Multimedia Tools and Applications (2021) 80:28061–28085
https://doi.org/10.1007/s11042-021-11050-4

A knowledge centric hybridized approach for crime classification incorporating deep bi-LSTM neural network

Gerard Deepak[1] · S. Rooban[1] · A. Santhanavijayan[1]

Received: 21 August 2020 / Revised: 7 January 2021 / Accepted: 5 May 2021 /
Published online: 28 May 2021

Abstract
In recent years, the crime rate has increased considerably and there is a need to properly identify the different types of crimes so that it can be tackled. In this paper, a Bi-LSTM neural network for classification is proposed that classifies the different types of crime on data collected from Google News and Twitter. The data is pre-processed and an initial step of labeling is performed with the help of Fuzzy c-means algorithm and Term Frequency – Inverse Document Frequency vectors. GloVe word embeddings were performed for feature extraction. Dynamically generated ontologies with minimal human supervision using a weighted graph modeled from Google News and Social Web like Twitter has been encompassed in order to enhance the quality of crime classification. The proposed method has proven, after experiments, to achieve evaluation metrics better than the existing methods; evaluated on four different datasets and compared with four different methods with an increase in Accuracy and decrease in FNR for four distinguished datasets.

Keywords Bi-LSTM · Crime classification · Crime forensics · Deep learning in crimes · Knowledge centric approach · Ontological model

1 Introduction

Crime is no longer observed as a solitary issue in a given society, it is has become a fundamental piece of understanding a country's social, political, and economic circumstances. India is also one of those societies. Crime in India has become much more complex in the past years and is ever increasing in various forms such as murder, drug trafficking, money

✉ Gerard Deepak
gerard.deepak.cse.nitt@gmail.com

[1] Department of Computer Science and Engineering, National Institute of Technology, Tiruchirappalli, Tiruchirappalli, India

laundering, fraud, and poaching, etc. As stated by the National Crime Records Bureau [5], major forms of crime have increased by a total of 1.3% in India, in a gap of only one year recorded in 2018 with 5 million new recorded crimes. Therefore, the analysis of crime has become crucial as it can help law enforcement agencies to find patterns, make predictions, and always be a step ahead.

Crime analysis has become one of the most important subjects of interest because it is important to generalize the specific forms of crime patterns in regular intervals of time, to optimize the consumption of limited police resources in a state-wise manner as well as in the whole country jointly, between the various police organizations, at the same time also being dynamic in identifying and prevention, to meet the policy requirements of the continuously changing society. It becomes difficult to accurately and efficiently analyze the ever-increasing collections of data relevant to criminology. The enormous spatial differences and the intricacy of crime relations have made analyzing crime data more difficult. The exchange of information among police agencies became very time consuming and therefore, not available in the time of need.

With the emergence of online news sources and social media, there has been an explosion in data related to crime. This can be used for analysis as it becomes easier to access and also there is a continuous flow of a variety of data. Twitter is a classic data source for crime-related news [12]. There are more than a million users in India, who publicly post tweets regarding everyday events, emotions, and various other topics; its data is formulated and distributed in real-time for free, and the tweets by users are often tagged with the precise location including co-ordinates. Also, Google News provides news from many other news sources thus providing a wide range and means to gather more data.

A problem arises as to how to handle the vast types of crimes, which to give more importance to, how they're related, and affect one another. In this paper, with the help of a Bi-LSTM model [15], a method of classification is proposed that classifies the crime-related news into types, and based on those, an ontological representation has been created that records frequency and spatial data. This ensures that in the future, crime statistics can be updated in a fast and efficient manner, and also statistical methods of analytics can be performed quite easily. There are a variety of techniques for data mining in crime data such as sequential pattern mining, clustering techniques, string comparison, association rule mining, and classification [8]. But LSTM has been proven to be the most effective in classifying sequential data.

Motivation: Most of the existing work take an unsupervised approach to the problem which incorporates text classification algorithms. This works only to an extent as the algorithms do not make use of the crime-related specifications required for efficiency and this might not be systematic in terms of accuracy because of the increase in data. Also, very few knowledge bases of ontologies of crime are present currently especially ones that give a detailed understanding of documents and this insufficient knowledge slows down law enforcement agencies to work in preventing and predicting crime. There is a need for an advanced classification model that can efficiently perform the classification of crime based on ontological representations using the current trending and the historical data from the social media, that can be used for affirmation and prediction of crime-related forensics.

Contributions: To overcome the problem of identifying, differentiating, and classifying the different types of crime, a Bi-LSTM neural network model is proposed. The standard metadata was Google news and Twitter. For training the model, initially Fuzzy c-means clustering with TF-IDF vectors were considered for labeling the data. The proposed system can also work on other datasets with proper pre-processing. The goal of the approach is to reduce the cognitive learning load and improve the credibility of the approach by infusing Natural Language

Processing (NLP), Ontologies, and Semantics. The metadata is aggregated by integrating knowledge from Google News and the Twitter API based on several events associated with the standard dataset. Experimentations are conducted on four distinct and standard datasets. An application based on hybridization of semantic infused learning has been put forth in the paper. The goal of this research is to reduce the learning load by supplying auxiliary Ontological Knowledge built from the metadata and promote semantic infused learning to promote eXplainable AI. An ontological depiction is constructed in the form of a weighted graph that gives the types of crime in a location in addition to all the relevant details such as the number of victims, accused, etc. The proposed system outperforms other methods in classification with an increase in accuracy of 9.2%, also a decrease in FNR of 0.11.

Organization: The rest of the paper is organized as follows. Section 2 comprises the Related Work. The Proposed Methodology is discussed in Section 3. Section 4 describes the implementation in detail. The Results and Performance Evaluation are depicted in Section 5. Finally, the paper is concluded in Section 6.

