1. INTRODUCTION

India has a large problem of illicit drug trade, both with smuggling of pharmaceuticals, and synthetic drugs. Currently, citizens can only report suspicious information related to illicit drug trade in-person or using a telephone connection, thus jeopardizing their safety and privacy. Verification of said information is difficult and resource-consuming, as it requires a manual review of each report. Such systems are open to sabotage by internal or external malicious actors, further bogging down the process. The proper identification of drug-related crimes and red flags such shell companies are identifiable today through an AI-enhanced technologies. Over the past two decades, the world of drug trafficking has seen a significant upheaval. Drug trafficking of illegal substances such cocaine, methamphetamines, designer drugs, restricted prescription medicines, fentanyl, and heroin is still prevalent worldwide [1].

Social media sites like Instagram and Twitter have developed into significant distribution channels for illegal narcotics. Fighting the online trade of illegal drugs has grown dependent on the detection of illicit drug trafficking. However, the legal status frequently varies geographically and over time; even the same substance may be affected differently by federal and state law.

In this paper, the solution aims to resolve all the above issues in a quick and timely manner.

2. PROBLEM STATEMENT

India is positioned between two of the "three primary producing locations" for opium in the globe, creating a crisis of illegal drugs. However, cross-border trafficking accounts for only a portion of the illegal drug trade that passes through India. According to the department of pharmaceuticals in the Ministry of Chemicals and Fertilizers, India's pharmaceutical industry is "the largest provider of generic medications internationally." However, there have also been rumours of illegal trading in chemical precursors and covert production of synthetic pharmaceuticals. Drug trafficking cannot yet be reported digitally. The only option is to contact the local police station or the Narcotics squad. This is not practical, though, as the majority of people won't be willing to reveal their identities or engage in physical activity. The public being able to submit evidence of drug trafficking in their area anonymously is the proposed solution. If the system is anonymous, there is a chance that there may be a lot of false alerts, some of which may be deliberate. As a result, the solution should make it possible to filter out the false alarms using any type of machine learning model. The solution should make it possible for the general public to upload any pertinent information regarding the findings, and it should be simple for users to use in many languages. supports. [2]

Workflow Diagram

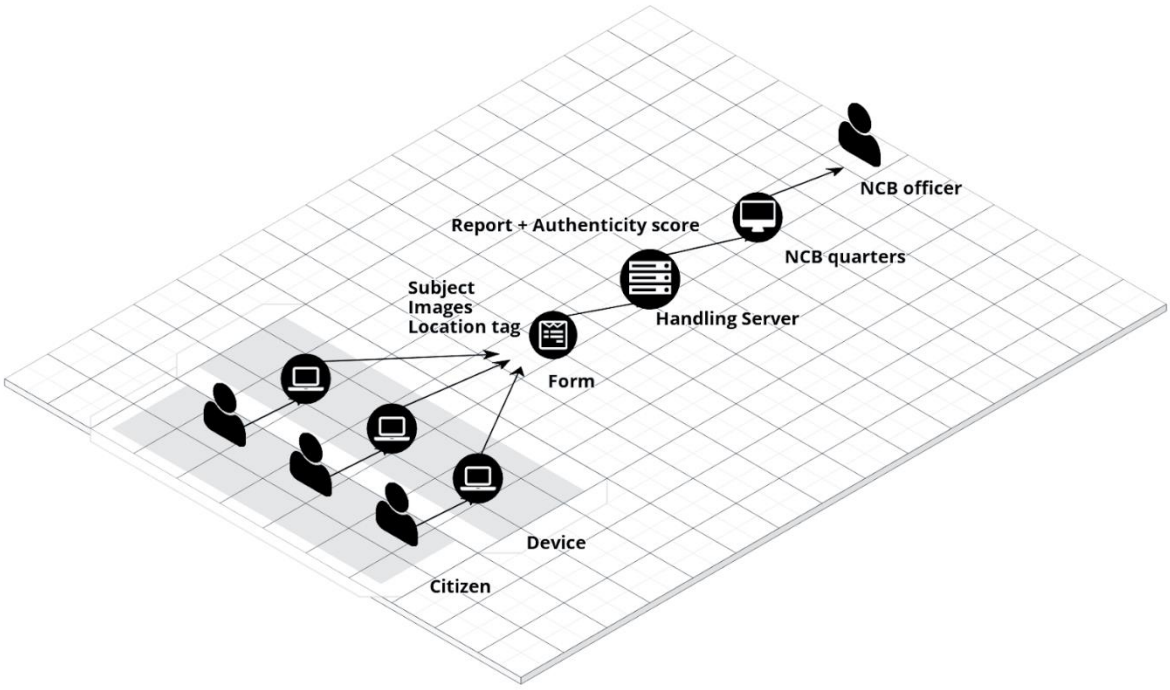


Fig 1. Abstract level process flowchart

3. METHODOLOGY

Generally, public does not report information on illegal drug trafficking as it may compromise their safety and security. Our application aims to solve this by keeping the informant's identity anonymous throughout the entire reporting process.

