Crime Prediction For Better Response Using KNN, PCA And Random Forest

Uchenna Akujuobi, *King Abdullah University of Science and Technology,*

***Abstract*—Fighting crime is not an easy task. With the quick growth in technological development, People have tried to incorporate technology and crime fighting. Predictive policing has gained attention during Recently, there has been a rise in predictive policing which tries to predict crimes using machine learning algorithms and statistics. Although there have been some works done on predicting future crime and trying to analyze the crime datasets to detect a pattern, this hasnt yet been perfect.**

Most research has been focused on trying to predict future crime and prevent it. This, however, is a great idea and should be applauded, but it cant detect all the crimes. This being the case, there should be a way to alert the police department faster and with less risk and to help the police officers respond effectively by providing them with some information about the crime being committed. In this paper, I present a model designed to predict a crime being committed in real time. This paper also tries to show the difference between some machine learning methods used. The algorithms used in this work include KNN, PCA, Decision tree, K-D tree and random forest.

***Index Terms*—Machine Learning, KNN, PCA, Decision tree, random forest, crime, prediction, SVM, ANN, Simulated Annealing, policing, predictive policing.**

I. Introduction

Crime has always been around and there have been many ways developed to fight crime in cities and communities. Even with all these efforts, it hasnt been possible to completely eradicate crime. Due to technological development, fighting crime has also gone digital and with the emergence of machine learning, the fight has been moved some steps further in progress with the rise of predictive policing. Although there have been many methods to help police respond to crime and conduct investigations more effectively, predicting where and when a crime is likely to occurand who is most likely responsible for prior crimeshas recently gained considerable currency [1].

Currently, there are some works in this area. Most of which are trying to predict future crimes or people likely to commit a crime. This, according to claims, reduced the crime rate in the areas it has been used. This is great, but there is still crime being committed and there should be a way to report and predict these crimes in real time without having to put one's self in harms way and also enable police officers to respond to crime calls ready and prepared to have an idea of what is really happening.

Let us consider two scenarios. The first scenarios: A victim is stuck in a building hiding from some dangerous people and knows that if he makes any sound, that would be the end of the day for him. So, in this situation, he just cant afford to risk his life to make that 911 call without which the police department would never know what is going on in that building. So his only option is to stay in hiding helpless and hoping he wont be found. The second scenario: A police officer on getting a crime alert goes to the crime area without knowing what to expect. And not knowing what to expect might lead to grave situations. For example, not knowing if to call for backup or for a standby ambulance or medical help for the victims. Both of these scenarios are really bad and can lead to grave results and are big problems.

The aim of this work is to provide solutions to these problems by providing a way where the victims can provide crime reports without putting themselves in harms way. This alert would just be just a click on a phone application and providing a word description of whatever situation the person is in or what he/she sees. On getting this report, the model would make a prediction of the type of crime being committed and send an alert to the police department with the predicted crime together with some other possible crimes. With this information, the police officers can now make better decisions on how to approach the crime.

The model in this work was built using the Chicago crime dataset [2], which ranged from 2001 to April 2015, which was when the dataset used was downloaded. This dataset is updated daily. The rest of this paper is organized as follows: section 2 talks about some of the related works in predictive policing, section 3 discusses the algorithms used in this work, giving a brief overview of the algorithms, and then the implementation, section 4 is about results and discussion, section 5 contains the conclusions and section 6 talks about the future works.

II. Related Works

There have been some works done in this area, both in the analysis of crime dataset and the building of predicting models. RAND [1] in its work in Predictive policing looked into the role of crime predicting in law enforcement. They looked deep into the idea of predictive policing, what has been done and what is currently being done. They also looked into the advantages, disadvantages, response, outcome and methods and algorithms currently used by software developers in building a model.

In a related study, Jay Feng in his work, Predicting And Plotting Crime In Seattle[3] analyzed the crime dataset of Seattle. Feng analyzed the false alarm calls and made it easily visible by plotting the places with high false alarm calls on the map. Also, he classified the crimes into 6 levels of urgency and plotting out the crimes with an urgency of 4 and above, he compared it with the false alarm calls and found out that they were way more concentrated in the downtown region.

Predpol, on the other hand developed a system of predictive policing that uses past type, place and time of crimes to predict future place and time a crime is to be committed. This prediction is automatically generated during each shift of the day. However, unlike Predpol, my aim is to make predictions based on real-time information on present crime and to provide information about the type of crime that could be going on in a place in a particular area at a given time.

