UNIVERSITY OF DHAKA

Crime and Criminal Activity Analysis to Forecast Crime

by

Exam Roll: 16

Registration No: 2016-514-407

Exam Roll: 49

Registration No: 2016-714-423

Bachelor of Science

UNIVERSITY OF DHAKA

Abstract

Crime is the common pressing predicament in any community or nation. Crime is not a random event. Everything is deliberate for the occurrence of crime. For public safety and security, crime prediction plays an important role. Various studies have been done on predicting crime occurrences for a long time and doing many types of research for crime prediction. These predictive techniques will help crime prevention facilitate the effective implementation of police forces and other security forces. In studying the cause and prediction of crime, some researchers have examined usual socio-demographic variables such as age, gender, ethnicity, education level, location, etc. Some other researchers using machine learning approaches concentrated only on variables like location, date, time, etc. However, some criminological hypotheses like routine activity theory, rational choice theory all recommend that weather could significantly impact crime flows and criminal action. If we can use all these variables together that impact our prediction result, we can make drastic improvements in our prediction accuracy. That is why we want to use the power of deep learning. Deep learning algorithms are more powerful than classical machine learning in that it creates mobile solutions through neural networks: that is, layers of neurons. We use Artificial Neural Network(ANN) for predicting the percentage of crime in a particular location. Additionally, a suitably extracted feature selection can lead to the increment of prediction accuracy.

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

Machine learning is [8] a battlefield of computer science that endeavours to train machines how to discover and rehearse without being explicitly programmed. Furthermore, concretely, machine learning is a data analysis method that involves building and adapting models, allowing programs to "learn" through experience. Computational statistics is nearly correlated to machine learning, which converges on making a prognostication using the power of computers. Machine learning includes the development of algorithms that adjust their models to increase their capability to make foresight .

Machine learning has a subfield called Deep Learning [7] which is involved with algorithms stimulated by the formation and gathering of the brain named artificial neural networks. It is a machine learning technique that assembles artificial neural networks to simulate the formation and purpose of the human brain. Artificial neural networks are like the human brain, with neuron joints interconnected like a network. The human mind has hundreds of billions of organisms named neurons. Each neuron is manufactured up of a cell group accountable for concocting knowledge by carrying information towards the inputs and away from the outputs from the brain.

An artificial neural network has billions of artificial neurons called processing complements interconnected by links. These processing complements are formed up of

input and output blocks. The input blocks accept numerous methods and arrangements of information based on an inherent weighting scheme—the neural network endeavours to learn regarding the knowledge offered to compose one output statement. Like individuals require commands and guidelines to produce a result or output, the artificial neural network also uses a set of learning rules named back-propagation, an abstraction for backward propagation of error, to improve their output results.

Now, crime is one of the numerous earnest obstacles in our community. The analysis and prediction of crimes have been the subject of many studies. Crime investigators investigate crime statements, checking statements, and police request assistance to recognise emerging exemplars, group, and bearings as instantly as possible. They examine these aspects for all applicable circumstances, prophesy or determine future incidents, and issue periodicals, articles, and signals to their agencies. However, some criminological theories intimate that weather [10] [11] can significantly influence crime rates and criminal behaviour. Other studies also say that [18] education can also influence crime incidents. Crime is unpredictable and mostly unstoppable. Crime commentary has been described in numerous diverse forms by both academics and practitioners. These representations include the following:

From Emig, Heck & Kravitz, [21] “Crime analysis introduces methodical, systematic, analytical processes that provide timely, relevant erudition about crime examplers and crime trend correlations. It is fundamentally a tactical tool. Patrol records and criminal records give data on crime scenes, arguments, modus operandi, stolen or getaway agencies, and defendants. Investigating and analyzing data on file with current circumstances can give patrol officers important information on pursuits in their best cities. This encompasses strengthening crime patterns, stolen property specifications, and suspect individualities. Using this knowledge, patrols can adequately extend resources.”

From Gottlieb, Arenberg, & Singh,[12] ”Crime Analysis is a collection of well-organized, penetrating methods focused on presenting up-to-date and pertinent erudition relevant to crime exemplars and trend correlations to support the operational and supervisory staff in preparation the deployment of resources for the

repression and destruction of unlawful activities, supporting the investigative process, and developing misgivings and the withdrawal of proceedings. Within these circumstances, Crime Analysis recommends several administration gatherings, including safeguarding deployment, special agencies, and tactical units, researches, preparation and experimentation, crime repression, and governmental assistance (budgeting and program preparation)."

From Boba, [12]"Crime analysis orderly analyzes crime and dysfunction predicaments and different police-related predicaments – including sociodemographic, spatial, and temporal circumstances – to attend the police in unlawful agitation, corruption and dysfunction decrease, crime repression, and evaluation."

At first sight, there may be no relationship between these definitions. Still, we can see that the key segments are quite related: crime analysis is a systematic process that supports the aims and objectives of a law enforcement agency through the use of qualitative and quantitative methodologies.

1.1 Motivations

Prevention is better than cure. It is better to refrain from committing a crime than investigate what may or may not have happened. Bangladesh is one of the most populated countries, with more than 160 million people. Every day, criminal events occurred in so many places. Comparatively, Bangladesh is relatively a poor country. It has limited resources. The number of police forces or other forces who are engaged in preventing crimes is also limited. If we can help the government or police force to utilize this limited resource properly, it will be a great help to the country to reduce the criminal activities. Ensuring so many people in our country is impossible with the limited number of police forces and other forces. If we can reach every citizen of our country that these places are not safe or safe with real-time updates, they could avoid these unsecured places. Another thing is, modern researches have revealed that crime prognostication is nearly linked to the sustainable improvement of the public and the status of citizen's behaviour. Consequently, there is a growing and essential desire for genuine crime prognostication.

1.2 Objectives

Prevention is the best cure. What offence could be performed than to examine what happened and prevent it from being the best? It is like kids are immunized to anticipate the attack. It has become consequential to have an immune system that blocks crime events in today's environment with such a high crime rate and brutal crime. Nowadays, crime has become a common issue in every society and country. Although violations could happen universally, it is obvious that offenders work on crime possibilities they encounter in the most common spaces for them. Consequently, our intended suspension can probably benefit souls staying away from the neighbourhoods at a particular moment of the day, simultaneously saving lives. Here are some objectives of our work.

- Identifying the crime hotspot of a certain country
- Identifying the type of crime that might occur next
- Helping the police forces to identify the pattern of activities of the criminals
- Improving the police or other force's efficiency by proper distribution of forces
- Reducing the financial loss of a country
- Identifying selves that possibly will be connected in the act of misconduct – either as a sufferer of an offender
- Analyzing the concentrations and deficiencies of the current research and projections
- Classify and analyze research and other paper that spouts the future forecast of crime

To minimize all these problems, we want to present an offering to circumscribe the common illegal hotspot and determine the nature, neighbourhood and participation of authorized offences. We expect to boost people's consciousness concerning the hazardous neighbourhoods in certain periods.

1.3 Contributions

Our main participation is to solve the challenge of prognosticating offence and unlawful hotspots using real-world datasets of violations. To extend this, our contributions are as follows -

- We have tried to determine the several acceptable crime neighbourhoods and their common appearance rate.
- We tried prognosticates what variety of crime might happen next by a particular group in a particular place within a singular moment.
- We become expected to produce investigation research by consolidating our conclusions of a particular offences dataset with its demographics data.

1.4 Challenges

The expeditious and proper classification of unlawful action is predominant to achieving any roots. The common wearisome piece of crime prediction is a hi-tech discovery that is nevertheless in its nascent degrees. It has obtained in the UK, USA, China and some portions of Europe. Nevertheless, the consequences ought not been reliable!. Algorithms are predisposed to imprecision too. For the inconsistent crime-related data, we cannot prognosticate crime models accurately. Certain conclusions ultimately guide poor crime forecasting. Specialists are currently of the prospect that imminent patrolling is not a go-to crime restraint system yet. It should be manipulated with real-time individual interference that begins with crime prevention at the inhabitants level itself. Here are some challenges that are faced by us during our project.

- Lack of enough quality data
- Appropriate feature selection

- Improving of an existing work
- Parameter tuning

1.5 Organization

The remainder of this book is governed as follows. Study of related works, Background and preliminary concepts, Problems of existing solution, are discussed in Chapter 2. Then, our proposed solution is illustrated in Chapter 3. Chapter 4 contains the implementation of our project and in chapter 5, we analyzed the result of our solution.

Chapter 2

Literature Review

In this chapter, we discuss introductory concepts and definitions of some terms in section 2.1 which will be helpful to understand our proposed solution in Chapter 3. Then, reviews of related works - their contributions and limitations are given in section 2.2. In section 2.3, we discuss the existing problems, which motivates us to solve this problem a little bit differently.

2.1 Background Study

In this section, we define some terminologies like Deep Learning, Artificial Neural Network, Transfer Learning, Clustering, Activation Function etc. From these terminologies, anyone can get the intuition of our work and the complete idea of our proposed methods.

2.1.1 Data Preprocessing

Dataset is nothing but a set of objects. Those objects include points, patterns, events, vectors, records, entities, samples, population, cases, observations and so on. They are usually addressed by a number. That number holds the characteristics of

that object. Such as, the weight of a person, speed of a moving object, dimension of a matrix, altitude of an airplane etc.

But all those data are not persable by a machine. Machine only parse some simple kind of data. For that purpose data preprocessing is important. While data processing, all those data gets transformed in such a way that it can be persable by a machine. And that paves the way to apply algorithm by machines easily on those data. Data preprocessing has so many steps. All those steps are not necessary for all the dataset. So those steps needs to be taken by observing the dataset first. Usually a dataset takes only a few steps to be processed perfectly. Following are the steps of the data preprocessing.

- Data Quality Assessment
- Feature Aggregation
- Feature Sampling
- Dimension Reduction
- Feature Encoding

2.1.1.1 Data Quality Assessment

Usually, data is collected from multiple sources. The sources may not be completely reliable. They also may not contain same structure or format. It is so important to ensure the quality data while working with machine learning algorithms. It is understandable that, all those data from different sources will not be perfect because of human errors, errors in measurement or errors in the process of the collection. There are some methods to decrease the effect of these errors in data.

1. **Missing values:** It is realistic to have some values missing in a dataset. Existence of missing values may happen during the collection of the data or some rules of validation. But it is important to keep those missing values in account.

- **Eliminate rows with missing data:** This is the easiest strategy. But this fails when the missing values are outnumbered. If any feature has a large number of missing values, the whole feature can also be deleted.
- **Estimate missing values:** When only a few missing values are in a feature, then they can be predicted using some simple interpolation methods. But the most common solution can be just replacing those values by the mean , mode or median of that particular feature.

2. **Inconsistent values:** Data can also have values which are not consistent. Such as, the 'father name' field contain the value of the field 'mother name'. This type of errors can be human errors. Detecting this type of errors is so tricky and sometimes impossible. So the reliable source plays an important role here.

3. **Duplicate values:** A dataset may have some data objects which are the repetition of some previous data objects. It can be happened while submitting any google form multiple times or so. So in this type of errors we have to deal with these type of duplicate values. the easiest solution is to delete those duplicate objects. Because having those duplicate object, may cause a biased result as output. Because same values always affects any algorithm's learning process.

2.1.1.2 Feature Aggregation

The process of aggregation of features takes the aggregated values so that it can hold the data in a better way. Think of a run scored by a batsman in an single innings. It may not always reflect the potential of a batsman. Or only looking at the the home runs do not reflect his ability properly. In that case an an overall collection of data helps us to visualize the proper fact. Following are some advantages of feature aggregation.

- This causes less processing time and less memory consumption.

- This provides us a view that helps to visualize a high level data. Because the behaviour of a population is always more stable than a bad sample or a unit.

2.1.1.3 Feature Sampling

A common method to select a subset of the dataset is sampling. When the dataset is huge, it is not feasible to work on the whole dataset. Because the computation time, the memory consumption becomes too high, which is not realistic at all. Running any algorithm over a good sample of the dataset is the solution of this problem. in this process the size of the dataset becomes computational worthy, though the accuracy or the performance of the algorithm remains almost same. The thing to keep in mind that the sampling should be done in such a way that the sample dataset should also have all the properties of the old dataset. Otherwise sampling has no meaning at all. A sample dataset is mainly a representative dataset which represent the contents of the original dataset. To do so, it is important to have a good choice of correct sample size. Selecting the right method to sample is also a thing to keep in the mind. Simple random sampling can be a good way to have samples because in this process it is ensured that every entity has an equal probability to be selected. There are two variations of this process.

- **Sampling without Replacement:**

  When an item is picked, the item is removed from the set of the items of that particular dataset.

- **Sampling with Replacement:**

  When an item is picked, the item is not removed from the set. It can be picked more than once.