Users can send in potential evidence in the from of **text (description of incident), image (proof), and/or location.** Sensitive information such as name, address, phone number, etc. will not be considered.

The login functionality will only utilize the username (unique) and password which will be used as a reference to check the nature of evidence and threshold of drug finding reports.
The information sent in by the users is duly checked for its authenticity via the AI/ML algos. A genuinity score is attached with each submission that rates its authenticity, thus making it easier for the NCB to narrow down on prospective data.

Authenticity of the evidence will be evaluated on the basis of:

1. **Text** (Fake websites, Missing information, Frequency of highlighted words).

2. **Image** (Barcode, Vehicle no. Plate, Pills Registration no. and Prescription, Drug color and packaging, ammunition, and other relevant background details)

The solution comprises of the following:

i. Quick identification and categorization of illegal drugs based on text, image and location tag evidence input by user.

ii. Identification of both synthetic and pharmaceutical illicit drugs through image processing.

iii. Checking the authenticity of user reports and assigning a ***genuinity score*** attached to report submissions.

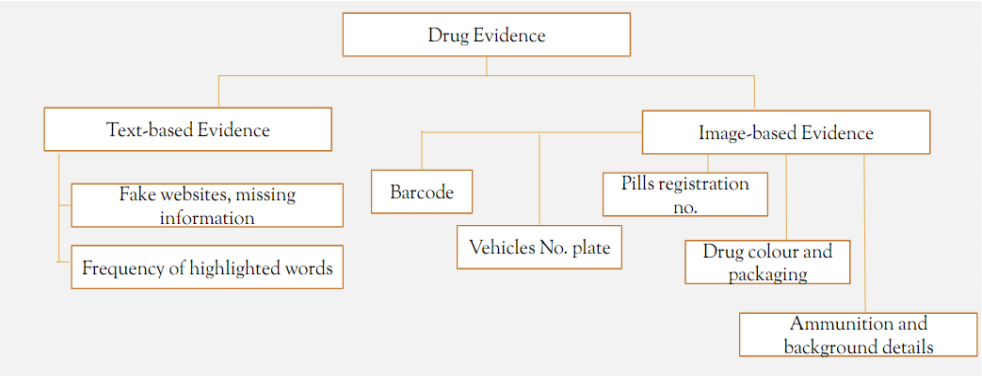


Fig 2. Classification of Drug Evidence Reports

4. Data Collection and Data Pre-processing

Additionally, all datasets have shortcomings. This is why the machine learning process relies so heavily on data preparation. Data preparation, in a nutshell, is a set of procedures for enhancing the machine learning capabilities of your dataset. In a larger sense, selecting the most effective data collection method is part of data preparation. And the majority of machine learning time is spent using these methods.

i. Constructing a toy dataset

A toy dataset is a fictitious dataset used for simple prediction model testing and exploratory data analysis (EDA). The table contains just fictitious information. The distributions of the data have been created in a way that makes statistical analysis easy. Text information can be acquired from a

Data Description

- ***key:*** These are the keywords extracted and used to check the frequency of highlighted words.
- ***class:*** Binary label (0 or 1) that determines the genuinity score.

ii. Data labelling

Data labelling, which involves giving meaning to digital data, is a crucial component of data pre-processing. For categorization purposes, input and output data are labelled, providing a learning foundation for subsequent data processing. For instance, the label "a dog" could have a dog image associated to it.

