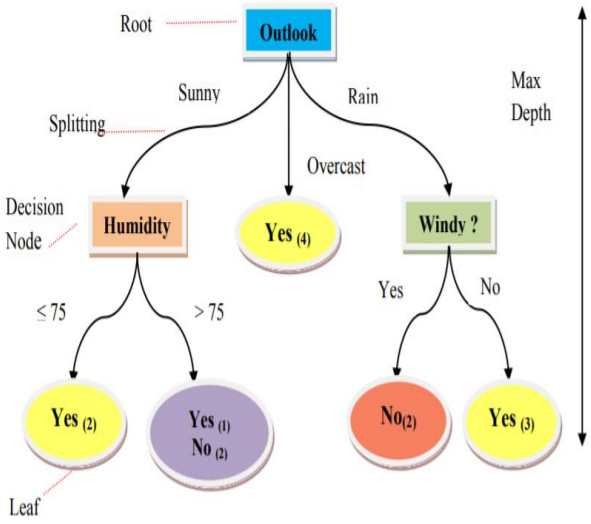


Fig 2. Simple Decision Tree.

Algorithms based on decision trees typically split the data into two parts, one for training and one for evaluation. When building trees, it employs a recursive approach. It builds a trained classifier by repeatedly subdividing the training set.

When the subset at a node has the same predicted value as the target variable, the recursion process is complete. Each "instance" in a training dataset includes some combination of "attributes" and "class label." Any attribute can hold values of any of these three types: ordinal, real, or boolean. A decision tree is depicted as an example in Figure 4.2. At each stage

International Journal of Science, Engineering and Technology

An Open Access Journal

of a decision tree's development, decision nodes are inserted to stand in for a test of certain criteria. Multiple forks off the decision node are generated, one for each test result.

Therefore, the number of forks at a given decision node is equal to the number of possible results from the test at that node. The term "internal node" is used to describe these points of decision (denoted by rectangles). The leaf nodes mark the conclusion of this division (denoted by ovals). In a decision tree, each node is a predicted category. The decision tree classifier builds trees using a greedy approach.

III. SIMULATION PARAMETER

Gradient boosting makes use of three key parameters: weight, center, and spread. Assessment criteria: Typically, the evaluation of a disordered problem is performed on a grid known as a disarray framework, where the number of tests successfully clustered and incorrectly arranged is represented as takes, respectively.

So, the accuracy can be measured according to Eq. 5.1

$$Accurancy = \frac{TN + TP}{TN + TP + FN + FP}$$

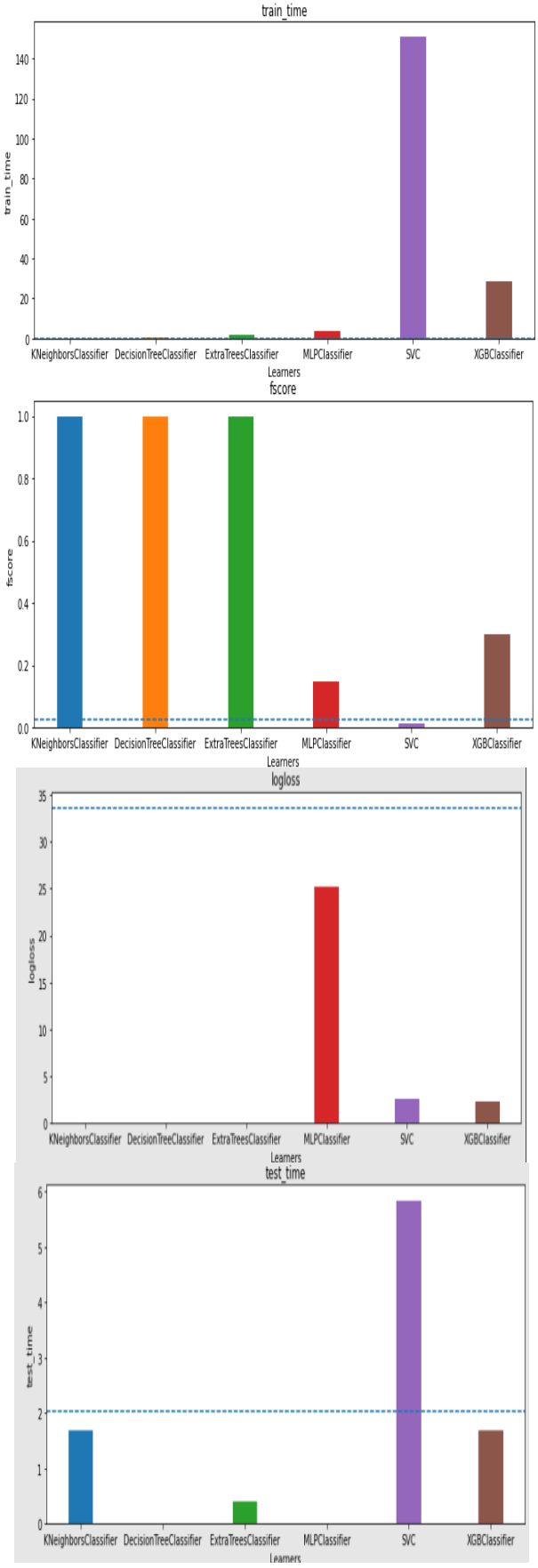
Precision, Sensitivity, Recall, and Specificity are all metrics that can be used to evaluate the success of a solution to a problem with parallel characterisation. Eq. 5.2 and 5.3 provide the formula to find these behaviors.