Though Simple Random Sampling is a very good way to sample but there is limitations. If a dataset contains such objects that varies a lot in the ratio, then this technique do not perform so well. Then it fails to give a well distribution of every type of objects. It mainly occurs when the dataset is imbalanced. This can be

solved using a small trick. We can predefine some objects in the sample. Then we will use this method to pick the rest of the samples. Thus the imbbalance between the dataset can be vanished.

2.1.1.4 Dimension Reduction

Every real world dataset has a very large number of features. Here we can consider every feature as a dimension. We can guess what this method does only by its name. But doing so in a random way may cause some serious problems. Moreover just sampling on those features also do not do any good. It must have an way to pick those feature which do not have much effectiveness over the result or output. Morever the more the dimension of a dataset, the more harder it becomes to visualize those data. Without visualizing it becomes impossible to make a correct observation. The more the dimension of a dataset becomes, the more the complexity becomes to compute or process those data.

The Dimensionality Curse

We can simply guess the effect of large dimension by simply reading he title. The more the dimension gets larger, the more the process of analysis becomes tougher. This complexes the sparsity of the data, thus make it impossible to visualize and harder to model. We all have gone through the linear algebra, how it projects a higher dimensional space over a lower dimensional space. But there is many way of making that projection. Simply rotating the object makes the projection different that the previous one. So when we reduce the dimension, we have to figure out that that projection must be performed in a way such that, it loses as less property as it could. But here, it doesn't happen like this. Here a higher dimensional space is mapped over a lower dimensional space, and those spaces are feature space. There are many good way to perform this task. Singular value Decomposition and Principal Component Analysis are the two of them.

Following are some major benefits of dimensionality reduction.

- When the dimension of the dataset becomes lower, the analysis algorithm works better than before. Because the less the dimension, the less the irrelevant features.
- The less the dimension, it becomes easy to understand an visualize the data.

2.1.1.5 Feature Encoding

Feature encoding includes the transformation on the data in such a way that data can be parsed by the machine but the data must not lose its own property. IT just change the values in some other form that the machine can understand them and the algorithms can run over them. It again reminds us about the goal of data processing what we have already learnt before. There are some rules to perform this task of encoding.

Following are the rules for numeric variables.

- **Interval:** Mathematical functions like $y = a * x + b$, where a and b are constants. For example, Kelvin and Celsius scales, whose unit are equidistant but there values are different. In this case the value of a is 1 and b is 273.
- **Ratio:** Each variable can be moved to any particular scale. Suppose we want to measure the distance between two places. We can simply change the matric of measuring and all the values get changed. We can simply measure them with respect to anything maintaining the equation $y = a * x$, where a is a constant.

Following are the rules for continuous variables:

- **Nominal:** Mapping one to one values can be done. We can convert all the 'Yes' to 1, and 'No. to 0.
- **Ordinal:** A change of values where the order is preserved. The different kind of values can be represented with the help of a new function like, $newValue = f(oldValue)$

2.1.2 Artificial Neural Network

Artificial Neural Network, known as ANN, is an computation system which is known for its efficiency. The main theme of ANN is taken from the biological neural network of human body. It is also familiar with the name of Parallel Distributed Processing System. It is mainly a huge group of neurons which are connected between themselves to allow the communication between those neurons. These single units can be processed in parallel. Here each unit is communication with every other unit using a link between them. Here, each link has its own weight in the network which refers to the info about the previous signals. And this little information becomes the base of solving a problem. Each neuron has its own characteristics. This character is activation signal. A neural network is made from layers. There we can see mainly 3 type of layers:

- **Input Layer:**

  This layer take the info from the input. These input data are passed to the network by the nodes of this layer. Here no types of calculation is performed. They just only pass the information to the hidden layer.

- **Hidden Layer:**

  Each node of hidden layer is connected to the every node of the input layer. They get all the information which is received by the input layer. The do the all sort of calculation to pass it to the output layer. There maybe more than one hidden layer. In that case every node of one hidden layer will be connected to the every other nodes of the another hidden layer.

- **Output Layer:**

  This layer show us the output which is calculated by the network.

In figure 2.1, the first layer of circles represent the input layer. For the further purpose let them mark as X. The next two levels of circles represent the hidden layers. As we already know, every node of the hidden layer are also activation

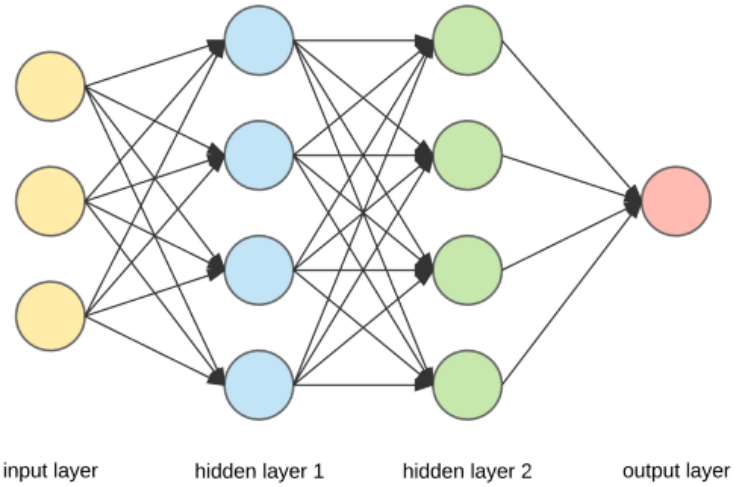


FIGURE 2.1: Neural Network Architecture

nodes. For the further purpose let them mark as W. That single circle is the output layer. It is possible to have more than 1 nodes in the output layer.

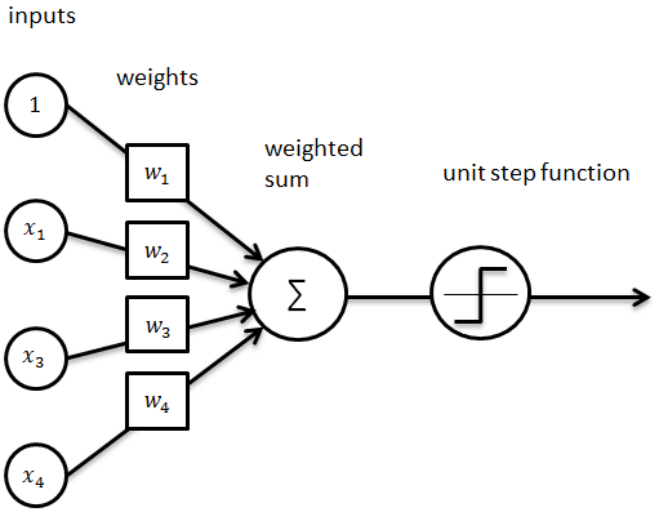


FIGURE 2.2: Nodes of Neural Network

In a Neural Network architecture, each node from a layer is connected with the each node of the next layer.The nodes are connected via a link, where each link have its own weight. That value of weight represents as an impact of that node. That impact means the impact of that node over the next layer of nodes.If we try to visualize this, lets make an example. Consider any node from the second layer of Figure 2.2. We can see that all the input layer's nodes are linked with it. The links represent the impacts which it has over the next layer of nodes. If we multiply

the input values of the input layer nodes with their corresponding weights and sum them up, then we will get the value of a second layer node. As each of the hidden layer node as a character of activation. This will depend on the summed value of those previous layer nodes. It will decide whether a hidden layer node will be activated or not. There is an additional node in each hidden layer. That node is called a bias node, which is not connected to any previous layer nodes.

There is many methods to choose the number of hidden layer and the number of hidden layer nodes. Choosing a perfect number will help to reduce ovverfitting and underfitting. Lets consider the number of input layer node is x, and the number of hidden layer node is and the number of output layer node is z. Then following are some norms for choosing the number of nodes in a hidden layer.

- $x \geq y \geq z$
- $y = \frac{2}{3}x + z$
- $2 \times x \geq y$

Lets dive deep into the equations of the neural networks and how they work. Consider a simple Neural Network architecture having four input nodes along with one bias, one hidden layer with four nodes along with one bias and one output layer node. Consider the input node as X, and the hidden node as A. Thus we can mark the two bias nodes as x_0 and a_0 and the other nodes numbering as these two.

$$X = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix} \qquad A = \begin{bmatrix} a_0^{(2)} \\ a_1^{(2)} \\ a_2^{(2)} \\ a_3^{(2)} \end{bmatrix}$$

Let mark the weights, arrows in the picture as W. Here, the links between the input layer nodes and hidden layer nodes will represent a 3×4 matrix. The links between the hidden layer nodes and the output layer node will represent 1×4

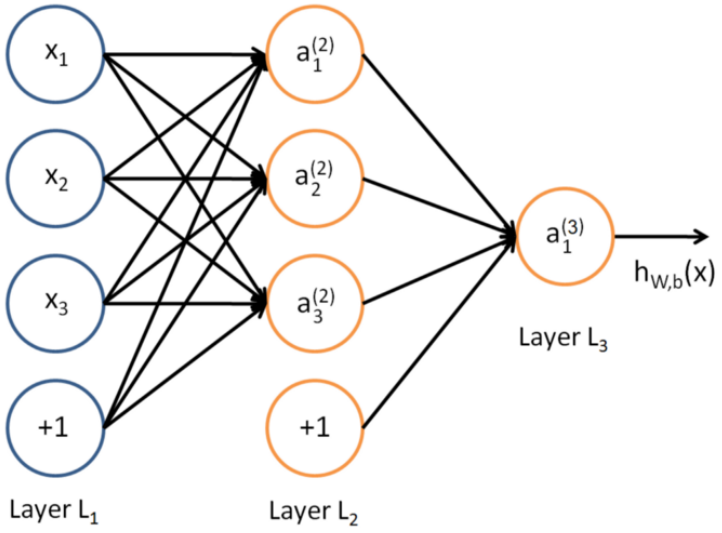


FIGURE 2.3: Simple Neural Network

matrix. Therefore, if the network has a unit of node in layer j and b unit of nodes in layer j+1, then W_j will be of dimension $b \times (a+1)$.

$$\theta^{(1)} = \begin{bmatrix} W_{10} & W_{11} & W_{12} & W_{13} \\ W_{20} & W_{21} & W_{22} & W_{23} \\ W_{30} & W_{31} & W_{32} & W_{33} \end{bmatrix}$$

If we want to do the computation of the activation for the nodes of the hidden layer, then we have to multiply X and W. Then with the product of those two we have to apply the activation function g.

$$a_1^{(2)} = g(W_{10}^{(1)}x_0 + W_{11}^{(1)}x_1 + W_{12}^{(1)}x_2 + W_{13}^{(1)}x_3)$$

$$a_2^{(2)} = g(W_{20}^{(1)}x_0 + W_{21}^{(1)}x_1 + W_{22}^{(1)}x_2 + W_{23}^{(1)}x_3)$$

$$a_3^{(2)} = g(W_{30}^{(1)}x_0 + W_{31}^{(1)}x_1 + W_{32}^{(1)}x_2 + W_{33}^{(1)}x_3)$$

If we multiply A with W for the second layer we will get this.

$$h_W(x) = a_1^{(3)} = g(W_{10}^{(2)}a_0^{(2)} + W_{11}^{(2)}a_1^{(2)} + W_{12}^{(2)}a_2^{(2)} + W_{13}^{(2)}a_3^{(2)})$$

This is just an example considering 4 input layer nodes with 1 layer of 4 hidden layer nodes and a single output node. If we do not give those constraints considering L layers with n nodes and L-1 layers with m node and try to make a generalized equation, the equation will look like this.

$$a_n^L = [\sigma(\sum_m W_{nm}^L[\dots[\sigma(\sum_j W_{kj}^2[\sigma(\sum_i W_{ji}^1 x_i + b_j^1)] + b_k^2)]\dots]_m + b_n^L)]_n$$

Activation functions

It is the activation function which takes the decision of activating any neuron. To make the decision activation functions take the weighted sum and bias as its input and calculate a value which triggers the activation based on some criteria. After that we will get a output. But the output may contain errors. Base on the error value we have to update the value of weights in the neuron and also the value of bias. That process of update is called back propagation. It is the activation function which makes back propagation possible because it continuously calculates the weights and based on the value of weights the back propagation also continues. If a neural network do not have any activation function, then it would become alike linear regression. Neural network can do non-linear transformation with the help of the activation of the nodes. By this it can adapt itself for the complex jobs. And the learning process also goes with this. There are a lot of variety in activation functions.

1. **Linear Function:**

   - They have the equation like a straight line, $A(x) = cx$
   - In case of multiple layers, if all the layers are linear, then the final activation function just will be a linear function of the first layer. Here the value of $'c'$ in the previous equation will be different.
   - The value can be in range from minus infinity to plus infinity.

- Mostly used in the output layer.
- Introducing non-linearity by force is worthless. Because if we differentiate the equation, it will become just a constant, so nothing new.