2 Related work

2.1 Deep learning driven approaches

Saha et al. (2020) [28] have developed a system that performs analytics as well as visualizations of the related crime information. They have used CNN-Bi-LSTM for extraction of crime and also the system can provide methods for crime patterns recognition and statistics of the data. Other works include Wang et al. (2020) [34], who had developed a hybrid model of Convolutional Neural Networks (CNN) and Bi-LSTM combined with the Attention Mechanism used to process and classify Chinese news data. Furthermore, Bhati et al. (2019) [6], have worked on the Indore police dataset to analyze, predict and classify crime in the city of Indore, using machine learning algorithms focusing on deep learning.

Das et al. (2019) [10], have worked on adaptive resonance theory neural network to perform classification on crime reports by clustering them into different types of crime. They have used GloVe vectorization for word embeddings of the documents. Sundhara et al. (2020) [31], have introduced a hybrid Recurrent Neural Networks (RNN) combining Extreme Machine Learning (ELM) structure for crime classification in particular locations. The Recurrent Neural Network was used to extract the features learned from LSTM and for classification, ELM was used at the end of all the layers. Anuar et al. (2015) [3], have researched the Artificial Bee Colony algorithm and ANNs to build a hybrid crime classification model. To address the ANN's local optima problem, the ABC algorithm was used as a learning procedure.

2.2 Other supervised learning driven approaches

Abbas et al. (2020) [1] have developed a framework that predicts the major types of crimes from social media sources mainly using Twitter data. For classification, they have used K-Nearest Neighbours, Naïve Bayes classifiers, and SVM. Other studies include the research by Zaidi et al. (2020) [35] where they have taken classification approaches to predict crime categories on the UCI Crime and Communities dataset. They have used Random Forests and Support Vector Machines. In the research of Kumar et al. (2020) [17], they have made use of

the KNN algorithm to identify crime rates. With this algorithm, they have managed to predict the time, location, and type of crime that may happen in the future. With this data, they have extracted behaviors for crime in a particular area which can help law enforcement agencies.

A study by Noormanshah et al. (2020) [23], presents a classification model constructed with the random forest algorithm that also performs ID labeling or tagging to non-structured and uncategorized data with textual analysis. The data and the trained model as inputs, the system can predict crime classifications also. Ramasubbareddy et al. (2020) [27] have collected crime data from major cities using the wamp server and have used the apriori algorithm to predict crime. Additionally, they have used a searching algorithm with decision trees and naive bayesian classifier to classify the types of crime in a given spatial location and a given time.

Other works by Alatrista-Salas et al. (2020) [2], in which they have implemented a method using MLP and SVM with document representations to form a multinomial classification that can differentiate between news containing crime relevant information and news which don't. It can also classify the types of crime. The work done by Lal et al. (2020) [18], includes using text mining approaches for classification of 369 tweets by analyzing Twitter data into two classes, one containing crime and the other not. Classification algorithms such as J48, Naive Bayesian Classifiers, Random Forest, etc. were used. Nair et al. (2019) [22], proposed a model that works on specific localities to predict the crime rate in the futures incorporating algorithms such as linear classification using decision trees and spatial analysis helping the law enforcement agencies to use resources efficiently on potential crime hotspots.

2.3 Clustering and data mining approaches

The work done by Gharehchopogh et al. (2020) [29] includes using the Elephant Herding Optimization (EHO) Algorithm and k-means clustering for detecting crime with the context of similarities of crime with each other. They have worked on the UCI Community and Crime Dataset. Das et al. (2020) [9] have proposed a graph-based clustering approach in which they partition the crime reports by extracting relations and patterns from the named entities found through a collection of Indian crime data. Another research by Sreejith et al. (2020) [30] utilizes graph mining techniques used on the collected data. This method can distinguish between the types of crime. They stored past information in the graph database and had made many inferences from it. Zhang et al. (2020) [36], have made use of data mining techniques mainly fuzzy association rule-based algorithm to discover patterns and relations between crime data acquired from two datasets, crime in Chicago from the years 2012–2017 and crimes in NSW from the years of 2008 to 2012.

2.4 Other alternative approaches

The studies of Hardy et al. (2020) [14] includes applying crime script analysis to the Madoff case, where they make use of the procedural entities of the intricate crime types and also to search for crime prevention techniques for law enforcement agencies. By using Digital crime news, Pangestuti et al. (2019) [24], and classification methods for an ontology-based text, they have retrieved information by analyzing the data. They have used testing methods such as Precision, Recall, F1 Score, and Accuracy. The study of Ghankutkar et al. (2019) [13], in which they have analyzed crime data in real-time by forming a web-system from online news articles using three different classifiers, it can classify between news articles that contain crime relevant information and articles that don't. Boppuru et al. (2019) [7], have analyzed the data

collected from various news sources for making classifications on the types of crime and also crime predictions in India specifically, the city of Bangalore with coordinates of the crime area, the crime which might happen there and also visualizations.

Other works by Thilagam et al. (2019) [32], where they have implemented an entity relationship-based system called Crime base; a knowledge base, to retrieve and use crime relevant image and text information from online news sources, mainly focusing on the reduction of duplicity as well as information loss. Wang et al. (2019) [33], have researched on Hierarchical Matching Network which is based on tree hierarchy and used it for crime classification. Abebe et al. (2020) [4], have introduced a generic Social-based Event Detection, Description, and Linkage framework. It takes data from social media and gives semantically meaningful events interconnected with spatial, temporal, and semantic relationships. Fares et al. (2018) [11], have taken a Lexical sentiment analysis approach for detecting user moods by introducing an unsupervised word-level knowledge graph-based LSA framework.