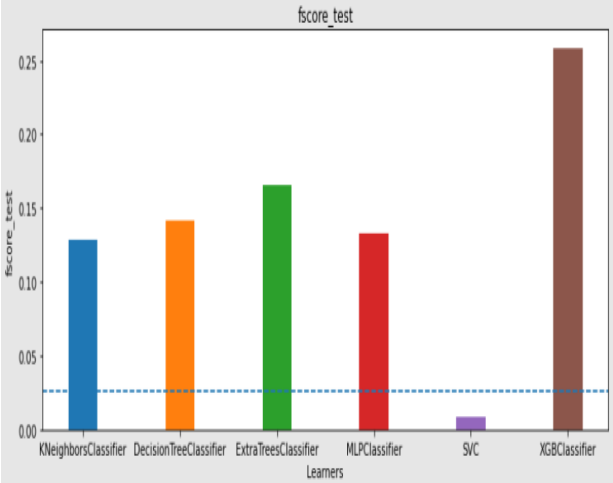
$$\Pr ecision = \frac{TP}{TP + FP}$$

$$\operatorname{Re} call = \frac{TP}{TP + FN}$$

For the recipes in (5.1), (5.2), and (5.3), TP stands for the number of good tests with proper analysis. FP reveals how many valid tests have been incorrectly interpreted. The FN value reflects the number of ineffective but audibly incorrectly evaluated tests. TN has many patient-tested models, and each patient is evaluated with care.

IV. RESULTS





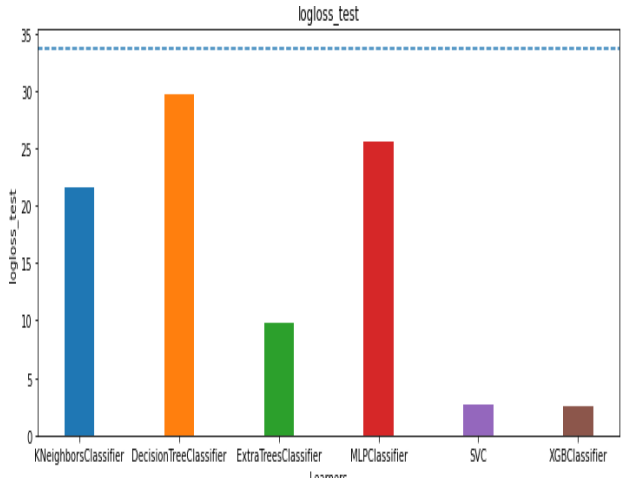


Fig 3.logloss Test

Table 1. Compared Results.