2. **Sigmoid Function:**
   - The equation of a sigmoid function looks like this: $A(x) = 1/(1 + e^{-x})$
   - Nonlinear in character. And very sensitive. Slight changes in the value of X can make a huge change in the value of A.
   - The output value can be in the range from 0 to +1.
   - Mostly it is used in the output layer where the output values are binary.
3. **Hyperbolic Function:** It is a better version of the sigmoid function. Most of the times it works way better than sigmoid.
   - The equation looks like this. $A(x) = 2/(1 + e^{-2x}) - 1$
   - The output value can be in the range form -1 to +1
   - This is nonlinear in character.
   - Commonly used in the hidden layers.
4. **ReLU:** The full form of RELU is Rectified Linear Unit. ReLU is the most used activation function which is also easy to calculate.
   - The equation looks like this. $A(x) = max(0, x)$
   - The output value can be in the range of zero to positive infinity.
   - It is nonlinear in character. This makes it easy to calculate the value while calculate the derivative value in backpropagattion.
   - Easy to compute. Save computational time than the other activation functions.

   In simple words, RELU learns much faster than sigmoid and Tanh Function.
5. **Softmax Function:** This is also a kind of a sigmoid function.

- Nonlinear in character.
- Mostly becomes handy while we have to deal with more than one classes.
- It is used in the output layer because of its probable characteristics.

Choosing the Right Activation Function:

It has become a norm that no matter what the problem is, we have to use ReLU as an activation function. Because it never produces any terrible results. Moreover, ReLU is linear in character, which allows it to compute faster and learn at a good rate. But there are some special case, where some other activation function works better than ReLU.If the output layer holds a binary value then the sigmoid function should be the go-to choice.

Cost Function:

The cost function generate the value of overall error. I does so by calculating the distance between the original and the guessed value. Following is the equation of a generalized cost function.

$$J(W) = \frac{1}{m}\sum_{i=1}^{m} Cost(h_W(x^{(i)}), y^{(i)})$$

Forward Propagation Calculation:

$$h_W(x) = a_1^{(3)} = g(W_{10}^{(2)}a_0^{(2)} + W_{11}^{(2)}a_1^{(2)} + W_{12}^{(2)}a_2^{(2)} + W_{13}^{(2)}a_3^{(2)})$$

Forward propagation does the task of calculating the output value for the output nodes. The process it follows and equations it uses are described previously. The above formula is used to calculate the value h(x). When the value of h(x) is calculated, we need to calculate the cost. For that the previously described equation of cost function is used which calculates the cost for the passed input values. Following

are the formulas to describe forward. And after the final forward propagation an final output value is generated.

$$x = a^{(1)} \tag{1}$$

$$z^{(j+1)} = W^{(j)} a^{(j)} \tag{2}$$

$$a^{(j+1)} = \sigma(z^{(j+1)}) \tag{3}$$

$$h_W(x) = a^{(L)} = \sigma(z^{(L)}) \tag{4}$$

That's how forward propagation works and how a Neural Network generates the predictions.

Backpropagation Algorithm

The main purpose of backpropagation is making value of cost function $J(W)$ relatively small. When we get the values of the weights perfectly the cost function generate the smallest value. In backpropagation, partial derivative of $J(W)$ is calculated. The main goal is to determine, what happens to the network when a particular value of weight is changed. The derivative of a function on each variable tells us about the sensitivity of the function.

$$W_j := W_j - \alpha \frac{\partial}{\partial W_j} J(W) \tag{5}$$

Then Gradient Descent algorithm is used to calculate the weight values to minimize the cost function , and we use partial derivative in that process. If we make a summary about backpropagation, it have a total five steps.

1. For the train set, define a(1) = X.
2. Run forward propagation process.
3. Calculate the value of error δ using y.

4. Calculate the value of errors values for other layers.

5. Get the value of derivatives for all the layers.

So backpropagation is mainly calculating the error backwards int eh network. In this process, the network gets to know which neuron is contributing how much on the value of output. The neuron which is contributing more to the overall error have relatively bigger derivative value. Making this a base the error value minimizes in the whole system slowly step by step in each iteration.

2.1.3 Principal Component Analysis

Principal Component Analysis, also known as PCA is one of the mostly used techniques to reduce the dimension of a huge dataset. It is a unsupervised method which also does some other things like comparison of information, de-noising of data, compression of impression and so. It has some give and take. Reducing the dimension is not free of cost. The main cost is accuracy. The goal is to make the huge dataset simple. It also make it easy of visualization and a lot easier of explore. As it reduces the dimension, it reduces the complexity of a running model. That's how the model can perform faster. We always have to take the decision of take over give because the cost of less complicated data costs from the accuracy. So it is important to decide how much we want to reduce the dimension.

The main goal of PCA is to search for components which are dominating over others. They are called principal components, which are vectors. Those dominating factors are not chosen randomly. The first component has to be calculated. The next principal components are simply perpendicular on the previous ones. Following images shows the idea of PCA. Normally PCA can be used separately. PCA can do the work of cleaning the data and preprocess it. The preprocessed data can be used in any machine learning algorithm which has a less complexity.

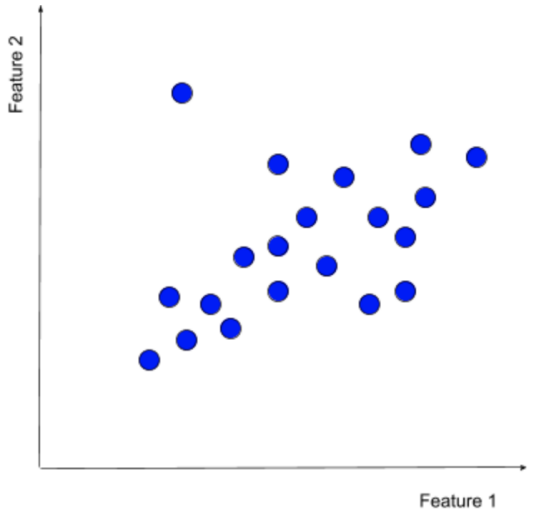


FIGURE 2.4: A simple 2-dimensional dataset

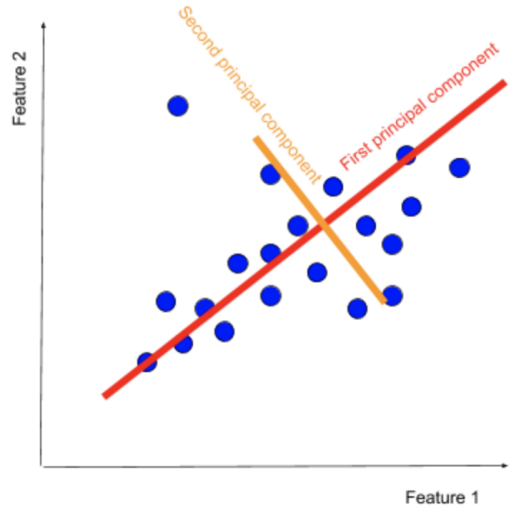


FIGURE 2.5: Dataset marked with dimensions

2.1.4 Deep Learning

Machine learning has a subfield called Deep Learning [7] which is involved with algorithms stimulated by the formation and gathering of the brain named artificial neural networks.Learning can be supervised, semi-supervised or unsupervised. Andrew Ng who is the founder of Coursera and one of the Chief Scientist at Baidu Research also founder of Google Brain explained the idea of [7] deep learning as:

"Using brain simulations, hope to:

– Make learning algorithms much better and easier to use.

– Make revolutionary advances in machine learning and AI.

I believe this is our best shot at progress towards real AI”

The essence of deep learning according to Andrew is that we presently have speedy machines and sufficient data to actually qualify comprehensive neural networks.

2.1.5 Transfer Learning

Transfer learning is the method of using a pre-trained model [25] on a new problem for forecasting or predicting something. When there is a lack of data in a specific field and a relationship between two fields, anyone can transfer one pre-trained model into another. It is worthwhile in deep learning because it can encourage deep neural networks with approximately small data, and most utmost natural problems typically do not have millions of specified data objects to enlighten such complicated models. Transfer learning is the advancement of studying a new task within

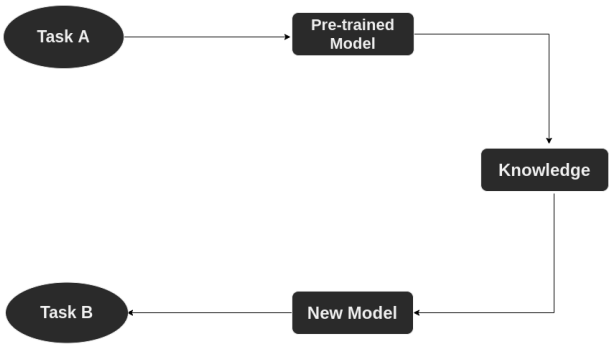
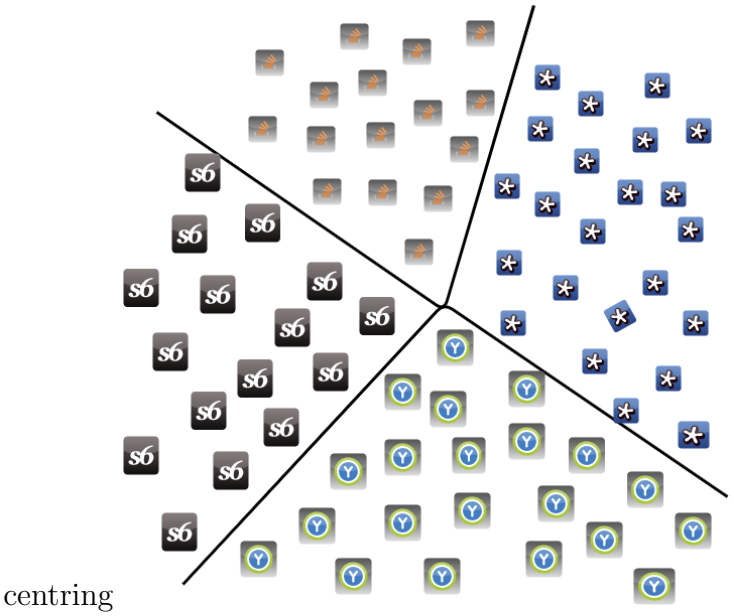


FIGURE 2.6: Transfer Learning Flow

the instructions of information from a similar task previously learned. The generic concept is to apply the information a model has received from a job with many possible labelled training data in a new task outwardly significant data. Alternatively, starting the training method from scratch, we spring with originals learned from resolving a similar task.

2.1.6 Clustering

Clustering or Cluster analysis is a sort of unsupervised learning method. An unsupervised learning method is a technique in which we express endorsements from datasets consisting of information data outwardly specified acknowledgements. It is frequently applied as a data investigation technique for determining attractive patterns in data, such as collections of shoppers based on their performance. Clustering



centring

FIGURE 2.7: Clustering System

is the responsibility of separating the group or data objects into several groups [5] such that data objects in the same groups are exceeding related to other data objects in the corresponding group and disparate to the data objects in other groups. It is a compilation of objects based on relationship and divergence among them.

2.2 Related Works

Criminal movements are every day throughout the globe. Consequently, researchers have performed numerous activities on this topic matter. In this section, we casually preface the work relevant to our study.

2.2.1 DNN and Feature level data fusion

In the writing [15], they introduce a feature-level data coalition system including environmental circumstances based on a deep neural network. They train their DNN, consisting of four layers: spatial, temporal, environmental context, and joint feature representation layers. They apply a deep neural network with feature-level knowledge fusion, a crime event prognostication method reflecting environmental context knowledge using multi-modal data fusion. They dissected the achievement of their prediction model by contrasting it with SVM and KDE. Later they estimated the completion of the prognostication model by calculating the accuracy, precision, recall, and area under the curve. They discovered that their DNN-based multi-modal data fusion technique is a more suitable prognosticating crime phenomenon than the previously proposed direct concatenation of the probabilistic method, both of which exhibited low competence.

2.2.2 CNN and RNN

The intention of this paper [23] is to catch the power of deep neural networks to make following day crime count prognostications. Paper, as mentioned earlier, includes three principal contributions. First of all, they are the earliest to use a joint recurrent and convolutional neural network to prognosticate crime. They said they could combine crime data with additional weather, public transportation, and census data. Three, they said they are prime to work to perform prognostications for the following day. While prognostications at the recurrent and hebdomadal level do serve a purpose, becoming prognostications every day would authorise police officers to be much more rigid with their efforts. They apply four diverse types of neural networks to anticipate crime.

1. Feed Forward Neural Network(FFN)
2. Convolutional Neural Network (CNN)
3. Recurrent Neural Network (RNN)

4. Recurrent Convolutional Neural Network(CNN+RNN)

This scheme examines the practicability of using machine learning methods, particularly neural networks, to prognosticate unlawful conduct based on the history of arrest bookings. They wanted to show that deep learning techniques could be used to classify crime and criminal behaviour analysis. The FFN(Feed Forward Network) has the most moderate accuracy, and the RNN+CNN has the most demanding accuracy.