III. Methodology

The dataset used in this work is the Chicago crime dataset which can be gotten from their database website. As at when this paper was written, this dataset has 22 features out of which seven were selected for this project. The features used in this work are:

- Location Description
- Time
- District
- Ward
- Longitude
- Latitude
- Primary Description (Target class)
- Secondary Description (The last word)

Some of these entries have some missing fields. Due to how critical the work is, the entries with empty fields were discarded. After the data selection, 3,127,449 entries were left. The target feature, which has the different types of recorded crimes in the dataset, comprises of 33 different crimes. We divided the data into two parts, 75% of which was used for training and 25% for training not taking into count the missing entries.

The prediction of the ward and district for the test dataset was made using KNN method and were added to the feature set of the test dataset. The prediction accuracy is 99.86% on predicting the ward using 3 nearest neighbors and an accuracy of 99.97% on predicting the district using the nearest neighbor. Even though this can be gotten from an internet search or from the Chicago city database of districts and wards, this work is built not to depend heavily on the internet so as to avoid issues of poor internet network speed or issues related to the internet and for the prediction accuracy for the district and ward was higher using the KNN search, an addition of new datasets with the ward and districts wasnt going to make much improvement.

Having all these features, PCA was applied to reduce the dimension and also to map the dataset into a new space to get the maximum variance. On applying PCA and mapping the data to the new space, the eigenvalues gotten were can be seen in Table I. In Table I we can see that the first 3 had the highest eigenvalues, thus, the rest could be discarded. Therefore, with the reduced dimension and new space, it would be easier and less expensive to run it with some simpler algorithms as well. The algorithms used in this paper and their results are briefly discussed in subsections below.

A. K Nearest neighbor (KNN)

This method requires three things: The set of stored records, distance Metric to compute distance between records,and the value of k, the number of nearest neighbors to retrieve. To classify a record, the distance is to be calculated from the record to the different records in the data space and the select the k nearest point. Different distance metrics can be used for this, e.g. Euclidean or cosine distance. The distance metric used in this work is Euclidean. After this, the class labels of the k nearest distance will be used to determine the class for this new record by assigning it to the majority class among the k nearest neighbors. This method gave good prediction accuracy when used to predict the crime. The results would be seen in the next section.

B. Artificial Neural Networks (ANN)

This is a powerful algorithm in that it breaks down the problem into bits to learn a given model. For this it has proved very successful in character recognition, image recognition, and handwriting recognition, just to mention a few. Neural computing requires a number of neurons, to be connected together into a neural network [4]. Neurons are arranged in layers. Each neuron takes several inputs and creates one output. At each neuron, every input has an associated weight, which modifies the strength of each input. The neuron simply adds together all the inputs and calculates an output to be passed on [4]. An activation function is used on the hidden layer and output layer. The activation functions can be the sigmoid function, hyperbolic tangent function, Elliot activation function, linear activation functions, etc. This, however, didnt give any good result for this work. Due to a large amount of local minima, it ended up falling into one of the local minima each time and also takes a long time to run. Combined with simulated annealing, didnt help either in getting a better result, but helped reduce the time to get to the local minima.

C. Support vector machine (SVM)

This is another powerful algorithm. A support vector machine constructs a hyperplane or a set of hyperplanes in a high- or infinite-dimensional space, which can be used for classification, regression, or other tasks. Intuitively, a good separation is achieved by the hyperplane that has the largest distance to the nearest training data point of any class (so-called functional margin), since in general the larger the margin the lower the generalization error of the classifier [5]. Training in SVM can be slow but accuracy is high due to their ability to model complex nonlinear decision boundaries (margin maximization) With a huge dataset, it takes weeks to learn a model. For this work, it was cut off after running for five days without completing the model training. It was ran on a system with low processing power, with a stronger system or GPU, it would have trained faster and gotten a result.

D. Decision tree

This is an easy, simple and less computationally intensive algorithm. This also takes less time since the model is ”learned” by splitting the source set into subsets based on an attribute value test. The tree is constructed in a top-down recursive divide-and-conquer manner

Eigenvalue 1	Eig.value 2	Eig.value 3	Eig.value 4	Eig.value 5	Eig.value 6	Eig.value 7
453974929.2467	4967.0609	1766.9420	215.2454	21.6550	0.0041	0.0020

TABLE I
EIGENVALUES GOTTEN FROM RUNNING THE PCA

- At the start, all the training examples are at the root
- Examples are partitioned recursively based on selected attributes
- Attributes are categorical (if continuous-valued, they are discretized in advance)
- Test attributes are selected on the basis of a heuristic or statistical measure (e.g., information gain)

This algorithm gave good prediction results for this work. The results would be seen in the next section.