We now have a sufficient amount of data to create a clean, appropriate dataset that is representative (captures the most crucial information).

	key	class
2	narcotic narcotics	1
3	drugs	1
4	pshychotropic	1
5	substances	1
6	addiction	1
7	amphetamine	1
8	steroids	1
9	analgesics	1
10	antagonist	1
11	benzodiazepine	1
12	dopamine	1
13	drug abuse	1
14	drugged driving	1
15	electronic cigarette	1
16	hallucinations	1
17	illicit	1
18	injection drug use idu	1
19	mental disorder	1
20	methadone	1
21	drug	1
22	naloxone	1
23	overdose	1

148	amenable	0
149	anachronistic	0
150	audacious	0
151	avaricious	0
152	banal	0
153	benign	0
154	brazen	0
155	calumny	0
156	candid	0
157	castigate	0
158	caustic	0
159	construe	0
160	contrite	0
161	spam	0
162	crazy	0
163	joke	0
164	already	0
165	darling	0
166	winner	0
167	urgent	0
168	won	0
169	brother	0
170	cash	0

Fig 3. Snapshot of toy dataset created in Microsoft Excel

iii. Data Availability Bias

The availability bias is a result of people's innate propensity to depend excessively on the info that is most easily accessible. It can also happen when algorithms in use in healthcare give more weight to data that is more easily accessible but does not accurately represent the intended demographic.

What we recall usually has an impact on our decisions. Many factors, including as beliefs, expectations, sentiments, and emotions as well as variables like frequency of exposure, affect what we recall. The use of media (such as radio, television, and the Internet) has a significant impact. Rare occurrences are made more evident to us because of the extensive media coverage they receive. Therefore, we are more likely to remember it, particularly in the short-term.

The toy dataset created includes some degree of availability and human bias because it has been made taking in account of previously curated research papers, blog posts, familiar crime incidents and personal judgement.

All data included is fictional and has not been created based on real-life drug trafficking cases but more so based on probable drug trafficking encounters that might have happened in past. A more concise database can be made based on NCB’s database for drug trafficking encounters, approved reports, evidence or situations analysis categories or parameter.

iv. Performing EDA on a sample drug report

The pre-processing of data is a critical stage in the development of a machine learning model. Data that isn't initially clean or in the model's required format can lead to inaccurate results. Pre-processing involves transforming data into the format we need. It is used to handle the dataset's noise, duplication, and missing values. Activities like importing datasets, partitioning

datasets, attribute scaling, etc. are all part of data pre-processing. Pre-processing the data is necessary to increase the model's accuracy [3].

a) *Information about the type of data in row and columns*

```
In [1]: import numpy as np
        import pandas as pd

In [2]: data=pd.read_csv('drugs1.csv')

In [3]: data.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 211 entries, 0 to 210
Data columns (total 2 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   key     211 non-null    object
 1   class   211 non-null    int64
dtypes: int64(1), object(1)
memory usage: 3.4+ KB
```

b) *Finding out the number of null values*

```
In [5]: data.tail()
```

Out[5]:

	key	class
206	quixotic	0
207	spendthrift	0
208	taciturn	0
209	wary	0
210	beautiful	0

```
In [213]: data.isnull().sum()

Out[213]: key      0
          class    0
          dtype: int64

In [214]: data = data.sample(frac = 1)
```

5. Implementation of Machine Learning Models

Text – Based Filtering

We will performing NLP using the following machine learning algorithms: Naïve Bayer, LSTM, and Random Forest Classifier.

Firstly, we will take the input as a sample drug report from the government and classify it as genuine or not. The pre-processing in text-based filtering are divided into the following stages: Tokenization, Normalization(Stemming and Lemmatization), removing stop words, POS(Part of Speech) and feature extraction(BOW – Bag of Words and TF-IDF).