2.2.3 Complex Neural Network and Graph

The offence that occurs on certain days has the following key attributes: crimes infrequently recurring, crimes resulting due to unusual extra motion and appearance of crimes pre-indicated by some other erudition. In this paper [4] they used a hybrid approach of deep learning. Have modern researches conferred a greater rate of brutality against ladies transpiring in a spot whither? There is, and a tremendous level of porn subscribers or the users of sexual services is High. Another investigation found that violence in India against women and terrorist attacks leads to get more high throughout the last four month of the year. There are many approaches to consider when predicting crime; each approach has its distinct parameter scope. This conveys a provocation to a kind of data that arrives in crime prognostication. The algorithm used in this paper has a deep learning model that uses graphs to interpret the knowledge. Typically this can be recognised as a complicated neural network wherever one neuron may correlate to another if there is a connection among knowledge. So fundamentally, it is a mixed-method to crime event prognostication using deep learning. It is reasonable to prognosticate crime by concentrating on designs and courses from various contributing factors. Each of the procedures adopted in crime prognostication has its boundaries and preferences; it was evident that prognosticating should consider all the potential factors influencing that prediction.

2.2.4 VGGNet-19(19 Layer Deep CNN) and RCNN

CCTV's are extensively practised to restrain violations in the surroundings. Nevertheless, no growth in the administration of crimes because CCTV needs personal surveillance, leading to human errors. To defeat this problem, they [20] polished up a system that identifies crime in real-time videos, images and informs the personal administrator to attain the imminent operations. To inform the administrators or nearby police headquarters about the crime phenomenon, they attached an SMS forwarding module to their system, which conveys SMS to concerned character whenever violations are exposed. They used Pre-trained deep learning model, VGGNet-19. VGG-19 is 19 layers deep convolutional neural network. They found that in terms of training accuracy, VGG19 performs well. They also used Fast RCNN, which describes the bounding box over an object in photographs like a person, gun, knife, and untrained images. It was invented to solve ATM, Bank and Public Places enigmas by discovering the evidence in the instruction of characters used to detect the violations before it occurs.

2.2.5 HDL and DCNN

A significant dependence has been put on standard video surveillance to achieve security for improving safety and security. Nevertheless, this creates a problem of video data that an administration director must observe. It is not feasible for sizeable metropolitan areas because it produces a frequently massive workload for controlling officials; the error rate is increasing for this problem. This problem could be solved using a regressive auto model, but there are lots of shortcoming for this model. Chakravarthy, SCHMITT and YANG [9] recommended a suspension for this problem using neural networks in union with a Composite Deep Learning algorithm to examine video stream data. Their operation will immediately distinguish and evaluate illegal movement, which will reduce workloads on the supervising directors. The frame data is obtained from a video data set using Hybrid Deep Learning (HDL) algorithm, which displays numerous expression patterns such as crowd movement,

facial features, object interaction. They explained that their crime apprehension method achieves significantly over a massive data set. Their system can be implemented in various video inspection operations, acting as an alert system, decreasing the total load on safety official.

2.2.6 K-Means Clustering and J48 Algorithm

In Lekha and Prakasam 2017 [17], several data mining methods and cybercrimes in banking applications were impersonated. They completed a fundamental data mining technique like K-Means, Influenced Association Classifier and J48 Prediction tree to examine cybercrime data sets and classify attainable difficulties. The classifier can drill the record and express the foresight of cybercrimes with the J48 algorithms using the K-means algorithm. The Influenced Association Classification is an advanced representation for organisation and assistance with weighted assistance and assurance measures. The combined acquaintance of all three algorithms tends to provide an improved, organised, and precise result over the cybercrime forecast in the banking sectors. In this paper, the authors described a novel way to put all those existing knowledge in a collective form, which helped foretell cybercrimes in the banking sector.

2.2.7 Apriori and Decision Tree + Naive Bayesian Classifier

The paper of Almanie, which is published in 2015 [3] used three datasets to predict the possible future crime in Denver and Los Angeles. For finding frequent crime patterns, they used the Decision tree and the Naive Bayes classifier. Subsequently, they involved Decision Tree and Naive Bayesian classifiers to help to predict future crimes in a specific time. The accuracy rate is 51% of in Denver and 54%of in Los Angeles. They intended to promote their models' findings and apprehend the circumstances that might influence the security of neighbourhoods.

2.2.8 Prophet Model and LSTM

Multiple investigations were conducted in three U.S. cities to search and interpret crime information. They state [13] a unique apparent description that is capable of managing massive datasets and empowers users to investigate, correlate, and interpret evolutionary inclinations and patterns of crime occurrences—predicting inclinations with the optimal parameters, ending-pitch and patterns progression and association of different machine learning, deep learning and time sequence modelling algorithms. In future, for continuing this scheme, competent in concocting numerous data types for a wide range of applications to incorporate multivariate visualization, graph tapping techniques and grained spatial investigation to reveal more models and inclinations. They are trying to accompany numerous true-to-life problem investigations to preserve effectiveness and scalability.

2.2.9 DBSCAN Clustering and KNN Classifier

Sivaranjani, Sivakumari and AASHA [22] introduced a process for predicting and forecasting violations in six cities of Tamilnadu. For crime discovery, clustering methods were applied, and classification methods were applied for crime prognostication. Three different clustering algorithm were implemented, and their achievement was judged based on correctness. In performance measurement, the DBSCAN clustering performs very well rather than the other two algorithms. Based on similarity exploration, KNN classification is applied for prognosticating crimes. Therefore the arrangement supports authority enforcing agencies for enhanced and specific criminal investigation. Improved classification algorithms can extend this algorithm and intensify isolation and other protection agencies to preserve the violation data.

2.2.10 Fuzzy Clustering Algorithm

In Yamini, which is published in 2019 [24], a multiple clustering approach is proposed based on fuzzy clustering theory. The FCM algorithm works how an individual data point has been grouped in multiple clusters. The final results are used to analyze the crime-prone states in the US to stop by enhancing the security level in those regions. The results are only helpful for crime analysis, but there is a need to analyze the crime patterns in the future. The prognostication of crimes is impracticable, but it can be restricted if the time in which the offence is happening is recognized. In the future, the pattern analysis of imminent crime can be performed using association rule mining and the proposed system. Moreover, the work can be extended to predict the time in which crime may happen.

2.2.11 TCP

Zhao and Tang wrote a paper in 2017 where they [26] introduce a novel framework called TCP. It catches temporal-spatial similarities, both intra-region and inter-region temporal correlation for crime prognostication. It uses all types of data for utilizing complex urban sources. For evaluating their passageway with extensive operations based on real-world metropolitan data about New York City. The outcomes reveal that their skeleton can precisely prognosticate offence quantities using temporal-spatial correlations. In the future, if they can use more temporal-spatial patterns, it could be used to advance crime prediction.

2.2.12 Improved DBSCAN

In this paper [6], they propose the peripheral continuations of DBSCAN associated with the classification of core intentions, yelling intentions, and neighboring groups. Spatial-temporal clustering algorithms extract valuable information, reflecting the spatial and temporal neighbors of an object. They attempted to improve the current algorithm in three significant areas. They apprehend specific noise points while

clusters of varied frequencies exist, which is one of the shortcomings. Moreover, they are satisfactory if the clusters are separated but not competent when clusters are contiguous. The proposed solution of 3 problems of the DBSCAN algorithm:

- To measure the correlation of spatial data with one dimension, the DBSCAN algorithm utilizes simply one distance parameter, Eps. For two-dimensional data, they introduce two distance metrics, Eps1 and Eps2, defining the relationship by an agreement of two frequency questionnaires. Eps1 is utilized for spatial consequences, and Eps2 is utilized to estimate the correlation of non-special substances.
- For noise objects, they proposed a new concept: density factor.
- The values of margin recipients on both factions may be somewhat unconventional if the non-spatial values have slight inconsistencies and the clusters are contiguous. To solve this obstacle, they matched the mediocre value of a cluster with the new coming value.

2.2.13 Fuzzy C-Means Clustering

One of the main advantages of this paper [24] is that it provides the most beneficial outcome for flapped dataset and approximately more beneficial than the k-means algorithm. One data object may relate to more than one group or cluster. That is why each data object is committed associated to each group or cluster centre due to which. The main disadvantage of this paper is the predictive description of the number of clusters. It provides a better result, however, at the expense of the number of repetitions. Euclidean distance measures can unequally weigh underlying factors. One of the shortcomings of this paper is that the final result of this paper can be used to analyze the crime-prone states in the US. Nevertheless, there is a requisite to analyzing the crime patterns that can occur in future. This paper does not predict or forecast crimes in future. They made only 3 clusters, namely Murder, Assault and Rape. The pattern analysis can be performed with

the proposed system. So prediction algorithms can also be used with this proposed system.

2.2.14 TCP and ADMM Optimization

In this exposition [26], they try to use temporal, spatial correspondence for crime prognostication with metropolitan data. There are two challenges, one is "observation of temporal spatial patterns about crimes," and the other is "mathmatical modeling."Three types of crime prediction systems are used in this exposition. One of the essential sights is that prognostication is more straightforward than the distant future for the near time. An impressive sight is that violations stretch before Christmas but diminish dramatically during Christmas and New Year. These pronouncements recommend that weekends could matter in crime judgment. This exposition shows that temporal-spatial correlations can help crime prognostication. If we can use more and more temporal-spatial patterns could advance crime predictions.

2.2.15 Machine Learning Algorithm and Orange Data Mining Tool

In Goel, Sharma, and Gurve 2019 [14], the Global Terrorism Database has examined that encompassed statistics on domestic and international terrorist attacks. Authors tried to do analysis and prediction repercussions using open source data mining tools. Their model was developed by the orange data mining tool, and different machine learning algorithms such as SVM, Neural Networks, Naive Bayes, and KNN were applied using orange. Their analysis results were compared, and the most suited result was used for prediction based on different attributes of the dataset. Their main objective was to analyze the Global Terrorism Dataset and produce some results which will be beneficial and exciting. They used the Orage data mining tool because it facilitated the analysis of the Global Terrorism Dataset using different data mining methods and compared their results quickly, and the best classifiers

can be found to produce more accurate results by just choosing the attributes. The authors claimed that their work would be helpful for the future because it will help look deeply into data.

2.2.16 Machine Learning Algorithm Comparison

In Agarwal, Sharma, and Chandra 2019 [2], the main focus was on examining the traditional dataset of the Global Terrorism Database and prognosticating the representatives that might supply a surprise to an expansion of terrorism. Different data mining methods and machine learning algorithms are used to examine the dataset and conduct out prognostications like the achievement of a selective strike, prognosticate the organization that conducted out an attack, and the effect of the external factors. A detailed illustration for each algorithm is conducted out to accomplish numerous significant results. The paper also explained the peculiarities that may influence the interventions and their association in imminent event prognostication. Numerous approaches for prognosticating the antagonist groups, prognostication of completion, and the impact of weather situations were introduced. The implementation of Random Forest, Logistic Regression, Naive Bayes, Decision Tree, K-Means Clustering, and the dummy classifier has conferred the development in prognosticating correctness.

2.2.17 Machine Learning Algorithm Comparison

Mo, Meng, and Zhao 2017 [19] focused on the prognostication of the terrorist incident from the Global Terrorism Database (GTD) with some methods along with some feature selection methods which were used for further improvement of the classification efficiency. It was pronounced that the classification arrangements could be applied to map diverse constitutional types in terrorism with both high accuracy and fast speed. It was seen that the more the number of features, the more the accuracy increases. Furthermore, a well-chosen feature selection can drive the decrement of classification error.

2.2.18 Lazy Tree and Multilayer Perceptron, Multiclass and Naive Bayes classifiers

In Kumar, Mazzara, Messina, and Lee 2020 [16], for inspecting the trends for terrorist aggression, they use three types of classifiers such as Lazy Tree, Multilayer Perceptron, Multiclass, and Naive Bayes classifiers. Numerous more significant patterns can be found using these approaches because it is a user need-based approach. In the future, there can be numerous diverse types of researches based on this approach which has an accuracy of classifiers almost to 99%. Furthermore, it was produced using different classifiers by combining the diverse classifiers. The drawback of this approach was restricting the categorization of characteristics, which diminished the calculation complexity. Nevertheless, groups having a marginally below appearance further produced some prejudicing, which was a benefit. They also improved the sub-classification layers and associated them with finding more valuable inclinations, which confirmed significance. They claimed that it is conceivable to prognosticate the organizations associated with the recorded outbreaks using their work where the parameters could be different sources.

2.3 Problems of Existing Systems

Many researchers have been working on crime prediction for many years. Early-mid 90's people used the statistical model to predict crime; still, many of them use the statistical model to prognosticate the crime. Then gradually the conventional machine learning and deep learning began to become famous. Initially, deep learning could not be so popular because of the lack of computational power. That is why machine learning plays a crucial role in crime prediction. So many researchers used machine learning techniques to solve this problem. Some of them used supervised learning techniques such as random forest, decision tree, SVM, KNN, etc.