Model	F-score	Log-loss	train time	F-score test	Log-loss test	test time
KNN	0.996	0.004	0.018	0.128	21.584	1.679
Decision Tree	0.996	0.006	0.196	0.1415	29.651	0.007
Extra Tree classifier	0.996	0.006	1.768	0.166	9.823	0.409
MLP-Classifier	0.1469	25.135	3.877	0.133	25.621	0.0178
SVM	0.0129	2.656	150.863	0.0085	2.697	5.84
XG-Boost classifier	0.3002	2.231	28.560	0.2585	2.561	1.687

V. CONCLUSION

In order to determine which category an observation or piece of data belongs to, classification must be performed. To do this, a training set made up of labeled or otherwise known observations is used. Modern machine learning methods are frequently applied to classifying data. Machine learning (ML) is applied to the criminal analysis process to help spot and categorize anomalies. In crime detection, the input features can be a combination of information about the suspect's criminal history and characteristics and information about the crime scene. As a decision support tool, it learns from the input features used during training and employs these learned distinctions between normal and abnormal conditions.

Conventional approaches are only useful for problems of a modest scale and low dimensions. There is a risk of mistake and computing difficulties if these techniques are applied to issues with a large number of dimensions. By projecting it into a high dimensional space, SVM is able to efficiently deal with dimensionality issues. In order for SVM to achieve its full potential, it needs to be able to deal with high-dimensional problems, which kernel techniques are able to do.

REFERENCES

[1] S. A. Chun, V. A. Paturu, S. Yuan, R. Pathak, V. Atluri, and N. R. Adam, "Crime prediction model using deep neural networks," in Proceedings of the 20th Annual International Conference on Digital Government Research, pp. 512–514, Dubai United Arab Emirates, 2019.

[2] B. Sivanagaleela and S. Rajesh, "Crime analysis and prediction using fuzzy c-means algorithm," in 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI), pp. 595–599, Tirunelveli, India, 2019.

[3] U. V. Navalgund and K. Priyadharshini, "Crime intention detection system using deep learning," in 2018 International Conference on Circuits and Systems in Digital Enterprise Technology (ICCSDET), pp. 1–6, Kottayam, India, 2018.

[4] M. Nakib, R. T. Khan, M. S. Hasan, and J. Uddin, "Crime scene prediction by detecting threatening objects using convolutional neural network," in 2018 International Conference on Computer, Communication, Chemical, Material and Electronic Engineering (IC4ME2), pp. 1–4, Rajshahi, Bangladesh, 2018.

[5] S. Ranjan, P. Garhwal, A. Bhan, M. Arora, and A. Mehra, "Framework for image forgery

detection and classification using machine learning," in 2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS), pp. 1872–1877, Tirunelveli, India, 2018.

[6] J. Borges, D. Ziehr, M. Beigl et al., "Feature engineering for crime hotspot detection," in 2017 IEEE Smart World, Ubiquitous Intelligence & Computing, Advanced & Trusted Computed, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (Smart World/SCALCOM/UIC/ATC/CBDCom/IOP/SCI), pp. 1–8, San Francisco, CA, USA, 2017.

[7] S. Yadav, M. Timbadia, A. Yadav, R. Vishwakarma, and N. Yadav, "Crime pattern detection, analysis & prediction," in 2017 International conference of Electronics, Communication and Aerospace Technology (ICECA), vol. 1, pp. 225–230, Coimbatore, India, 2017.

[8] Mukaka, M., White, S. A., Mwapasa, V., Kalilani-Phiri, L., Terlouw, D. J., and Faragher, E. B. (2016). Model choices to obtain adjusted risk difference estimates from a binomial regression model with convergence problems: An assessment of methods of adjusted risk difference estimation. Journal of Medical Statistics and Informatics, 4(1), 5

[9] Budur, E., Lee, S., and Kong, V. S. (2015). Structural Analysis of Criminal Network and Predicting Hidden Links using Machine Learning. arXiv preprint arXiv:1507.05739.