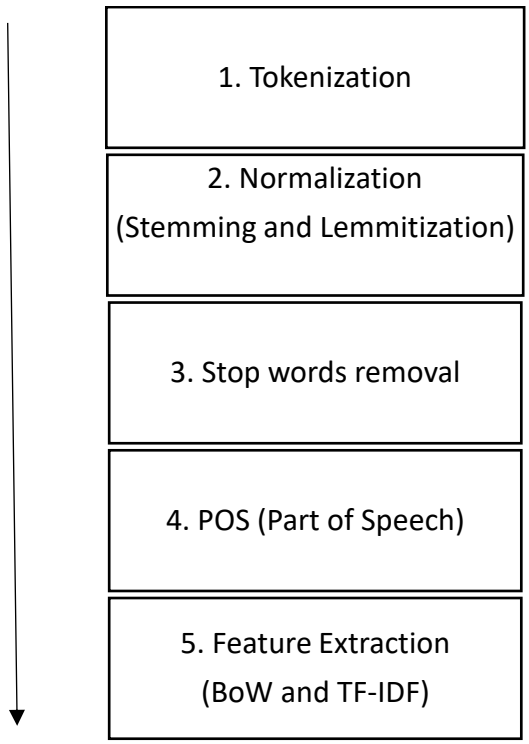


Fig 4. Text- based classification process flowchart

a) Tokenization

Tokenization is essentially splitting a phrase, sentence, paragraph, or an entire text document into smaller units, such as individual words or terms. Each of these smaller units are called tokens.Before processing a natural language, we need to identify the *words* that constitute a string of characters. That's why tokenization is the most basic step to proceed with NLP (text data). **This is important because the meaning of the text could easily be interpreted by analyzing the words present in the text.**

b) Normalization (Stemming and Lemmatization)

and frequently stems undesired words, it is not employed for manufacturing. Lemmatization was a new method that was introduced to the market to address the

issue. There are several different kinds of stemming algorithms, including the Porter and Snowball stemmers. Porter stemmer is a popular tool that is available in the NLTK library. Lemmatization stems words into root words similarly to stemming, although it varies in that.

```
In [105]: corpus = []
          for i in range(len(sentences)):
              review = re.sub('[^a-zA-Z]',' ', sentences[i])
              review=review.lower()
              corpus.append(review)

In [106]: ## REMOVING THE STOPWORDS
          corpus

Out[106]: ['drug trafficking is the worldwide practice of distributing drugs and other substances which are legally banned under narcotic
          s and associated laws ',
           'it includes the cultivation  manufacture  distribution  and sale of controlled substances ',
           'in india  the use of drugs in society especially among the younger generation has increased in the past decade and continuous
          to increase ',
           'this has direct and hidden effects on the future growth of the country ',
           'this paper discusses the problem of drug trafficking in india  their entry inside the country through the country s border an
          d the various drugs which are seized in the country ',
           'it also discussed on how the investigation of drug related cases can be improved by implementing different approaches to redu
          ce the illegal trade of drugs ',
           'while drug traffi cking has engulfed all the parts of india  so are the efforts of offi cers of dri to thwart such activities
          happening anywhere in india  even at the risk to their lives ',
           'in april       when the whole manipur was in deep sleep  at around      am  a dedicated team of dri offi cers was deployed in
          a dense forest to nab the drug smugglers ',
           'fully aware of the repercussions of any mistake  the offi cers with their wit and commitment and with active support of assam
```

Fig 5. Applying Stemming and Lemmatization on sample drug report

```
In [97]: type(sentences)

Out[97]: list

In [98]: ## APPLYING STEMMING - to find the base root words
         stemmer = PorterStemmer()

In [99]: ## LIKE THIS
         stemmer.stem('thinking')

Out[99]: 'think'

In [100]: ## LEMMITIZATION - find out words from stem of proper meaning
          from nltk.stem import WordNetLemmatizer

In [101]: lemmatizer = WordNetLemmatizer()

In [102]: ## LIKE THIS
          lemmatizer.lemmatize('goes')

Out[102]: 'go'
```

c) Stop Words Removal

```
In [107]: ### LIST OF COMMON ENGLISH STOPWORDS
          stopwords.words('english')

          'itself',
          'they',
          'them',
          'their',
          'theirs',
          'themselves',
          'what',
          'which',
          'who',
          'whom',
          'this',
          'that',
          "that'll",
          'these',
          'those',
          'am',
          'is',
          'are',
          'was',
          'were',
```

```
In [111]: corpus

Out[111]: ['drug trafficking worldwide practice distributing drug substance legally banned narcotic associated law',
 'includes cultivation manufacture distribution sale controlled substance',
 'india use drug society especially among younger generation increased past decade continuous increase',
 'direct hidden effect future growth country',
 'paper discus problem drug trafficking india entry inside country country border various drug seized country',
 'also discussed investigation drug related case improved implementing different approach reduce illegal trade drug',
 'drug traffi cking engulfed part india effort offi cer dri thwart activity happening anywhere india even risk life',
 'april whole manipur deep sleep around dedicated team dri offi cer deployed dense forest nab drug smuggler',
 'fully aware repercussion mistake offi cer wit commitment active support assam rifl e daredevil drug enforcement operation for
est area clutch extremist',
 'darkness spotted two group people shoulder load walking within suffi cient distance',
 'mission planned manner group smuggler covered offi cer appropriate moment ensuring indeed target looking offi cer intercepted
person group seized around lakh tablet methamphetamine commonly known yaba tablet',
 'drug may legal example alcohol caffeine tobacco illegal variety',
 'latter category includes usage self prescribed medicine induce intoxication certain intoxicating inhalant cannabinoids like m
arijuana hashish opium derivative like heroin brown sugar stimulant like cocaine called designer drug club rave party drug',
 'latter basically combination one kind traditional chemical drug giving enormous high',
 'besides several dissociative hallucinogen anabolic steroid also exist',
 'purpose article confine one variety dreadful opioids particularly heroin nothing derivative opium',
 'know golden crescent comprising afghanistan part pakistan iran shaped like crescent highest producer opium',
 'next deadly area called golden triangle triangle shaped geographical area lying region myanmar cambodia lao including part th
ailand well',
 'region sit india shoulder either side']
```