Nevertheless, one of the main issues of using traditional machine learning is. It cannot capture the dynamics of the data. From so many studies, we found that there

are so many responsible factors for crime. We cannot use all the factors/variables to predict the crime using traditional machine learning techniques. Then some researchers tried to use clustering techniques to solve this problem. The K-means clustering, DBSCAN clustering, works very well in the prediction of crime. However, here the same problem, these techniques also could not capture the dynamics of crime. All these techniques used some factors responsible for crime occurrence, but they cannot use all the factors responsible for the crime. Then comes the deep learning techniques. So many researchers also work on deep learning. Some of them proposed a feature-level data fusion method with environmental context based on a deep neural network. Some of them try to detect crime anomalies from video data of CCTV using deep learning. Some of them try to collect real-time data from social media and predict the occurrence of crime. Some problems in the existing solution are summarized below

- Could not capture the dynamics of crime
- Not utilizing all the possible factors that are responsible for the crime

Everyone tries to predict crime without considering all the factors that are affecting crime. To predict crime correctly, all of you should have considered all the possible factors that affect that prediction.

Chapter 3

Proposed Methodologies

In this section, we will try to provide a solution to our previously introduced problem. We will also try to find out if the provided solution will work or not and how good the solution is.

3.1 Prototype Tools

Nowadays, the thing what plays an important role is data. So gathering insight from it is important. To do this data mining plays an important role. It helps us to get those insight. To use those data mining techniques python comes handy. It gives us flexibility and the ease to perform data analysis and program those machine learning algorithms. It is the python libraries which has made the job easy. Python gives us the ease to try something new, something that has never been tried before.

Learning python is easy because it is simple and readable. That why python is the go-to tool for adventure. We can dodo the long task by writing only some lines of codes in Python. Those tasks usually need a long process to complete with some other languages. Moreover it is free and available for anyone to use. A community based model is used for the development of python. Python can be run in any platform, Windows or Linux. The libraries of python are open-sourced. Those open source libraries include things like Statistics, Machine Learning, Data Visualization,

Data Manipulation, Mathematics, Natural Language Processing and so. If someone faces trouble while using python, it's only a step of a google search. Because the number of user and forums are huge. So it is almost impossible for something to go wrong while using python. So getting help is never a challenge. Moreover python has a huge used base and it is also heavily used in academic because of the availability of libraries which can perform the task of analyze. Moreover, when a user gets blocked with some problems, they usually post about those solutions in the public forums or in the places like stack overflow. So it becomes also easy to get access to those solutions. The popularity of python is still rising as well as the available support material. So python is the go-to language for the data scientists and data analysts because of its huge advantages.

if anyone needs something to automate or need something over and over, python is also good at those tasks. That's why libraries like Numpy, Pandas, Matplotlib is used in our work. It has helped us to generate functions only having a small basics in it. Different types of graph generation, formatting outputs, visualization of data became so easy using python for us.

3.2 Overall Methods

If we want to describe the whole process, we can divide it into 4 steps for better understanding. Figure 3.1 is a diagram of our overall methodologies. Following are those 4 steps which with help to describe our methodology. We will discuss about them in next section of our report. The output of those will also be used in the in the later chapters.

1. Data preprocessing
2. Correlation analysis
3. Feature Selection
4. Training the model

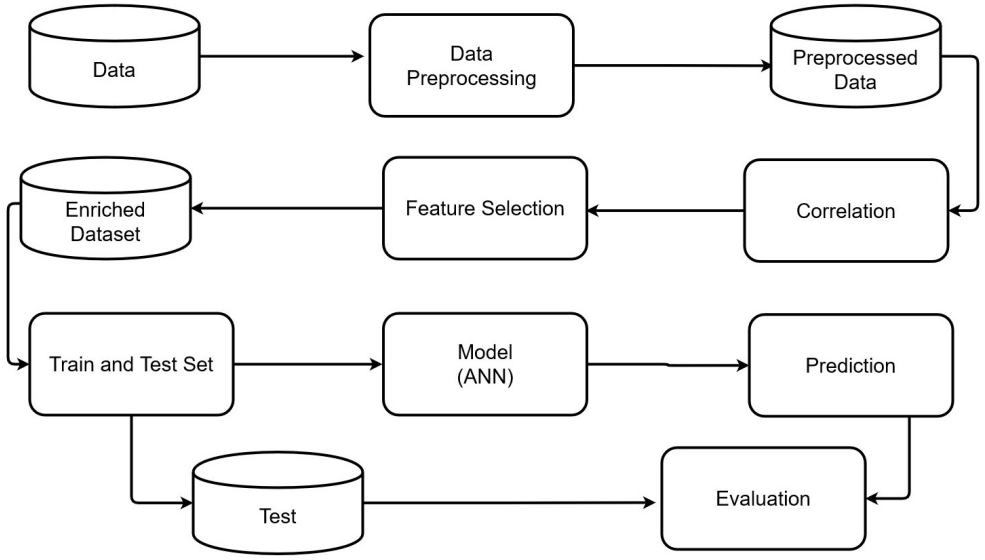


FIGURE 3.1: Overall Methods

Now we will try to describe each steps one by one to have a better understanding of the whole process.

3.2.1 Data Preprocessing

In the GTD database, there were so many columns of data which had so relation with prediction or out work. So those columns were simply ignored. There were some rows with some missing values. Around 1% of rows had this type of inconsistency. As GTD is a huge database, they were also ignored. So there was no need to estimate those missing values. Our target of prepossessing was to turn the database into numerical format. Some columns had non numerical values, which was turned into some ranged numerical values according to their possible impact or meaning.

At first we will process the dataset. To process the GTD, we will follow the Algorithm 1. In the Algorithm 1, we have pass our dataset(GTD) along with a binary string containing a series of 0s and 1s. The length of that binary string will be the same as the number of column in the data. An 1 in the i_{th} place in sting will mean that we want to keep the i^{th} column of data in our preprossed dataset. And

Algorithm 1 ModData Process

Input: δ ←dataset
$s \leftarrow$ a binary string ▷ indicating whether an attribute will be considered
Output: δ' ←modified dataset

```
1: l ← length(s)
2: δ' ← δ
3: for i = 1, 2, . . . ,l do
4:     if s[l+1-i] = '1' then
5:         Delete (l + 1 − i)^th column from δ'
6:     end if
7: end for
8: N ← number of row in δ'
9: M ← number of column in (δ'
10: for i = N, N − 1, . . . , 1 do
11:     for j = 1, 2, . . . , M do
12:         value ← δ'[i][j]
13:         if value is NULL then
14:             Delete i^th row from δ'
15:             break
16:         end if
17:         if value = "YES" then
18:             δ'[i][j] = 1
19:         end if
20:         if value = "NO" then
21:             δ'[i][j] = 0
22:         end if
23:     end for
24: end for
```

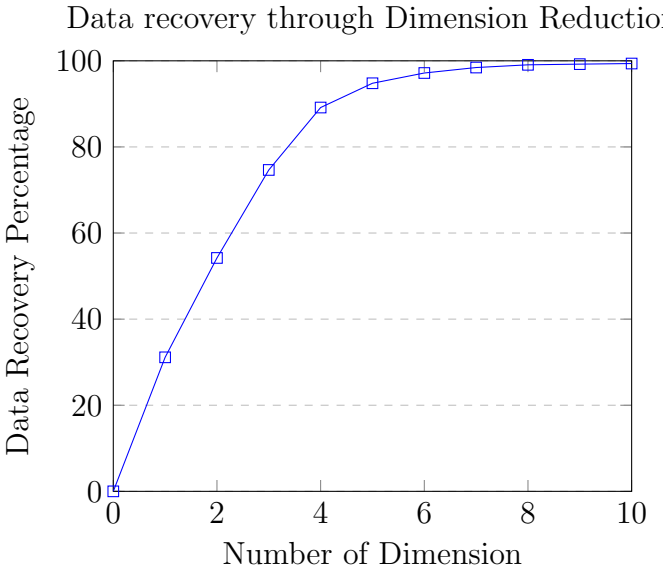
0 in the corresponding place means we want to delete that corresponding column. As GTD has a lot of columns which has no impact on our outcome, so all those columns were just deleted.In the Algorithm 1, from line 1 to 6 has done the work of deletion. As GTD is a huge dataset we just deteted any row if this has a missing value, And some columns consist values like "yes" or "no", we just converted those values in 0s and 1s. In Algorithm 1, in line 9,10 we loop through the dataset and from line 11 to 21, we searched for missing value or yes-no values and act upon it.

3.2.2 Correlation Analysis

Correlation analysis is a process where we get to know if any feature is correlated with other features. Suppose, we have a total of n features, from where we want to keep m features. If we run correlation analysis over all the features, then we can rank those features over their scoring and can easily eliminate the $m - n$ features. Their are various correlation analysis techniques. Here we have used Pearson Correlation Analysis. This allows us to calculation both positive and negative correlation effects.

3.2.3 Feature Selection

In GTD, there were about a hundred of features. It is not feasible to execute an algorithm on all the available features because it will degrade the performance. Moreover when the number of feature becomes large, the visualisation also becomes tough in a graph. So it is an important thing to reduce the number of feature in way that it also stays easy for visualization. But in the meantime, reducing the number of features will reduce the accuracy. Here is a table which represents the accuracy of prediction after running the algorithm with preprocessed data using PCA.



There are so many algorithms to reduce dimensionality. But PCA is used mainly for two reasons.

- **Removes Correlated Features:**

  If we want to find correlation manually, it becomes an frustrating an impossible task, which is also time consuming. PCA makes it easy for us. When we run PCA on a dataset the components of the dataset becomes independent. There stays no dependency among the features of that dataset.

- **Reduces Overfitting:**

  When the number of attributes become so large, the overfitting happens. As PCA reduces the dimension of a dataset, the issur of overfitting also vanishes as it reduces the number of features.

Following is the algorithm what is used while the dimension reduction process of PCA.

Algorithm 2 RedDim Process

Input: $\delta \leftarrow$ dataset
$n \leftarrow$number of dimensions of δ
Output: $\delta^{'} \leftarrow$dataset with reduced dimensions

1: $\delta^{M} \leftarrow$ toMatrix(δ)
2: meanValue $\leftarrow$ getMean(δ)
3: adjustedData $\leftarrow$ adjustToMean(δ^{M}, meanValue)
4: $\delta^{T} \leftarrow$ getTranspose(adjustedData)
5: deltaMatrix $\leftarrow$ mul(adjustedData, δ^{T}, n)
6: reducedMatrix $\leftarrow$ getEigenValues(deltaMatrix, n)
7: $\delta^{'} \leftarrow$ toArray(reducedMatrix)

After doing preproccesing, we went for reducing the dimension of the data. In fact, this is also a part of preprocessing. In Algorithm 3 line 2, we called the function "reduce dimension" to reduce the dimension of out dataset. It takes the processed dataset from line 1 and the number of dimension as parameters. Then it reduces the dimension of the data following the Algorithm 2. In Algorithm 2, we tried to reduce higher multi-dimension data to lower multi-dimension data. To do this we project those higher dimension data into a lower dimension in such a way, thus the error will be minimum. For this, we used the ideas of linear algebra. In line 1, we just rewrite our dataset in matrix form. And then calculated our mean

value. This helps to reduce the error while dimension conversion. And then we adjust those data, mainly fit them in a range from 0 to X as per line 3. And then we have to transpose the matrix as per line 4. And then we multiply those 2 matrix and compute the reduced matrix by calculating their eigen values [line 5,6]. After that we just transform those values from matrix form to array form in line 7.

3.2.4 Training the Model

Artificial Neural Network can be an effective algorithmic approach to predict something. Though they are resource-intensive and their results are often hard to interpret, in case of big dataset those concerns can be ignored and those will not affect much in case of using GTD. Before training our model with ANN, we need to fix some values for training our model properly.

3.2.4.1 Number of hidden nodes and layers

There are some things that need to be fixed before working with hidden layers. The first one is the number of hidden layer that will be included the artificial neural network structure. And the second one is the number of nodes allowed in those hidden layers. We will now discuss about them in the upcoming sections. To determine the number of hidden layer to be used, we should recap about the necessity of the hidden layer in the network. If the relation of the input and output attributes are linear, there is no use of a hidden layer in the network. Hidden layer is required, when we need to map something from a finite space to another finite space. In this scenario, only one hidden layer is enough. When the things become complex, suppose we want to move an arbitrary boundary to an arbitrary accuracy, then more than one hidden layer is necessary. This can also be done using one layer but to have smoothness and high precision, it takes more than one hidden layer to accurately perform these complex mapping. In our case, the function is not that complex, the behaviour behind the criminal acts are somehow not arbitrary. That's why only one hidden layer will be good for our problem. It is hard to prove this

fact theoretically. Usually, all the practical problems can be solved using only one hidden layer in the Artificial Neural Network.