3 Proposed methodology

The architecture of the proposed system comprises of three different constituents of Data preparation, Classification, and Ontology construction. The aggregation of distinct knowledge from a set of varied sources for the Criminology as a Domain is quite interesting, where a Dynamic Ontology Modeling scheme based on the real world intertwining of popular domain-relevant tweets, hashtags, and auxiliary knowledge from news articles makes this approach quite concrete and unique for Crime Classification, and Analysis based on the information from the handle of several social networking sites thereby interlinking Social Web with the quickly changing Web 2.0 bridging the gap between Web 2.0 and Web 3.0. Figure 1 depicts the architecture of the Proposed System Architecture.

3.1 Data preparation

In this research, data were first collected from recent years from various news sources for training the classifier model from Google News as Google News India provides links to various other major news websites and also tweets from Twitter API. The reason for data preparation is to formulate the metadata which is a core constituent of the Semantic Web or the Web 3.0. The approach proposed is on the lines of semantic infused learning, where ontology modelling and the incorporation of the Ontologies as dynamic auxiliary knowledge into a learning infused semnatic strategy. Although the experimentations were conducted on a standard categorical dataset, the approach requires metadata which need to be dynamically generated as the Web 3.0 is not fully functional in the present day structure of the World Wide Web as the current structure of the Web is between Web 2.0 and Web 3.0, and very soon the organized intelligent Semantic Web with metadata will be fully modelled. A web scraper was implemented that returns the location, headlines, and respective context for the past one year from Google news. Tweets were acquitted through the API using a set of keywords consisting of several crime-related words, that were manually selected, for narrowing down the data to be collected.

The web scraper was implemented using Python incoporating several packages like Natural Language Tool Kit (NLTK) and the set of keywords were used for reference. In the scraper,

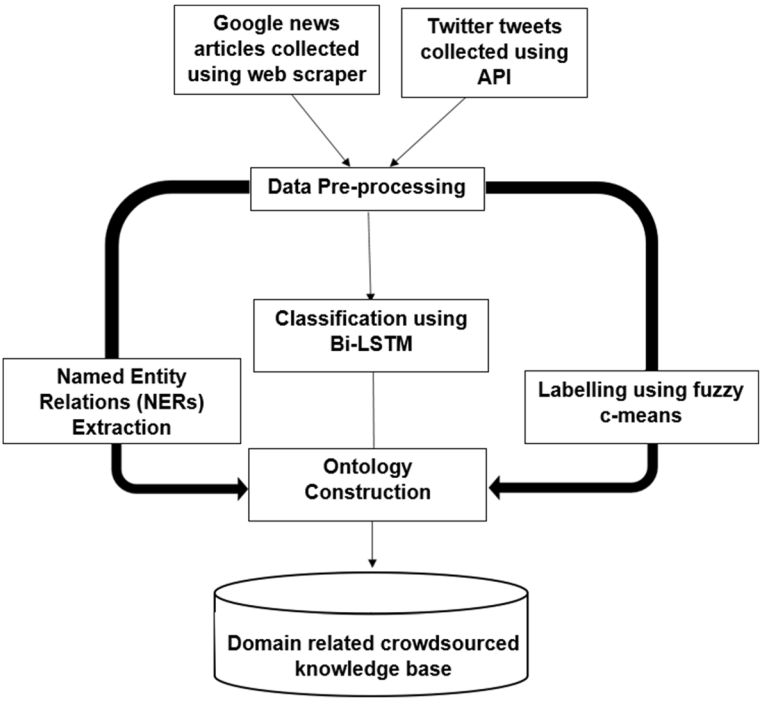

Fig. 1 Proposed System Architecture

the base URL was specified, in this case, the Google News URL. From there sub – URLs were crawled and with the help of the specified keywords, the required data was retrieved. The data was then arranged with the title and the respective context concatenated together and duplicate detection algorithms had been used to reduce redundancy. After the acquisition of data from both sources, some existing pre-processing algorithms were used to clean the textual context of the data.

3.2 Word Embeddings using GloVe

A pre-trained GloVe model [25] (Global Vectors) was used to convert the cleaned data into word embeddings (for input to the classification model) and store it in a database of vectors with each document being separate from each other. This was done for machine-readable input. Stanford University's pre-trained Wikipedia 2014 GloVe model was used for the embedding done here. Wikipedia is a good source of common and rare English words.

Word2Vec [19] [20] is known to be a very good method of word embeddings in deep learning techniques. But what Word2Vec does is consider the context only locally for predicting the neighboring words in a sentence. The problem with Word2Vec is that it does not consider the global context. GloVe, on the other hand, is a much more powerful method for generating word embeddings which give useful and deep insights into the embeddings or vectorization of words. The GloVe makes use of the global count statistics and also takes in the local context to generate word vectors. The vectors are then split into training, dev, and test sets.

3.3 Labelling using fuzzy c-means

Fuzzy c-means clustering [26], which is only an intermediate step, was used for labeling of the data. With the help of TF-IDF [16] vectors of the data, weighted representations of the documents were made. This was accomplished to ensure that different types of crime-related news articles are clustered together for labeling the data with types of crime. TF-IDF was used as it is an efficient method of representation of documents in context with documents in the corpus and takes in the importance of words in all the documents. Clustering of the documents could have been done with cosine similarities of the TF-IDF vectors, but most of the crime-related news articles usually report more than one type of crime. The Fuzzy c-means approach was used in order to justify the fact that one entity of one cluster can belong to other clusters as well. TF is Term frequency and IDF is Inverse document frequency. Equation (1), (2), and (3) depict the equations to compute the TF-IDF.

$$TF = \frac{Number\ of\ times\ a\ term\ occurs\ in\ document}{N} \tag{1}$$

$$IDF(t) = log\frac{N}{Number\ of\ Documents\ with\ term} \tag{2}$$

$$TF{-}IDF = TF{*}IDF \tag{3}$$

With this, clusters of the existing data were created and each node was manually labeled for the type of crime committed and also a node that doesn't consist of any crime was present. This is due to false positives and was because of the data being acquitted with the help of keywords. This node was denoted as no-crime and this was needed to train the classification model. The database of vectors was then updated with labels being given to each document based on the types of crime and that of no-crime. The clusters were stored for further reference for the ontology. The following model was created to avoid manual labeling of data in the future as this is a tedious and time-consuming task. After manually labeling the clusters, the vectors were then separated into training sets, dev sets, and test sets.