Fig 6. List of common English stop words

Fig 7.

```
In [112]: ##Bag OF Words
          from sklearn.feature_extraction.text import CountVectorizer
          cv=CountVectorizer(binary=True,ngram_range=(2,3))

In [113]: X=cv.fit_transform(corpus)

In [114]: cv.vocabulary_

          'thwart activity': 444,
          'activity happening': 2,
          'happening anywhere': 220,
          'anywhere india': 15,
          'india even': 253,
          'even risk': 189,
          'risk life': 397,
          'drug traffi cking': 169,
          'traffi cking engulfed': 452,
          'cking engulfed part': 80,
          'engulfed part india': 181,
          'part india effort': 358,
          'india effort offi': 250,
          'effort offi cer': 176,
          'offi cer dri': 340,
          'cer dri thwart': 69,
          'dri thwart activity': 150,
          'thwart activity happening': 445,
          'activity happening anywhere': 3,

In [115]: corpus[0]

Out[115]: 'drug trafficking worldwide practice distributing drug substance legally banned narcotic associated law'
```

Corpus after removing stopwords and performing lemmatization

d) BOW (Bag of words)

Fig 8. Applying feature extraction with BoW

e) TF-IDF Vectorizer

TF-IDF or Term Frequency Inverse Document Frequency is very common algorithm to transform text into a meaningful representation of numbers which is used to fit machine algorithm for prediction. **tf–idf** or **TFIDF**, short for **term frequency-inverse document frequency**, is a metric that quantifies the significance of a word in a document inside a corpus or collection. To account for the fact that some words are

used more frequently than others overall, the tf-idf value rises according to the number of times a word appears in the document and is offset by the number of documents in the corpus that contain the term. Less frequently occurring terms in the document corpus are given higher scores by TF-IDF. When both IDF and TF values are high, or when a term is uncommon throughout the manuscript but common inside one section, the TF-IDF value is high. Additionally, TF-IDF disregards the semantic significance of the words.

Counts. Count how many times each term is used.

Frequencies. Calculate the frequency that each word appears in a document out of all the words in the document.

TF(t) = (Number of times term t appears in a document) / (Total number of terms in the document).

IDF(t) = log_e(Total number of documents / Number of documents with term t in it).

TFIDF works by proportionally increasing the frequency with which a term appears in the document, but it is balanced by the frequency with which it appears in other documents. As a result, frequent terms like "this," "are," etc. that appear in all of the documents are not assigned a very high score. However, a word that appears too frequently in a small number of the documents will be

Term Frequency: Term frequency is defined as the number of times a word (i) appears in a document (j) divided by the total number of words in the document.

Inverse Document Frequency: Inverse document frequency refers to the log of the total number of documents divided by the number of documents that contain the word. The logarithm is added to dampen the importance of a very high value of IDF [7].

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{i,j}}$$

TFIDF is computed by multiplying the term frequency with the inverse document frequency.