We have talked about deciding the number of hidden layers this far. But the most important part is to decide the number of neuron in a hidden layer. People may think a neuron more or less will not affect the result this much. But deciding the number of neuron currently can result in a tremendous success. There can be two kind of errors while deciding the number of neurons. The first one is underfitting. This one happens when the number of neurons are too few than the perfect number. When underfitting happens,it can not detect any major signal coming from the input data. The second one is overfitting. Overfitting happens when the selected number of neuron for a hidden layer becomes way too large than the perfect number of neuron. If the number of neurons gets so much that means the input data can not provide enough resources to process to the hidden layer. What results in incomplete training of some neurons in the hidden layer. Moreover the computational times gets much more greater. Sometimes the computational time can get close to the impossible to compute. So we have to make compromise between using too many or too few neurons in the hidden layer. Setting in between those two can produce better result and can also help to reduce computational time.

All those rules are just some points to keep in mind while deciding the those numbers. Actually the process to decide those numbers come from trial and error. But just using random numbers to decide if an architecture is useful is time consuming and foolishness. We can decide a range of useful numbers by remembering all those theoretical facts. As previously mentioned in the dimension reduction part, the number of input node was only 8, and if we want to avoid underfitting and overfitting the number of nodes in between 6 and 100 will be okay. As the size of the test set is huge in our case, we will keep it as minimal, so that the number of hidden layer nodes will be 6.

3.2.4.2 Choosing the activation function

It is the activation function which performs the task of the transformation of the summation of the weighted input from the node to the activation of a node in the Artificial Neural Network. The thing a Neural Network does is mapping. It maps the values of the input nodes in the output node withe help of the neurons.To do this, a neuron multiplies the value of the input node by its own weight, which activates the node based on some criteria. The criteria is dependable on the activation function. So the output is heavily dependant over the activation function.

There are so many type of activation function. the simplest of them is a linear activation function. In this no kind of transformation is applied. That also makes a network easily trainable. The drawback of this is, sometimes the complex things can not be learned in this process. So it is only usable in the output layer, where the neurons are directly connected to the output node. Functions like sigmoid and hyperbolic, can learn complex things. That's why those activation function perform well near the input nodes. These functions produce output values in a particular range. They are so much sensitive in the middle of their range, which is logical. Thus the outlier values have less effect on a dataset. The full form of ReLU is Rectified Linear Unit. The best thing about ReLU is that it is a linear function. It just output the input value if its possible, otherwise it outputs a zero. IT seems naive, but it has become everyone's favourite as it can produce extremely good result in a much less computation time. Following are some more reasons to use ReLU as an activation function in our network.

- The calculation of derivation of the ReLU is so easy. It saves computation time. The derivative values are used while updating the value of the weights of a node in the calculation of backpropagation errors.
- The rectifier function is trivial to implement, requiring a max() function. That doesn't need much computational power, which reduces the complexity.

- The ability to output a zero value comes handy. Whenever an input value becomes negative, it outputs zero. It can help the activation of a neuron because of the repetition of the zeros. This thing is called sparse representation. And this property is a good property as it can activate any neuron so faster as well as the learning process and the learning rate.

- The ReLU looks like a linear activation function. It also acts like it. In general, The optimisation process of an Artificial Neural Network becomes easy when it behaves like a linear function or close to a linear function. A network which is trained using RelU as an activating function can don't get in the trouble of the vanishing gradients, because the activation of the nodes remain proportional to it.

As preprocessing of the GTD doesn't process much outlier values and the values are distributed fairly through a range. That makes ReLU a perfect activation function for this case.

3.2.4.3 CriPred Algorithm

If we summarize the whole process this would look like Algorithm 3, where we have to input the dataset, number of dimension, test-train ratio, number of hidden layer node, threshold value and number of iteration. We will figure out what those terms mean one by one later in the description.

When all those preprocessing parts are finished, we will have a reduced dataset. Reduced dataset mean this dataset now have less value than before and the dimension of the data is also less but the impact of those values are now effective. By following Algorithm 3 line 3, we will now randomly split those reduced data into test set and train set with the help of the test-train ratio provided as input. After that, we have to build the network and assign the values of the nodes. Here the value of hidden layer node is a parameter to build the network [line 6]. Then we have to assign the weight value of each hidden layer nodes in the network as per line 5. After this the training phase starts. We will iterate a loop a fixed time,

Algorithm 3 CriPred Algorithm

Input: $\delta \leftarrow$ dataset
$\omega \leftarrow$ number of dimension
$r \leftarrow$ dataset split ratio
$n \leftarrow$ number of epoch
$k \leftarrow$ number of hidden layer node
Output: $\eta \leftarrow$learned model

1: $\delta' \leftarrow$ ModData Process(δ)
2: $\delta^r \leftarrow$ RedDim Process(δ', ω)
3: testSet, trainSet $\leftarrow$ randomSplit(δ^r , r)
4: $r' \leftarrow$ number of rows in δ'
5: $c' \leftarrow$ number of columns in δ'
6: $\eta \leftarrow$Initialize network(ω, c')
7: **while** number of iteration **do**
8: **for** $i = 1, 2, \ldots, r'$ **do**
9: Feed forward δ'[i] in η
10: **end for**
11: **for** $j = 1, 2, \ldots, k$ **do** ▷ Forward propagation
12: Calculate the input's weighted sum from the node j
13: Calculate the activation for the node j
14: **end for**
15: Calculate the error signal for the output node
16: **for** $j = 1, 2, \ldots, k$ **do** ▷ Backpropagation
17: Calculate the signal error of node j
18: Update the weight of node j in the η
19: **end for**
20: Calculate global error ▷ Error function
21: **end while**

which is given as input [line 7]. In each iteration we will do some tasks: forward propagation, backward propagation, calculating errors and update weights. At first we will pass all the values of training set into the network [line 8-10]. As We have only one hidden layer, we don't need to loop through every layer. So we will loop through the nodes of the hidden layer, and calculate the input's weighted sum which are connected to that node, described in Section 2.1.2. Then we have to add the value of threshold to the weighted sum and then we have to calculate the value of activation for that node. The value of activation defines how much impact that node will make in the output [line 11-14]. Then we have to calculate the value of error signal for every node in the output layer [line 15], which helps to calculate the global error value later [line 21]. This error value means the difference of actual

and the predicted value. After that the backward propagation phase starts. Again we have to loop through the nodes of the hidden layer to calculate the signal error of that node, which will help to update to weight of that node. And then we have to update the changed weight value of that node to the whole network [line 16-19]. In each iteration the error value will decrease and the model will start finding pattern in the data. With this pattern the model will predict a output value for other inputs.

3.3 Summary

These are the all four steps. Where we firstly preprocessed the dataset. Then on the preprocessed dataset we run correlation analysis. After that we fixed some parameter to run our model and then we finally run our model. After performing those steps we will have a trained model, with this we can get results and have result by analyzing them.

Chapter 4

Implementation

Our work's composition is based on the Artificial Neural Network for crime prognosis in any area using the modified Artificial Neural Network algorithm. We possess two parts in our project, one is algorithm design and implementation and the other is user interface with a server design and implementation. Figure 4.1 is our overall

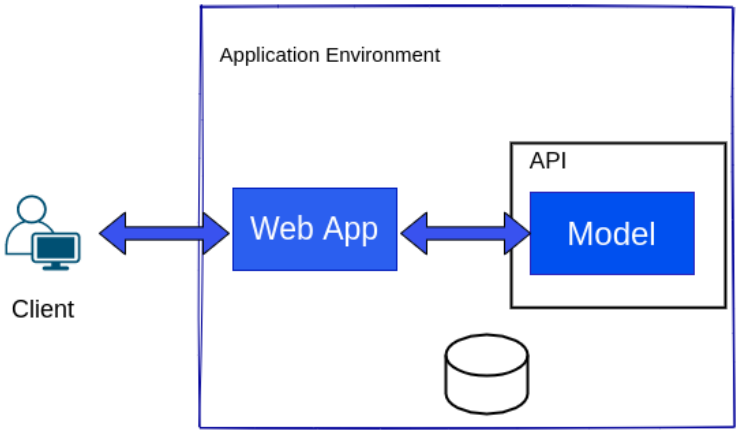


FIGURE 4.1: Architecture of our system

architecture of our system. The emerging subsections will move into detail about the design and implementation of these two parts in our work.

4.1 Web Application Architecture

Essentially, a web application is a computer program or instructions that use web browsers and web technology to accomplish jobs over the Internet.We are making an web application for showing all the results and outcomes of our project to the users. For this we need a server and some user interface. Figure 4.2 is the architecture of

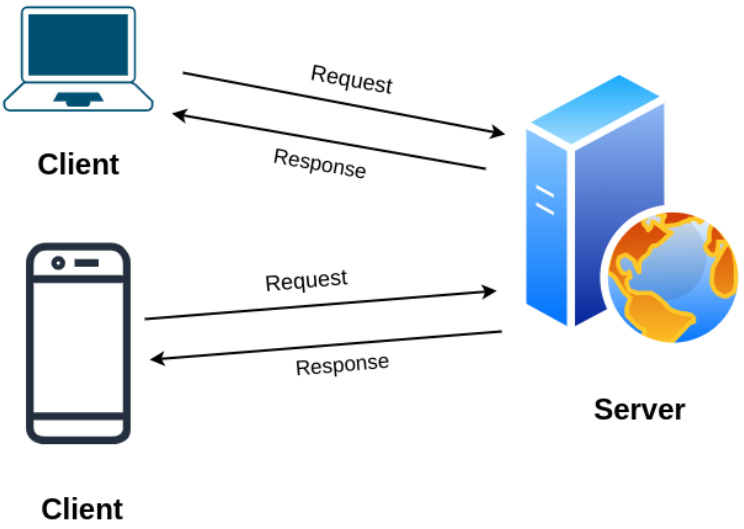


FIGURE 4.2: Architecture of web App

our web app. For a web application to function, it needs a Web server for responding to the requests and a client for making the requests. Clients can be anyone with a browser with any device. Web servers handle the requests from a client while the application server executes the demanded task.

4.1.1 Web Server

A web server is a machine for running websites or web applications. It is an instruction of a computer application that assigns web pages as they are requested. The fundamental purpose of the webserver is to stock, process, and release web pages to the recipients. This interchange is made utilizing Hypertext Transfer Protocol (HTTP) which is a widely used web protocol. These web pages are mostly static content that includes HTML texts, pictures, style sheets, tests, etc. For making a

web server, we use Flask, a Python-based micro web framework. We make an API endpoint for communicating with the server and client.

4.1.2 High Level User Interface

The user interface (UI) is during individual users communicate with a machine, website, webpage, or application. We are making a website to show the outcomes of our project. So we need a user interface for the user of our application. Figure 4.3 is the initial high-level design of our website. We use basic HTML and CSS

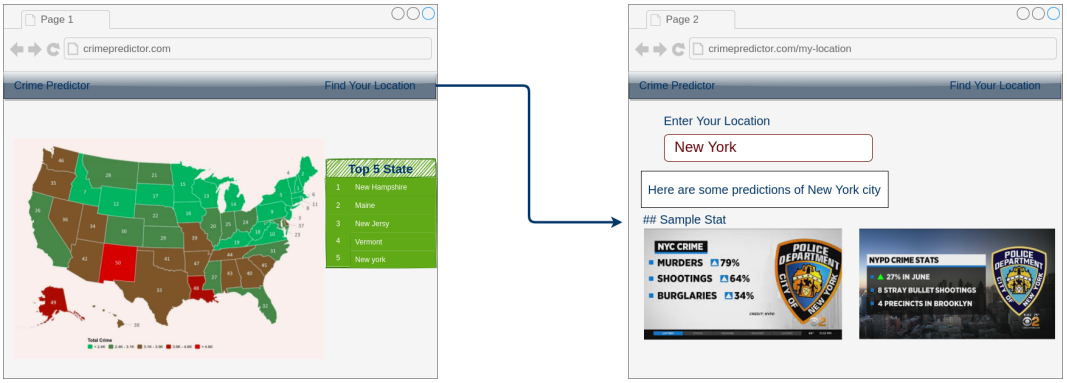


FIGURE 4.3: High Level UI desing

for designing our website. For making a choropleth map, we use D3.js; a special javascript library.D3.js is specially used to produce effective, interactive, dynamic data visualizations in websites. It advances the use of SVGs, HTML5, and CSS standards.

Our website has two routes. The main route is a landing page where we will show the overall predictions of our model using a choropleth map. In choropleth map we will show all the states crime prediction. If the states color is red means it's very crime prone state and it the states color is green then it's very less crime prone area. We will also show a table where the top five crime prone states will be shown. Another route is '/location' route where user give a specific location name and then for his location he can find the useful crime statistics of his given location.