3.4 The bi-LSTM classification model

A Sequence to one Deep Bi-Directional LSTM Recurrent Neural Network was constructed for the multi-label classification task, this model was created in the hope of having a pre-trained model for further crime classification. It takes in as input the training set from the vector database for training and the validation is done with the use of the development set. LSTM was used as it is very good at remembering long sentences and also makes sure to forget the irrelevant context. Thus, it helps in the classification task by keeping in context words that are relevant to a particular word but might be far away from the relevant word. After tuning the hyperparameters, the model then gives the types of relevant crime from the corresponding docuemnts.

Algorithm 1: Labelling Algorithm

Input: Raw Data collected from online sources (g_data, t_data) (1), key words, number of clusters, time line (2)
// Google news and Twitter tweets
Output: labelled entities, GloVe vectors, clean data, clusters written on files

begin

Step 1: data = [ ]
for each_doc in t_data:
c_data = **expand joint hashtags and expand slangs in each_data**
data.append(c_data) **store in data list**
end for
intermediate_data = **concatenate both the data from Google news and Twitter**

Step 2: clean_data = [ ]
Named, Chunked, Identified and Labelled Entities = [ ]
for each_doc in intermediate_data:
c_data = **convert each_doc to lower characters, remove stop words and lemmatize**
Named, Chunked, Identified and Labelled Entities.append(NER(c_data))
store Named, Chunked, Identified and Labelled Entities of cleaned data
clean_data.append(lem_data) **store cleand data**
end for

Step 3: glove_model = LoadModel("GloVe") **load the Glove pre-trained model**
glove_vectors = glove_model.predict(clean_data) **convert the cleaned data to glove vectors**

Step 4: google_data = [ ]
twitter_data = [ ]
for each_date in timeline:
c_data = **crawl for Google news for each_date**
google_data.append(c_data) **with text, headlines and location**
c_data = **get Twitter tweets with API**
twitter_data.append(c_data)
google_data, twitter_data = **perform duplicate detection**
end for

Step 5: tf_idf = **get the TF-IDF vectors of clean_data**
clusters = **perform Fuzzy c-means for clustering of tf_idf**
label the clusters appropriately and write in a file
end

(1) Google news and Twitter tweets
(2) Duration from which news must be collected

With the labeled data, a classification model is built. Bi-directional LSTM was used, as by using bi-directional networks, the inputs are fed one by one from the direction, left to right and one input from the direction, right to left and what contrasts this methodology from unidirectional is that in the LSTM that runs in reverse you preserve data from the future and, utilizing the two concealed states joined you are capable in any point so, as to use data from both past and future. 'tanh' activation function was used to solve the vanishing gradient problem in LSTM, whose second derivative can be preserved for a

long-range of sequences before descending to 0. Sigmoid activation was also used for the gates.

The model was trained to classify the news articles and, tweets into one of the 14 categories of crime (including no-crime). 6 layers of bi-directional LSTM was used, pairs of 2 were created as one used the original input, the other used the same input, but reversed. A layer of sigmoid activation layer was used it the end. Since more than one class has to been chosen, the Softmax activation function cannot be used, as it converts scores into probabilities considering other scores. But, since it is a problem of multi-label classification, the scores in the last layer must be independent of each other. The model was built with the Keras open-source package. Short-term memory is a major disadvantage of RNNs. If an input is very long, RNNs will face difficulty in transferring data from a one-time step to another further away step. So, in processing pages of text, RNN's may not be efficient in carrying data from the early time steps.

Figure 2 depicts an LSTM unit that has a forget, update, and output gate. The information in a sequence of data that is necessary or important, are decided by these gates. Thus, to make predictions, only the relevant data is passed down the long chain of sequences. This will allow the unit to forget information that is not necessary and, update relevant information from even far off entities. In the LSTM units, the forget gate is the one that decides which data should be kept, and which should be thrown from the previous hidden states. The previous hidden outputs along with the incoming inputs are passed through a sigmoid function, which in turn, produces values in the range of zero to one. The nearer to 1 signifies to keep the data, and values that are nearer to zero signifies to throw it or forget it.

To change the cell state, the input gate was operated. The values comprising of the previous hidden states, and the incoming input is computed from a sigmoid function. The cell states will be updated by the values, ranging from zero to one, by the transformation from the sigmoid function. It is indicated that 1 denotes relevant and 0 represents not

Algorithm 2: Algorithm for training the model and constructing the knowledge graph

Input: GloVe vectors, clean data, labelled entities read from files
Output: trained classification model, ontology representation

begin
Step 1: training_set, dev_set, text_set = **Split the GloVe vectors into training, dev and test sets**
error = **Initiate some error value as threshold for training**
train_error = **Some large number initialized for training error**
model = ModelLoad("Bi-LSTM") **Constructed model as explained above**
mode.fit(training_set, dev_set)
while train_error > error:
train_error = **Get the training error after each epoch**
end while
Step 2: *ontology_base = NULL*
for each doc:
nodes, edges = **Create nodes with labelled and clean data**
graph = **Generate the graph with nodes, edges and graphs as described above**
ontology_base.update(graph)
Step 3: **return** model, ontology_base