$$idf(w) = log(\frac{N}{df_t})$$

$$w_{i,j} = tf_{i,j} \times \log\left(\frac{N}{df_i}\right)$$

Scikit-learn provides you a pre-built TF-IDF vectorizer that calculates the TF-IDF score for each document's description, word-by-word.

```
In [117]: ### TFIDF
          from sklearn.feature_extraction.text import TfidfVectorizer
          cv=TfidfVectorizer(ngram_range=(1,1),max_features=10)
          X=cv.fit_transform(corpus)

In [118]: corpus[0]

Out[118]: 'drug trafficking worldwide practice distributing drug substance legally banned narcotic associated law'

In [119]: X[0].toarray()

Out[119]: array([[0., 0., 0., 1., 0., 0., 0., 0., 0., 0.]])

In [120]: print(X)

            (0, 3)        1.0
            (2, 5)        0.830927778273119
            (2, 3)        0.5563802901739946
            (3, 2)        1.0
            (4, 2)        0.9088752175887859
            (4, 5)        0.24954021211967684
            (4, 3)        0.33417887633452215
            (5, 3)        1.0
            (6, 1)        0.36168991220121594
            (6, 7)        0.36168991220121594
            (6, 9)        0.3955117125616825
            (6, 5)        0.7233798244024319
            (6, 3)        0.24218366934577815
            (7, 1)        0.6390918401874626
            (7, 7)        0.6390918401874626
            (7, 3)        0.4279290123508869
            (8, 0)        0.6130157990574606
```

Fig 9. Applying feature extraction with tf-idf

i. Naïve Bayes Classifier

```
In [224]: x_train, x_test, y_train, y_test = train_test_split(x.toarray(), y, test_size=0.12, random_state=42)

In [225]: model = MultinomialNB()

In [226]: model.fit(x_train,y_train)

Out[226]: MultinomialNB()

In [227]: print("The score for MultinomialNB model over the given data is ",model.score(x_test,y_test)*100)

          The score for MultinomialNB model over the given data is  53.84615384615385
```

Fig 10. Accuracy of the MultinomialNB Model

```
In [241]: text4 = "Here in lucknow narcotic and steroids drugs are being trafficked in gomti nagar"
data4 = [text4]
data4=cv.transform(data4).toarray()
prediction4 = model.predict(data4)
predict_score4=model.predict_proba(data4)
print(predict_score4)
print(prediction4)
if prediction4[0] == 1:
    print("It is a Genuine Reason with ",predict_score4[0][1]*100,"% probability")
else:
    print("It is Not a Genuine Reason with ",predict_score4[0][0]*100,"% proability")

[[0.00577655 0.99422345]]
[1]
It is a Genuine Reason with  99.42234475225906 % probability
```

Fig 11. Multinomial Naïve Bayes model prediction result

ii. Random Forest Classifier

The broad category of ensemble-based learning techniques includes random forest classifiers [30]. They have shown to be incredibly effective across a wide range of domains and are easy to adopt and quick to use. The main idea behind the random forest method entails building a lot of "simple" decision trees during training and using a majority vote (mode) across them for classification.

This voting method corrects for the unfavourable tendency of decision trees to overfit training data, among other things. Random forests apply the general bagging strategy to each individual tree in the ensemble during the training phase.No pruning is done as the trees grow. A free parameter that is easily learned automatically utilising the so-called out-of-bag error is the ensemble's number of trees.

Random forests are well-liked in part because of their simplicity on the one hand, and generally strong performance on the other, much like in the case of naive Bayes- and k-nearest neighbor-based algorithms. The final trained model's structure is somewhat unpredictable with random forests, in contrast to the first two methods.

The decision tree is the main building block of random forest classifiers. The independent variables (or features) in a data collection are used to construct the hierarchical structure known as the decision tree. The decision tree's nodes are divided up based on a measure connected to a subset of the features. The random forest is made up of a set of bootstrap samples that are produced from the original data set and a collection of decision trees. The entropy (or Gini index) of a chosen subset of the characteristics determines how the nodes are divided..

```
In [242]: from sklearn.ensemble import RandomForestClassifier
          model2 = RandomForestClassifier(n_estimators = 100)
          model2.fit(x_train,y_train)

Out[242]: RandomForestClassifier()

In [243]: print("The score for Random Forest classifier model over the given data is ",model2.score(x_test,y_test)*100)

          The score for Random Forest classifier model over the given data is  80.76923076923077
```

Fig 12. Accuracy of the Random Forest Classifier Model

```
In [244]: text4 = "Here in lucknow narcotic and steroids drugs are being trafficked in gomti nagar"
          data4 = [text4]
          data4=cv.transform(data4).toarray()
          prediction4 = model2.predict(data4)
          predict_score4=model2.predict_proba(data4)
          print(predict_score4)
          print(prediction4)
          if prediction4[0] == 1:
              print("It is a Genuine Reason with ",predict_score4[0][1]*100,"% probability")
          else:
              print("It is Not a Genuine Reason with ",predict_score4[0][0]*100,"% proability")

          [[0.02 0.98]]
          [1]
          It is a Genuine Reason with  98.0 % probability

In [245]: text4 = "kailash ritik afrin muskan vishnu"
          data4 = [text4]
          data4=cv.transform(data4).toarray()
          prediction4 = model2.predict(data4)
          predict_score4=model2.predict_proba(data4)
          print(predict_score4)
          print(prediction4)
          if prediction4[0] == 1:
              print("It is a Genuine Reason with ",predict_score4[0][1]*100,"% probability")
          else:
              print("It is Not a Genuine Reason with ",predict_score4[0][0]*100,"% proability")
```