4.2 Implementation Of Our Model

For model implementation, we use some open source python libraries. Keras [1] is a compelling and easy-to-use available, open-source Python library for generating and appraising deep learning models. It envelops the practical mathematical calculation libraries called Theano and TensorFlow and enables us to establish and train neural network models is simply a few lines of code.Figure 4.4 shows the steps for implementing a neural network model using Keras [1] sequential API library. All

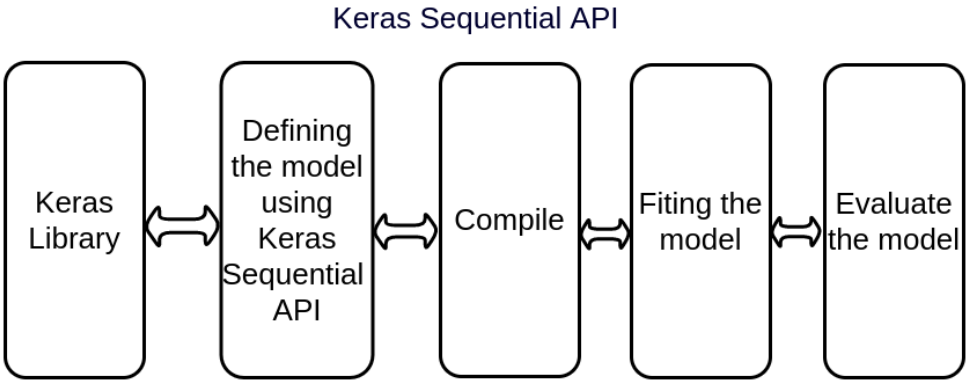


FIGURE 4.4: Steps of model implementation

the steps that we followed for implementing our model is described below.

4.2.0.1 Required Library

Keras library has two classes, and we will use these two classes to define our model. Sequential groups a linear stack of layers into a "tf.keras.Model," where a sequential model is suitable for a traditional stack of panels where each panel has precisely one input tensor and one output tensor. The training and inference features on this model are provided by the Sequential class. If we want to add more layers, we can combine panels occurrence on top of the panel pipe.

4.2.0.2 The Definition Of Our Model

The definition of Models in Keras is interpreted as a sequence of panels. For creating a Sequential model and attach panels one at a time until we are satisfied with our

system structure. Using the "Dense" class, fully connected panels are determined. To particularize the number of neurons or links in the layer, we can particularize the number of neurons or links in the layer as the first case and specify the activation function using the activation case. We can add layers using "add" method.

4.2.0.3 The Compilation Of Our Model

After defining the model, we can now compile our model. In the compilation process, the model utilizes the effective mathematical libraries following the top means in the backend, such as Theano or TensorFlow.The backend determines the most dependable way to interpret the system for preparation and prognostications to run on our devices, such as CPU or GPU or even shared. In the compilation process, when training the network, we require to define some supplementary properties. We designate the loss function to estimate a set of measurements; the optimizer is used to explore various measurements for the network and any arbitrary metrics we would like to handle and reach while training.

4.2.0.4 Fitting Our Model

After the definition and compilation of our model, it is time for computations and execution on some data. There is a function name fit(), and we can train or fit our model on our loaded data by calling this function on the model. The process of training transpires over every epoch, and each epoch is divided into batches.

- **Epoch**: Passing through all the rows of a dataset using one pass.
- **Batch**: Number of samples considering by a model within one epoch before the weights are updated.

One epoch includes one or more batches, based on the chosen batch size, and the model fits many epochs means multiple batches can be within one epoch. We can use any number of epochs and use any batch size.

4.2.1 Evaluating Our Model

We trained our model on 80% of our data in the training process, and we judge the model's achievement based on the remaining 20% data. To evaluate our model on the training dataset, there is a function called evaluate(). We use this evaluate() function on our model with the arguments. The evaluate() function takes our input, and it returns a list with two values. The one value is the loss of our model on our dataset, and the other value is the accuracy of our model on the dataset.

4.3 Saving Model for API

For connecting the model with web app is done by a simple python libray called pickle. Pickle is the conventional method of serializing objects in Python programming.We can apply the pickle formula to serialize our deep learning models and we can save the that to a file. Succeeding we can load this file to deserialize our model

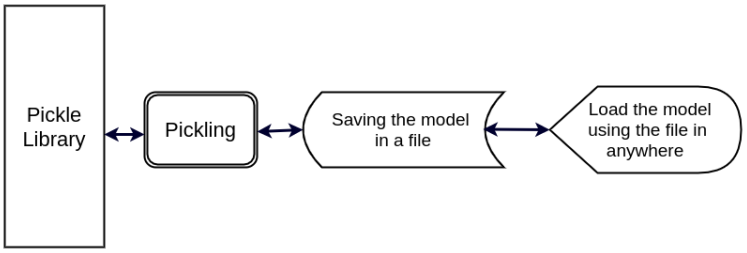


FIGURE 4.5: Connection between web app and model

and we can use it to make new prognostications.Figure 4.5 describes the overall process of saving a model and load the model in anywhere.

4.4 Summary

In this chapter, we describe the overall architecture of our system. We need a web server for our application that will connect our deep learning model using an API. We also had some frontend functionalities, and we describe the implementation

process and tools that we use for building the overall system. We use Flask, a python-based web framework for our server, and we use D3.js for our visualization purposes and basic HTML, CSS, and javascript.

Chapter 5

Experimental Results

In this section, we will discuss about the result we found using our algorithms. We will also try to highlight any significant improvement in result.

5.1 Result Analysis

After getting the trained model We have to generate some results to find out the validity of out proposed method. For this we will try to see how the correlation analysis has performed to train the model. Then we will evaluate the model by generating some outputs and comparing them with some other methods, which we will call performance analysis.

5.1.1 Correlation Analysis

We conducted analysis on Global Terrorism Dataset using python script and portrayed the results using graphs. Figure 5.1 shows the correlation between the attributes of the dataset using Pearson Correlation Coefficient, ρ. In the Pearson correlation coefficient method, the covariance of any pair of the variables is divided by the product of their standard deviations. This method is familiar in the name of Pearson product moment correlation coefficient or the bivariate correlation. These

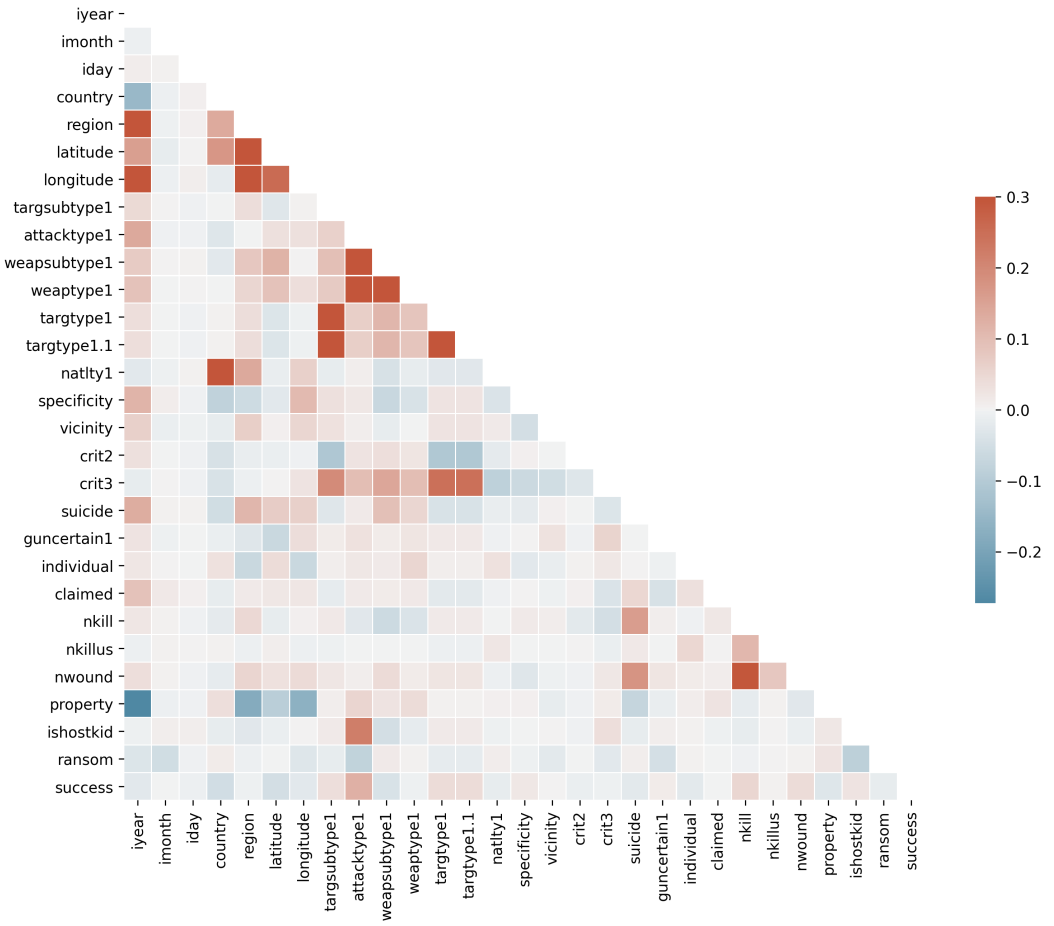


FIGURE 5.1: Initial correlation diagram

are the metric of linear correlation between any pair of data. If we look at the process of how the coefficient is measured, we can agree to the fact that the output will also be in a range from -1 to +1. As the covarience is used here, the relation will be linear. So the other types of relation except linear correlation will be ignored. The this method, the correlation can be dived into 2 types. Positive and negative correaltion. When the value of coefficient is greater than zero, it will be classified as a positive correlation. It tells us that the change of both of the parameters is in the same direction. If the value of the coefficient become highest (+1), which is the highest score, there exists a perfect positive relationship. The perfect relationship means that in any time the change of the direction remains same in between those two attributes. Here the positive correlation is displayed with the shades of red. On the other hand, a negative correlation means the value of correlation coefficient is less than zero. Like the positive correlation, it tells us that the change between

those two attribute are opposite in direction. If the value of the coefficient become lowest (+1), which is the lowest score, there exists a perfect negative relationship. The perfect negative relationship means that in any time the change of the direction remains opposite in between those two attributes. Here the negative correlation is displayed with the shades of blue. When the value of ρ gets close to 0, it means that there is no correlation between those attributes. So from the above description, we can say that, any lighter color in the graph will represent a week correlation and darker color will represent strong correlation.

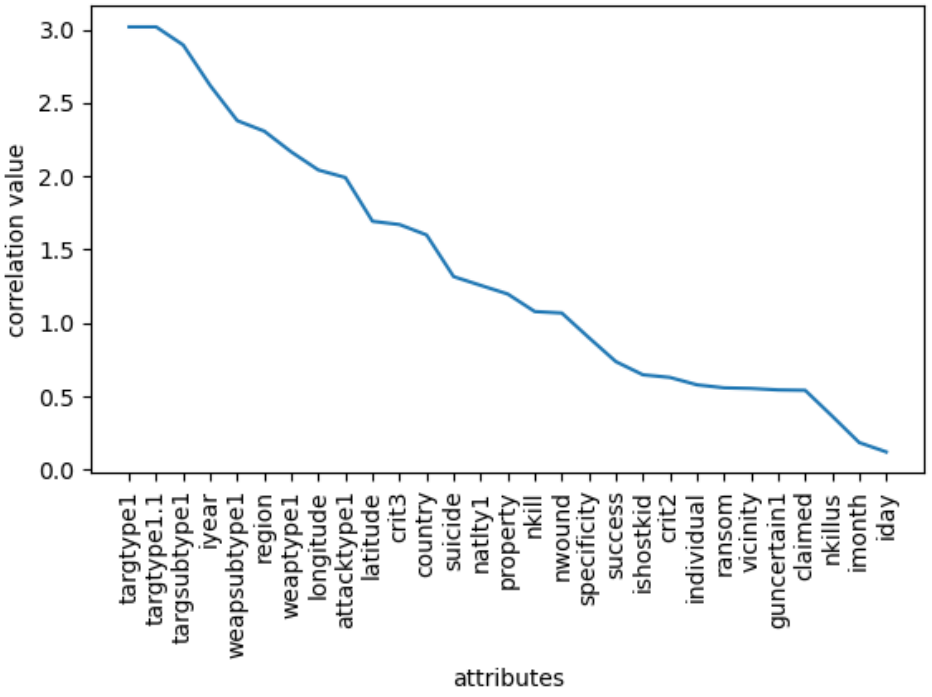


FIGURE 5.2: Correlation score of the attributes

After removing the attrributes containg a huge number of missing values, we left over with 28 attributes. If we measure the pearson correlation coefficient of those attributes with each other then we have a pearson correlation coefficient matrix. And after mapping the values of the matrix in the color shade, we will have a graph like figure 5.1. Here we can see the correlation of attributes guncertain1, individual, claimed, imonth, iday with the rest of the attributes is not strong enough. We can say that by just with our eyes because they hold lighter color shades with every other attributes. So if we want to ignore some attributes to do the calculations of

the further process, these should be the attributes what we should remove from our dataset. After that we made a ranking of those attributes from their importance over the dataset using their correlation coefficients. We can get the average correlation coefficient of an attribute ρ_k using this formula.

$$\rho_k = \sum_{i=1}^{i=\#\text{attributes}} \sqrt{\text{correlation coefficient(attribute k, attribute i)}}$$

Now, using these value from the formula we can rant all attribute from their corresponding values. Figure 5.2 shows that ranking.