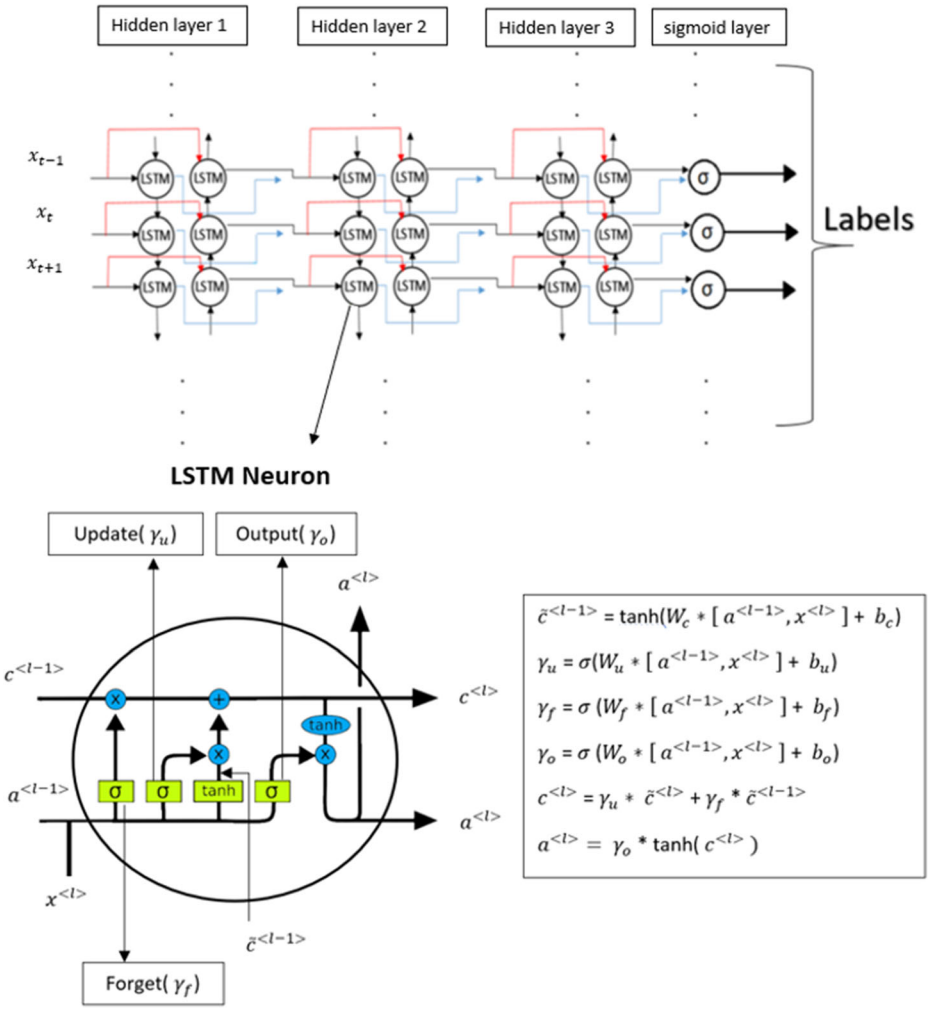

Fig. 2 Complete Architecture of the Proposed Bi-LSTM model

relevant. A tanh function is used to transform the values of the hidden state and current input values, into values ranging from −1 to 1, which makes sure the network is regulated. Furthermore, the sigmoid output and the tanh output is multiplied to make sure which data is relevant to preserve from the tanh output. The hidden state for the next stage is decided by the output gate.

The model could then classify a news article or tweet as the types of crime committed and this was used for frequency, the relevance of the crime (concerning the location) and was able to create a historical reference also, using the test set. With the help of the web scraper and Twitter API, future analysis of crime related news becomes easier as the trained model will update the ontology base with the new results. Many other classification algorithms exist like SVM, Random Forests, etc., but for sequential data, sequence Deep learning models are proven to be the best as they consider the sequences' meaning part by part.

3.5 Ontology construction

An ontology is an aggregation of knowledge with defined semantic relations between several concepts of a specific domain. The reason for indicating the relations or properties of a branch of knowledge and how they are connected, by characterizing a lot of ideas and classes that represent the subject. Ontologies are useful in formulating and constructing data searching strategies because of its ability to combine the associated domain knowledge with the semantic model of the data. It is very useful in defining connections between the various types of semantic knowledge [21]. The types of crime to choose is a cumbersome task as there are more than 1 million types of crime according to the Indian constitution. In this paper, the major and frequent crimes were generalized and chosen and those are 'Gambling', 'Robbery', 'Arson', 'Drunkenness', 'Harassment', 'Fraud', 'Burglary', 'Vandalism', 'Cybercrime', 'Assault', 'Molested', 'Trespass', 'Murder'. Also, a category called no-crime is used in case there were news documents unrelevant to crime. This was useful in constructing and training the classification model, for it to identify news not related to crime also.

Usually, when a crime is committed, it falls under many categories. E.g. the news phrase "The accused reportedly burnt down the house and murdered 3 people." falls under two categories, Arson and Murder. For this reason, a Multi label classification model was created. From the stored Named Entity Relations (NER), Clusters, and the predicted labels, an ontology database was created that classifies crime from the fourteen categories and gives the relation and frequency between the categories. This was done for an easier understanding of the relation between different types of crimes in India. An ontology knowledge base is created with the categories of crime as described above. The predicted values of the classification model were written to a CSV file. The Named Entities and the predicted values were considered and the crime related terms were arranged with importance given to frequency. For each location, the type of crime was mentioned, and then the document was connected. Later using Named Entities, and information gathered during crawling, the name of the accused, the time, and other relevant information was also included. A snippet of the knowledge aggregated graph is structured and is depicted in Fig. 3 for the criminology as a potential domain.