E. Random forest

In this method, a different subset of the training dataset is selected 2/3, with replacement, to train each tree. Remaining training data (Out Of Bounds) are used to estimate the error and variable importance. The class assignment is made by the number of votes from all of the trees (weak models). This method makes a better prediction than just using one tree. Hundred trees were used in training the random forest in this work.

F. K-D tree algorithm of KNN

This is an implementation of the KNN search. This algorithm partitions a n-by-k data set by recursively splitting n points in K-dimensional space into a binary tree. To get the nearest neighbors of a new record, it restricts the training data space to the training observations in the leaf node that the new record falls into. Each node is called a bucket. This algorithm is faster than the usual KNN search where the distance of the new record has to be compared with all the other points in the data space. Using this method combined with the random forest I was able to catch the wrong prediction made by the random forest method. In this work, this algorithm was used to calculate the possibility of a predicted crime to be another crime.

IV. EVALUATIONS

To show the performance of the algorithms used, and the different performance of some of the algorithms with and without the dimension reduction using PCA, the test dataset was run against the trained model. Table II shows the performance of the different algorithms. From table II, we can notice that for the KNN algorithm, the model trained without applying PCA performed worse than the model trained on the data mapped into the new space using PCA. On the other hand, decision tree model trained on the dataset without PCA performed a bit better than the model trained with the reduced dataset. The difference is actually not much and can be ignored compared with the time it takes and the reduced dataset. The random forest performed better than all other algorithms. This is expected since the random forest was trained on 100 weak classifiers. Fig 1 shows the mean squared error with the different number of trees used for training the random forest.

Algorithms	Accuracy
Out of bag random forest of trees (100)	88.99%
Decision tree (With PCA)	87.96%
Decision tree (Without PCA)	88.55%
KNN (With PCA)	81.47%
KNN (Without PCA)	74.84%

TABLE II
PERFORMANCE OF THE DIFFERENT ALGORITHMS

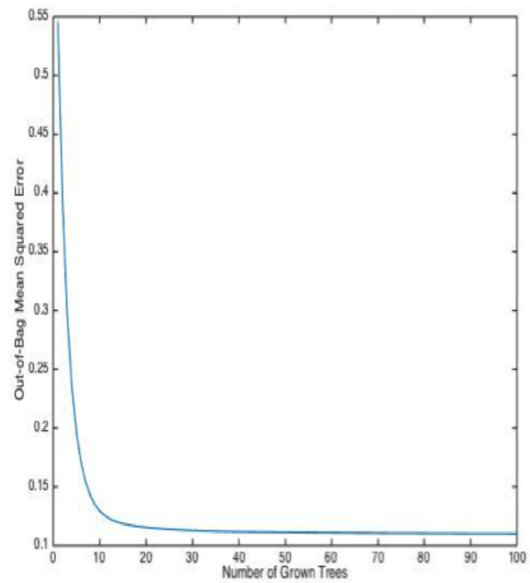


Fig. 1. Mean square error of the random forest

Figure 2 shows the feature ranking of the features used in training the dataset. This is with the dataset mapped into the new space using PCA. This shows that the second feature has more importance than the rest. Looking at the eigenvalues gotten from PCA, we could easily see the relationship with feature 2 being the feature with the highest eigenvalue and feature 1 and 3 being the other two with high eigenvalues. The aim was to get a more accurate prediction so as not to provide wrong information to the officers when trying to respond to crime. For this reason, the possibility of it being a different crime from the predicted crime was calculated to show the possible crimes. This was done using the information gotten from the K-D tree search using a K=10. With this method, all the wrong predictions gotten from the random forest were caught. The actual crime was always among the possible crimes predicted using this method. Figure 3a,b, and c show this. In figure 3a, the real crime committed was robbery while the predicted crime from the random forest was battery.