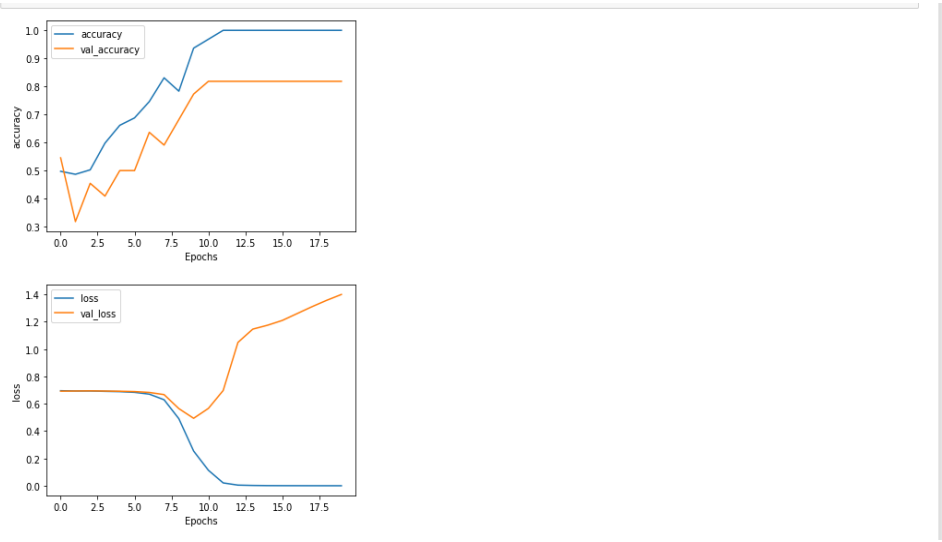
Fig 13. Random Forest Classifier Model prediction result