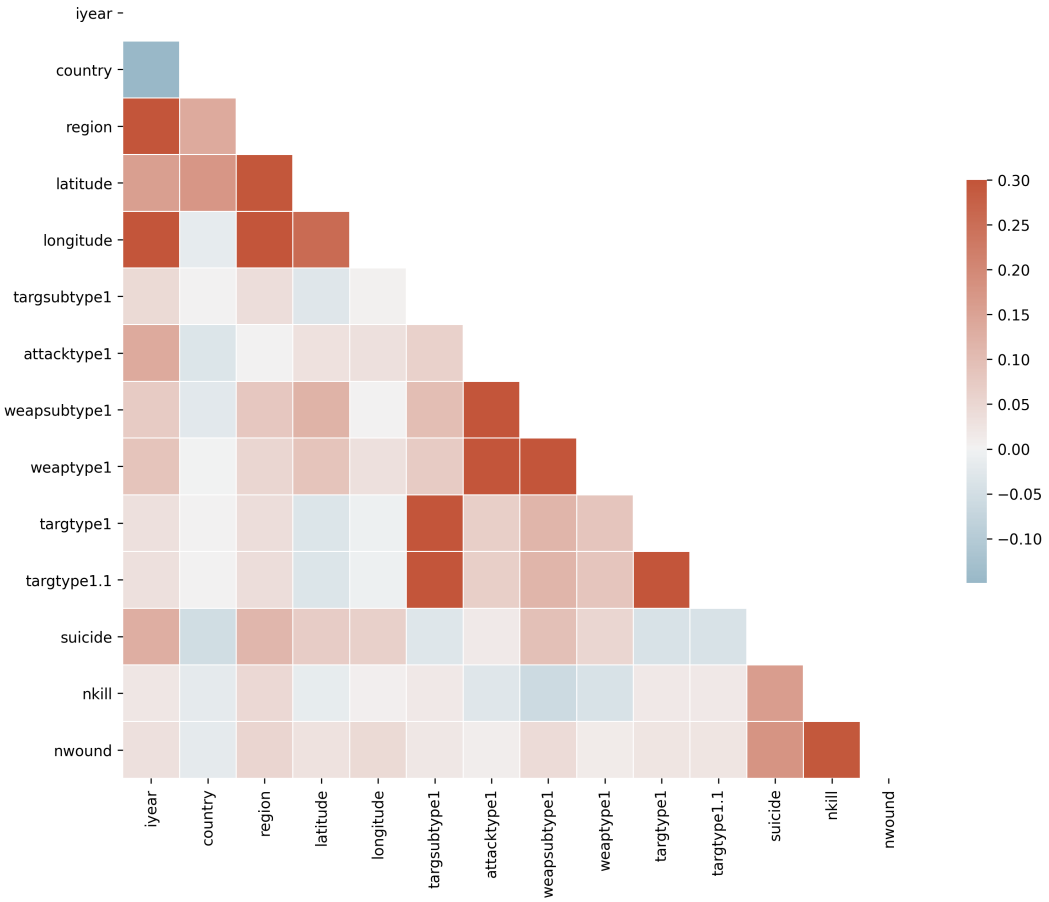


FIGURE 5.3: Final correlation diagram

Now using a correlation coefficient threshold value, attributes with minor effect over the dataset can be dropped. By this the remaining attributes will indicate strong correlation among them. Figure 5.3 shows strong correlation among the final attributes. If we compare this result with the previous image, it can be seen

that those final attributes are also holding relatively large rooted correlation values. Moreover attributes like guncertain1, individual, claimed, imonth, iday are absent here, as we have seen previously those attributes had relatively small correlation efficient values. Now we can say, if we run our model over these attributes, we can expect a good outcome training our dataset.

5.1.2 Performance Alalysis

After training our model with this preprocessed dataset, we can get several results based on different criteria. First you should select the value of Y, to train the dataset with the model. Selecting the value of Y is an important thing to do. A good and meaningful value of Y will produce a meaningful result. So without setting a proper Y producing any kind of result will go in vain. We can select the value of Y based on different criteria to judge our model's effectiveness. In some criteria it may perform well and in some other criteria it may struggle to produce a good result. We can guess 3 different types of Y values to produce results on different criteria.

1. The value of Y is a binary value, where we can get the answer in yes or no.
    - Is this a crime?
    - Is this a suicide?
    - Anyone is killed?
2. The value of Y is short range value.
    - Number of states in a country
    - Number of person hurt in an incident
    - Number of person involved in a crime
3. The value of Y is short range value.
    - Number of crime happened in a year

- Number of people affected from a bomb blast
- Anyone is people got killed over a year

No.	Prediction criteria	F1score (our model)	F1score (SVM)	Precision (our model)	Precision (SVM)
1	If an attack is successful or not	0.9179	0.9187	0.9241	0.9203
2	If a weapon is used in an attack	0.9345	0.8971	0.9375	0.9019
3	How many crime may happen in a year	0.9667	0.8117	0.9669	0.8047
4	In a region which country/state is particularly more dangerous	0.7811	0.6157	0.8021	0.6287
5	Which type of weapon is used	0.6070	0.6145	0.6241	0.6053

TABLE 5.1: Performance Analysis

Moreover we will need another model to compare our result here. We have chosen SVM for comparison because SVM has performed better than the other supervised learning techniques. For evaluating the results we need some scoring techniques. We have chosen F1 score and precision to display our result. We have chosen F1 score to show us an overall result. We have also chosen Precision to give an idea of the False Positive and False Negative ratio over a criteria and how they change. Table 5.1 shows our model's result with respect to SVM based on 5 different criteria.

In case of binary valued criteria, both SVM and our model has performed well. In case of criteria number 3, we have allowed an error of 10 percent as a correct prediction. We can see our model has performed excellently well in that case compared to SVM. Like criteria number 3, criteria 4 and 5 are also range valued criteria. In criteria 4 our model has performed relatively better than SVM but in criteria 5 both of the models have suffered to perform well. To find the reason we can think of an example. Consider, we are trying to predict which area of a city is more dangerous. There may be 2 or more arenas which are equally dangerous. But

the model has to predict a single value. That value might not be the top answer, but that might be the second or third top answer which is so close to the original prediction. But they will be considered as wrong. If we somehow allow both of the top 2 predictions to be correct we will find a similar result like criteria 3, where 10 percent error was allowed. That way we can get a high prediction accuracy. In this criteria our model performs better than the other supervised learning techniques. But in case of predicting binary values, our model doesn't give much advantage over the other models.

5.2 Summary

The result of correlation analysis was significant. It helped to remove all the unnecessary and very less important which further helped us in the time of applying PCA. The generation of the model using criPred algorithm also produces a good result in some cases where the range of the output variation is relatively large. It also performs as good as other approaches when it comes to the binary outputs. It also preserves the fact of criminal justice, we can make sure that by seeing over the values of F1 score and precision. Because every time in the cases of binary value output the value of precision is larger than the F1 score which indicates that the ratio of false negative and false positive is less than 1.

Chapter 6

Conclusions

This chapter includes an overall discussion regarding our work and conclusions based on the results shown previously. We will mention our observations and recommendations in Section 6.1. We have found few more challenges through our study during this work. Some of them are solvable by extending our solution, and some of them need more concentration. We discuss future work and unsolved research questions in Section 6.2.

6.1 Research Summary

In this exposition, we propose a method that captures all the possible factors affecting and encouraging crime occurrence, and by using this, we predict the crime occurrence. Our method combines preceding criminal action credentials in particular areas and parades them based on our model to prognosticate the phenomenon of crimes. We utilize the global terrorism dataset for conducting our research work. Consequently, our method, which utilized an 80:20 ratio of training data creation empirically, dispenses an efficiency of almost 91%. Each of the methods has its shortcomings and benefits. The higher the amount of data, the better our model will work. The expeditious and proper association of criminal activity is predominant to acquiring any nation. The most tiresome part of crime prognostication

is that it is a powerful innovation in its nascent degrees. It has been used in several countries where most of which are developed countries. However, the results have not been dependable. Algorithms are predisposed to exaggerations too. If the crime-related data is irregular, no one cannot predict crime exemplars correctly, and this ultimately guides to hapless crime forecasting.

6.2 Future Work Plan

In the future, we will try to mitigate the composition of our CriPred algorithm to increase performance by incorporating some other interpretable variables and correlating them to real-life variables to make the model more accountable and comprehensive.The more data or information we own, the more trustworthy the consequences will be. That is why in the future, we will examine to build an adequate data scraping or data acquisition tool and efficient data merging tool, which will help to enhance the overall performance of our solution.Since crime is a relative matter and the definition of crime varies from the different region to region. In the future, we will try to provide a comprehensive solution based on the region and the meaning of crime in that region.

Bibliography

[1] Simple. flexible. powerful ecosystem. https://keras.io/, Last Accessed Date: JUNE 3, 2021.

[2] Agarwal, P., Sharma, M., and Chandra, S. Comparison of machine learning approaches in the prediction of terrorist attacks. 1–7.

[3] Almanie, T., Mirza, R., and Lor, E. Crime prediction based on crime types and using spatial and temporal criminal hotspots. *arXiv preprint arXiv:1508.02050* (2015).

[4] Azeez, J., and Aravindhar, D. J. Hybrid approach to crime prediction using deep learning. In *2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI)* (2015), IEEE, pp. 1701–1710.

[5] Bharathi, A., and Shilpa, R. A survey on crime data analysis of data mining using clustering techniques. *International Journal of Advance Research in Computer Science and Management Studies 2*, 8 (2014), 9–13.

[6] Birant, D., and Kut, A. St-dbscan: An algorithm for clustering spatial–temporal data. *Data & knowledge engineering 60*, 1 (2007), 208–221.

[7] Brownlee, J. What is deep learning? https://machinelearningmastery.com/what-is-deep-learning, Last Accessed Date: JUNE 3, 2021.

[8] Brownlee, J. What is machine learning? https://deepai.org/machine-learning-glossary-and-terms/machine-learning, Last Accessed Date: JUNE 3, 2021.

[9] Chackravarthy, S., Schmitt, S., and Yang, L. Intelligent crime anomaly detection in smart cities using deep learning. In *2018 IEEE 4th International Conference on Collaboration and Internet Computing (CIC)* (2018), IEEE, pp. 399–404.

[10] Cohn, E. G. Weather and crime. *The British Journal of Criminology 30*, 1 (1990), 51–64.

[11] Cohn, E. G. The prediction of police calls for service: The influence of weather and temporal variables on rape and domestic violence. *Journal of Environmental Psychology 13*, 1 (1993), 71–83.

[12] Emig, M., and Heck, R. *Crime Analysis: A Sel. Bibl. Marjorie Kravitz, Supervising Ed.* National Criminal Justice Reference Service, 1980.

[13] Feng, M., Zheng, J., Ren, J., Hussain, A., Li, X., Xi, Y., and Liu, Q. Big data analytics and mining for effective visualization and trends forecasting of crime data. *IEEE Access 7* (2019), 106111–106123.

[14] Goel, M., Sharma, N., and Gurve, M. K. Analysis of global terrorism dataset using open source data mining tools. 165–170.

[15] Kang, H.-W., and Kang, H.-B. Prediction of crime occurrence from multimodal data using deep learning. *PloS one 12*, 4 (2017), e0176244.

[16] Kumar, V., Mazzara, M., Messina, A., and Lee, J. A conjoint application of data mining techniques for analysis of global terrorist attacks: Prevention and prediction for combating terrorism. 146–158.

[17] Lekha, K. C., and Prakasam, S. Data mining techniques in detecting and predicting cyber crimes in banking sector. 1639–1643.

[18] Lochner, L. Education and crime. In *The Economics of Education*. Elsevier, 2020, pp. 109–117.

[19] Mo, H., Meng, X., Li, J., and Zhao, S. Terrorist event prediction based on revealing data. 239–244.

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

[21] Rachel Boba, R. B. S. *Crime Analysis and Crime Mapping*. Sage Publications, 2005.

[22] Sivaranjani, S., Sivakumari, S., and Aasha, M. Crime prediction and forecasting in tamilnadu using clustering approaches. In *2016 International Conference on Emerging Technological Trends (ICETT)* (2016), IEEE, pp. 1–6.

[23] Stec, A., and Klabjan, D. Forecasting crime with deep learning. *arXiv preprint arXiv:1806.01486* (2018).

[24] Yamini, M. P. C. A violent crime analysis using fuzzy c-means clustering approach. *ICTACT Journal on Soft Computing 9*, 3 (2019), 1939–1944.

[25] Zhao, X., and Tang, J. Exploring transfer learning for crime prediction. In *2017 IEEE International Conference on Data Mining Workshops (ICDMW)* (2017), IEEE, pp. 1158–1159.

[26] Zhao, X., and Tang, J. Modeling temporal-spatial correlations for crime prediction. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management* (2017), pp. 497–506.