The Ontology is a Knowledge Modeling Scheme for provison of auxiliary knowledge for a specific domain. The Ontology modeled for the proposed scheme is based on a dynamic human centred model with automatic aggregation of terms. The cognitive gap between the crime events and human cognition is bridged by the ontologies where the events on the Social Web, both historical and trending from various handles like twitter and online News portals like Google News is formulated by defining the relationships between ontological elements and entities. The details like location, crime type, severiety, and the frequency of past occurences have been used as important entity markers for Ontology Modeling in the domains like Criminology and Crime. The information of the dynamically generated ontology for crime events have been depicted in Table 1. The Generic Rules for Dynamic Ontology Modeling for Criminology as a Domain from Google News and Twitter API is depicted in Table 2.

An example of the proposed ontology is shown in Fig. 3. The database comprises of many documents related to crime which in turn consists of many relations, some of which are depicted in Fig. 3. Each document consists of the location of where the relevant crimes had taken place in. The documents further consists of the type of the crime and the necessary details such as the date, details of the accused, the victims involved, etc. In the future, if a news article is to be considered, with the help of the classification model and ontology, it can be

found out what types of crimes have been committed, the frequency of that crime in that location and also if any patterns are forming, thus helping in many domains of crime analysis.

With the help of the Named Entities and the Clusters, a satisfactory ontology of criminology could be created. But with the help of the trained model, it was less time consuming and more efficient to classify articles for the type of crime, as the process is only once time-consuming. As the proposed system works on classifying crime-related news crime types, the different domains in named entity pairs were used in which the contextual words from intermediate positions of different entities highlighting locations, names, the crime, etc. were considered. Now, the Clusters were considered and all from these entity pairs, every intervening context words have been collected and used for the representation of crime. In the future for updating the ontology base, the Bi-LSTM model can be used.

4 Implementation

The entire system was implemented and carried out using a Python as a programming language and Pandas was the package used for reading and writing data. Tensor flow and Keras were the packages used for the construction and classification part. The ontological representation using graphs were modeled using Protégé. NLTK suite was used for the pre-processing steps, which include the algorithms for removal of stop words, hashtag expansion, slang expansion, and lemmatization. The Named Entity Relations was also obtained with the NLTK package. The experiments were carried out on the Google Colab platform with an environment of default Python 3.6 (4 vCPU 16 GB RAM).

The data is collected and prepared for generating the metadata, Ontologies and dynamically generate crime related knowledge from Google News and Twitter API based on crime related events and news present in the dataset. However, the experimentations have been considered on standard datasets as discussed in Section 4. This approach is based on the standards of the

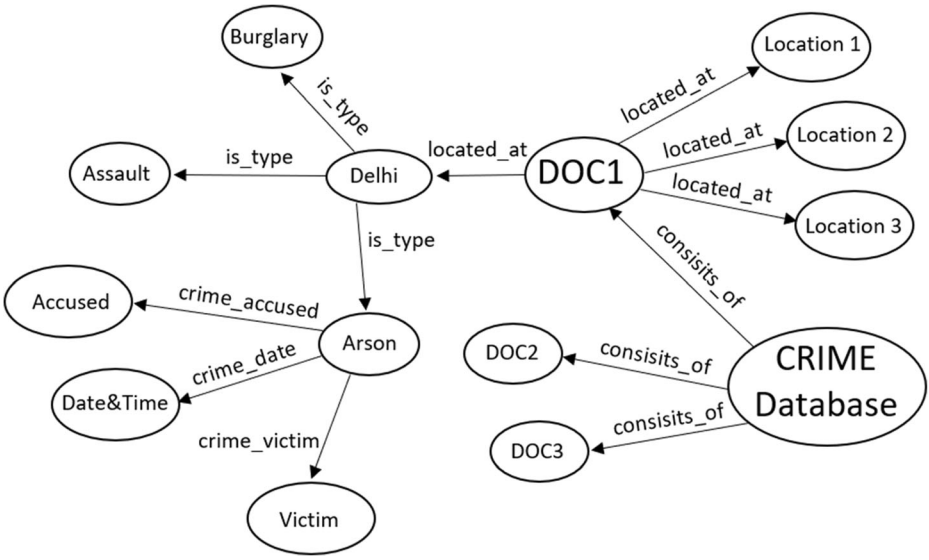

Fig. 3 A Snippet of the Ontological representation of Criminology

Table 1 Summary of the Dynamically Generated Ontology for Crime Events

No. of Sub-Domains	20
No. of Distinct Concepts	452
No. of Individuals	2457
No. of Strong Relationships	847
No. of Location Entities	674
No. of Distinct Axioms	12

Semantic Web, which is a Metadata driven. It is needed to generate, and model metadata based on the dataset as Semantic Web or the Web 3.0 is not yet fully formulated, and it is the Futuristic Vision of the World Wide Web and in a decade as per the vision of Sir Tim Berners Lee, the Web 2.0 will transform into a fully grown Metadata driven Semantic Web or the Web 3.0. This is the reason why metadata is currently generated by preparing the data. However, the experimentations are carried out on the standard datasets.

For evaluation and comparison with other methods, four datasets were considered: "UCI Community and Crime Dataset", "Fraud and Civil Action", "CAIL" [37], and "Crime in India". The UCI Community and Crime Dataset were put together from law agencies using socio-economic data from the surveys conducted by 1990 US LEMAS, 1990 US Census, and also data collected from the FBI, all using real-time data. The dataset comprises 2216 instances and 147 attributes. The Fraud and Civil Action Dataset consists of crime cases relevant to civil action, fraud, etc. It has 40,256 cases. The CAIL Dataset comprises cases of crime organized by the Supreme People's Court. For every instance, there are 2 parts, one the explanation of the crime and the other the judgment result for that crime. The Crime in India Dataset is a Kaggle public dataset that contains data about various types of crimes that have happened in India from 2001.