iii. Neural network for binary classification using LSTM

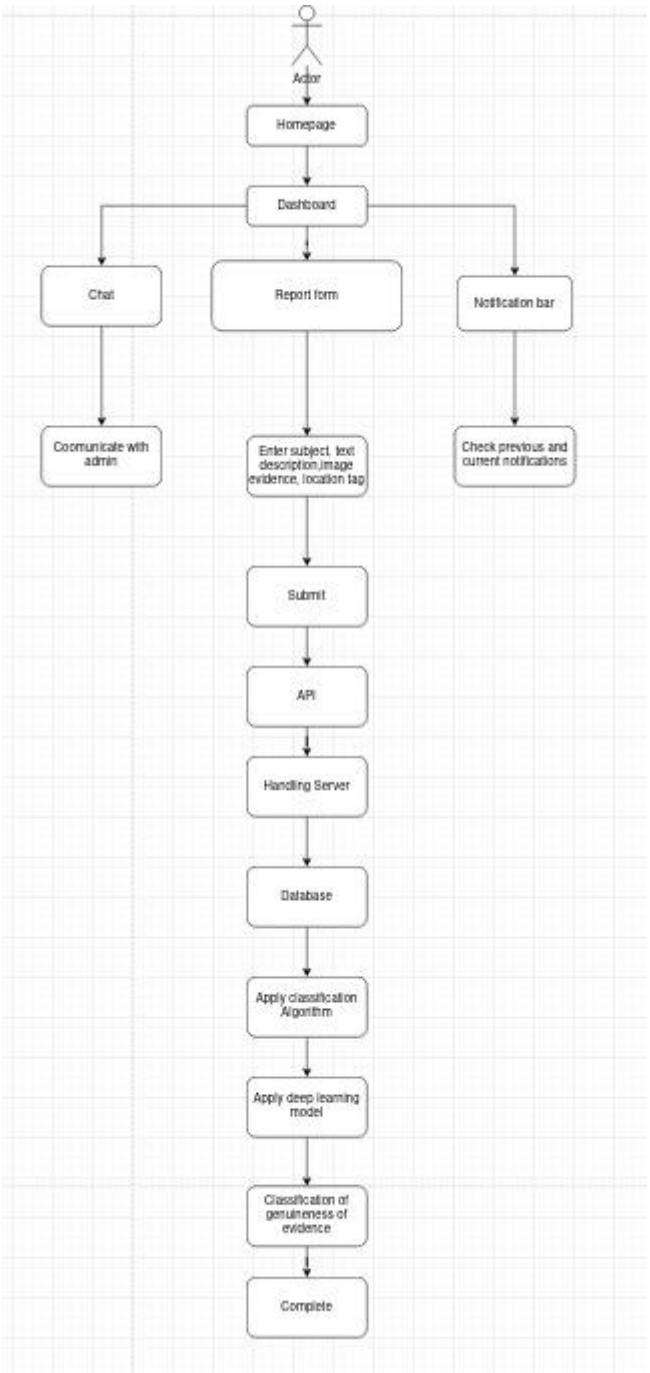
```
In [166]: text4 = "Here in lucknow narcotic and steroids drugs are being trafficked in gomti nagar"
          data4 = [text4]
          data4=tokenizer.texts_to_sequences(data4)
          data4=pad_sequences(data4,maxlen=max_length,truncating='post')
          prediction4 = model.predict(data4)
          print(prediction4)
          if prediction4 > 0.5: #threshold value for sigmoid activation
              print("It is a Genuine Reason")
          else:
              print("It is Not a Genuine Reason")

          [[0.9996106]]
          It is a Genuine Reason

In [167]: text4 = "afrin muskan kailash ritik vishnu"
          data4 = [text4]
          data4=tokenizer.texts_to_sequences(data4)
          data4=pad_sequences(data4,maxlen=max_length,truncating='post')
          prediction4 = model.predict(data4)
          print(prediction4)
          if prediction4 > 0.5: #threshold value for sigmoid activation
              print("It is a Genuine Reason")
          else:
              print("It is Not a Genuine Reason")

          [[0.00035964]]
          It is Not a Genuine Reason

In [168]: text4 = "you won 100rs in the game"
          data4 = [text4]
          data4=tokenizer.texts_to_sequences(data4)
          data4=pad_sequences(data4,maxlen=max_length,truncating='post')
          prediction4 = model.predict(data4)
          print(prediction4)
          if prediction4 > 0.5: #threshold value for sigmoid activation
              print("It is a Genuine Reason")
          else:
              print("It is Not a Genuine Reason")
```

Fig 14. Relation between accuracy and result of LSTM Neural Network

Fig 15. LSTM Deep Neural Network Prediction



6. Final Prototype



Result

This paper compares the classification results and accuracy of Naïve Bayes Classifier, Random Forest Classifier, and LSTM deep neural network The results show that the three machine learning algorithms perform well in dealing with the binary classification problem, but compared with the other two models, the LSTM deep learning model has higher accuracy.

The performance of the above mentioned three algorithms have been compared with each other on the same dataset. The results from the implementation has showed that LSTM has got the best overall performance values compared to the other algorithms. In general, the effectiveness and the efficiency of a machine learning solution depend on the nature and characteristics of data and the performance of the learning algorithms.

The main aim of this implementation was to showcase the scope of the defining a comprehensive solution for the drug trafficking problem statement. As the use case mainly pertains to collecting real-world data from various sources such as verified crime reports or databases handled by government, it is safe to say that the capabilities of the various machine learning techniques were tested well with the dataset. Moreover, this paper aims to provide a comprehensive over view on how the powerful machine learning would work in a scenario where they have to applied unambiguous decision-making related to crime-related cases such as drug trafficking and evidence-matching.

S.No.	Algorithms	Accuracy Score
1	Multinomial Naïve Bayes Classifier	~53.7%
2	Random Forest Classifier	~80.7%
3	LSTM Deep Neural Network	~90.0%

Conclusion

As the country progresses forward, there exists some critical issues such as illegal drug trafficking that arise within the crime timeline. These acts of illegal crime happen everywhere around us but somehow go unnoticed or are just known about at the right time. However, even though these issues have been on the frontline since a while, there no concise methods in which they can addressed or solved. In this paper, a prototype has been proposed which has the potential to be applied in such a real-world domain application to solve such problems. It has the capabilities to become a comprehensive data-driven application enhanced by artificial intelligence and machine learning.