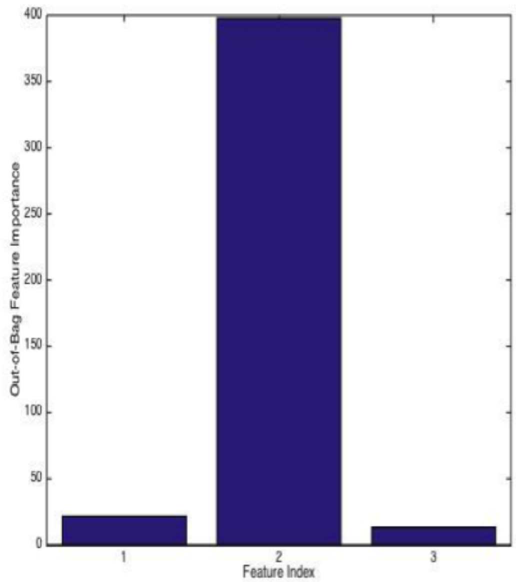
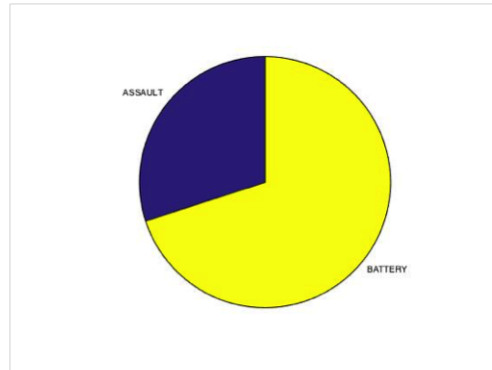
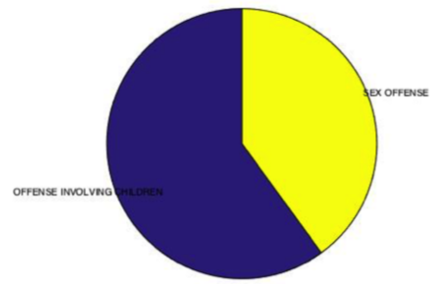


Fig. 2. Feature ranking for random forest

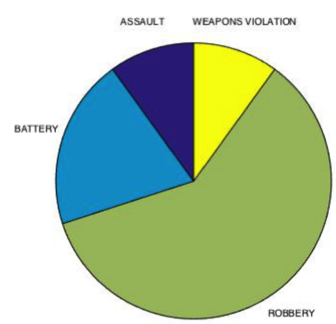
But using this method, we could see that it also predicted the possibility of the crime being robbery. In figure 3b, the real crime was sex offense, but the predicted crime was that it was an offense involving children, which actually could be true for a sex offense could involve children but we cant say this for sure. But using this method, it predicted that there is a possibility that it might also be a sex offense. In figure 3c on the other hand, the real offense was robbery, but the predicted crime was battery and using this method, it actually gave a higher probability for the crime to be robbery while also predicting it could be a battery, assault or weapons violation. Looking at these, there seems to be a relationship between the predicted crime by the random forest, the real crime and also the other possible crimes predicted. For example, in figure 3a, battery can be a result of a robbery, in figure 3b, sex offense might involve children and in figure 3c, battery and assault and weapons violation has some relationship with robbery. Therefore, giving this information, there is no conflict in actions to be taken by the police since if you are prepared for one, you are prepared for the other since they are all related.



(a) image 1



(b) image 2



(c) image 3

Fig. 3. Various edge detection algorithms

V. Conclusion

This is an interesting work, which would help in fast crime response and provide the necessary information that would enable the police to make appropriate decisions when responding to a crime call to reduce casualties. This would also provide victims with the help they need without having them takes some unnecessary risks to make the call for help. This would also provide a means for crime reporting by not just the victims but for people around. After running this different algorithms, we can see that using a combination of random forest and K-D tree for the crime prediction is better since it gives a better idea of what is really going on which will thereby, help in the decision making on how to respond.

VI. Future Work

This work is not yet perfect as such, there are some issues I would like to address in the future. Currently, work is prone to false alarms. It wont be able to detect false alarms. Would be interesting to see the relationship with false alarms. At the moment, The Chicago crime dataset does not have this information on false alarms and so, couldnt analyze it at the time of this paper. Although the proposed algorithm gives good accuracy more optimizations/Tuning could improve the performance and accuracy. With that said, I would like to improve the performance by more tuning on the random forest and bagging.

Another issue is that this model cant detect the exact place in a particular location the crime is being committed for instance, which room in a building. Trying to find a way for it to get this would be great. The aim of this work is to help people and the police, therefore, making a phone application and a backend server that would handle the crime alert would be a way to implement this. Finally, it would be nice to see this put to use in cities.

REFERENCES

[1] W. L.Perry, B. McInnis, C. C Price, S. C. Smith, and J. S. Hollywood. *Predictive POLICING* The Role of Crime Forecasting in Law Enforcement Operations, RAND, 2013.

[2] Crimes - 2001 to Present — City of Chicago — Data Portal *Predictive POLICING* Chicago. City of Chicago, n.d. Web., 17 May 2015.

[3] F. Jay. Predicting and Plotting Crime in Seattle, Web 17 May 2015.

[4] C. M. Bishop. Pattern Recognition and Machine Learning, Springer Science+Business Media, LLC, 2007.

[5] ”Support Vector Machine. *Wikipedia*, Wikimedia Foundation. Web 17 May 2015.