For data preparation, the major part is data pre-processing. The acquitted data from the online sources first underwent the data pre-processing techniques. This clean data was then stored in a csv file. The Named Entities generated are stored in a separate csv file. The word embeddings is done with the GloVe model and the embeddings or vectors were stored in a database of vectors. For the Clusters, at first, the clean data was taken and TF-IDF vectorization is applied so that every document can be represented as documents. Fuzzy c-means clustering algorithm was used for forming clusters and they were manually gone through for labeling. It was performed with skfuzzy package.

For data pre-processing, the following were the steps implemented. Hashtag Expansion: In Twitter, many Hashtags may contain useful information. For example, in the tweet "The city calls for national #HumanCivilRights", "#HumanCivilRights" had been split into "Human Civil Rights". Lower casing and punctuation: This was done to reduce redundancy of the same word. Stop words removal: Words like 'the', 'is', 'to', etc. were removed as they provide negligible semantic meaning to the text. Slang words Treatment: the common and well-known slang words frequently used on Twitter were expanded for simplicity. For example, slangs such as 'atb' was expanded to 'all the best'. The complete process pipeline for processing the data is depicted in Fig. 4.

Once the tokenization and lemmatization are achieved using the WordNet 3.0 Lemmatizer, Named Entity Relations is used to detect and classify named entities in textual data into categories that are created beforehand such as the names of organizations, locations, persons, etc. which has also been incorporated into the approach for interpreting the appropriate entities

Table 2 Generic rules for criminology domain dynamic ontology modeling

<owl: ObjectProperty rdfID: "SpecifyCrime#No">
<rdfs: CrimeName>
<rdf:owl is OntoEntity Rdf: CrimeName>
<owl:addAxiom rdf: parseType=CrimeSeriousness>
<owl:addAxiom rdf: parseType=CrimeNature>
<owl:addAxiom rdf: parseType=CrimeAssociationWithEntities>
<owl:CheckParticipatingInstances">
<owl:addAxiom rdf: parseTyoe = "DefineCardinality">
<owl:addrole rdf:roleType= "">
<owl.addrole rdf:roleType = "CheckOntologyRole">
<owl.addrole rdf:roleType = "DefineEventAsSubclass">
< owl.addrole rdf:roleType = "MentionParseType">
/* If an event or crime has more thab three unique instances or associations*/
<owl: checkOwlInstanceType="InstancetoConceptTransform">
<owl:isOwlOntoCommited>
<owl:checkCommitments as OntoRole>
<owl:ifOntoRoleIsUnique= "RetainOntologyRoles">
<owl:addrole rdf:roleType = "CheckOntologyRole">
<owl:addAxion edf:parseType = "MentionParseType">
<owl:checkCommitments as IndividualEntities>
<owl:ifOntoRoleIsUnique= "EntityLinktoDocument">
<owl:addrole rdf:roleType = "DocumentMatchIndex">
<owl:addAxion edf:parseType = "AddIndexToCrimeAsEvent">
owl:addAxiom rdf: parseType=CheckEventLocation>
<owl:addAxiom rdf: parseType=LinkwithPreviousEventsinGeoLocation>
<owl:addAxiom rdf: parseType=TagGeoLocationUsingLocationServices>
<owl:addAxiom rdf: parseType=IncludeUpdatesFromTwitterAPI>
<owl:addAxiom rdf: parseType=IncludeGoogleNewsContents>
<owl:addAxiom rdf: parseType=MapRelevantEntities>
</rdfs:>
</owl: ObjectProperty>

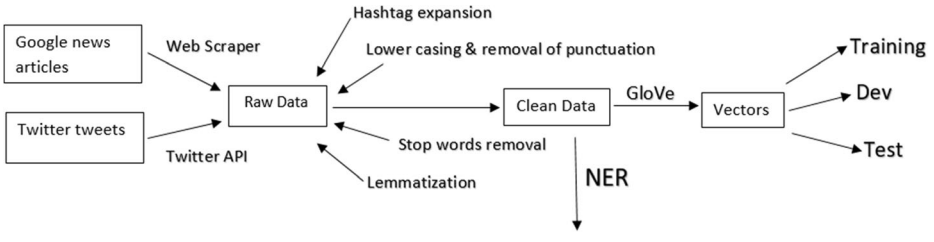

Fig. 4 Preparation of data

that are domain-centric as well. The ontology base is based on the Named Entity Relations, so these Named Entities are required in further steps of the system. After performing several stages of pre-processing, the cleaned data was stored as separate documents. All the data pre-processing techniques had been carried out with the NLTK package. The data pre-processing and labeling is depicted in Algorithm 1.

Algorithm 1 inputs the raw web-scraped data collected from online sources (Google News and Twitter API) using the web crawling and the scraping scripts. The algorithm first processes the Twitter data as tweets have hashtags and slangs that need to be expanded or transformed. After this is done for every file in the twitter data, then both the Twitter and Google News data are concatenated as one for the other data pre-processing steps such as removal of stop words, lemmatization, etc. Also, the Named Entity Relations are obtained using the NLTK package. The cleaned data is then used to form the GloVe word embeddings using a pre-trained embedding model. The algorithm returns the Named Entity Relations, GloVe vectors, and the clean data.

In Algorithm 1, with the keywords; the crime relevant documents that had been manually collected, the number of types of crime that are being considered added with one for the no-crime category and the duration in which the data must be collected, for example, from the year 2018 to 2019 were the inputs. The algorithm crawls through Google News India and searches for news articles checking with the keywords and also gets tweets from Twitter API using the keywords. For each date, this is done, and also duplicate detection algorithms run at every iteration making sure the news articles are unique. Then the data pre-processing algorithm is run on this data and Named Entity Relations, GloVe vectors, and the cleaned data is retrieved. After this TF-IDF vectors were created which was used for clustering through Fuzzy c-means. The algorithm outputs the Named Entity Relations, GloVe vectors, clean data, and the clusters written on files.

Algorithm 2 has been encompassed to train the model and create the ontological representation, the inputs are the GloVe vectors, the cleaned data, and the labeled entities which are all read from files, written on from Algorithm 1. The GloVe vectors are first separated into the training set, dev set, and test set for training and evaluation. An error measure is initialized for which the model will be trained. The constructed model is loaded and trained. The algorithm

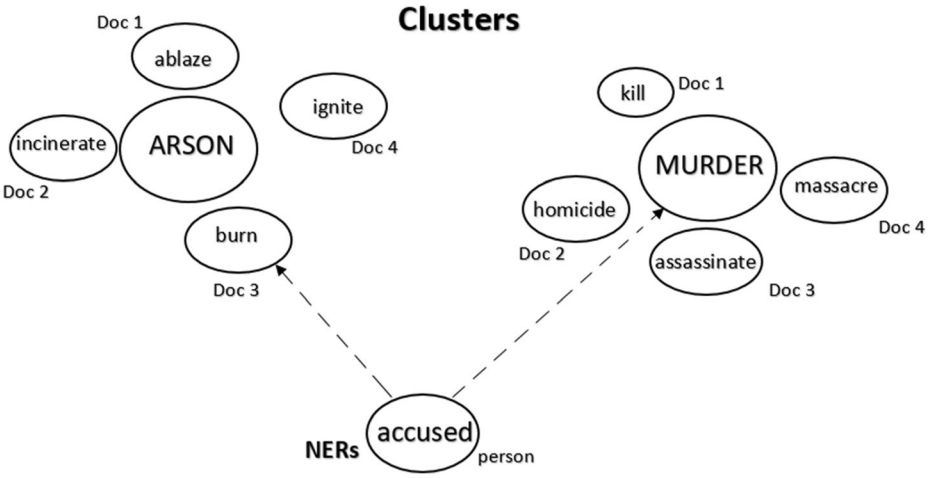

Fig. 5 Relationship between Clusters of documents containing certain words and Named Entities

then uses the labeled entities and the clean data to generate nodes and edges, and is shown in Fig. 5. The ontology knowledge base is then updated iteratively using the Named Entity Relations and the classifications for appropriate modifications. The algorithm outputs the trained model and the dynamically constructed knowledge graph.

5 Results and performance evaluation

The evaluation and performance of the Bi-LSTM classification model were done with Precision, Recall, F-Measure, Accuracy, and FNR (False Negative Rate) as potential metrics. The precision determines how precise a given classification model predicts positive labels. Precision is a good evaluation metric to use when the rate of a FP is very high and the rate of a FN is low. Recall is the fraction of the total amount of relevant instances that were retrieved. It highlights the sensitivity of the algorithm out of all the actual positives for how many were actually caught by the program. F-Measure is the weighted average of precision and recall, whereas accuracy is the average of precision and recall. FNR was used here as a measure of the error in the model. Table 3 and Eqs. (4), (5), (6), (7), and (8) show how the metrics mentioned above were calculated.

$$\textbf{Precision}\ (\mathrm{P}) = \frac{TP}{TP + FP} \tag{4}$$

$$\textbf{Recall}\ (\mathrm{R}) = \frac{TP}{TP + TN} \tag{5}$$

$$\textbf{F–Measure} = \frac{\mathbf{2}.P.R}{P + R} \tag{6}$$

$$\textbf{Accuracy} = \frac{P + R}{\mathbf{2}} \tag{7}$$

$$\textbf{FNR} = 1\text{–R} \tag{8}$$

To compare the Proposed Work with potential existing approaches, the HREACC [7], HANNCC [32], HMNCC [33], and GSCA [14] were considered as Baseline Approaches for comparison with the proposed model which is depicted in Table 4. The four datasets were

Table 3 Type 1 and Type 2 errors

Predicted Actual	Positives	Negatives
Positives	True Positives (TP)	False Positives (FP)
Negatives	False Negatives (FN)	True Negatives (TN)

Table 4 Performance metrics for all datasets and comparison between all method

Dataset	Method	Precision %	Recall %	F – Measure %	Accuracy %	FNR
UCI Community and Crime	HREACC	96.18	90.12	93.05	93.15	0.10
	HANNCC	92.83	87.14	89.89	89.98	0.13
	HMNCC	84.32	77.81	80.93	81.06	0.23
	GSCA	88.12	81.17	84.50	84.64	0.19
	Proposed work	**98.93**	**94.17**	**96.49**	**96.55**	**0.06**
Fraud and Civil Action	HREACC	81.16	74.88	77.89	78.02	0.26
	HANNCC	76.04	72.87	79.28	74.45	0.28
	HMNCC	69.32	66.11	67.67	67.71	0.34
	GSCA	73.16	70.47	71.78	71.81	0.30
	Proposed work	**85.12**	**80.87**	**82.94**	**82.99**	**0.20**
CAIL	HREACC	86.41	79.17	82.63	82.76	0.14
	HANNCC	81.37	76.81	79.02	79.09	0.19
	HMNCC	73.14	71.18	72.14	72.16	0.27
	GSCA	79.41	73.83	76.51	76.62	0.21
	Proposed work	**90.01**	**85.14**	**87.50**	**87.57**	**0.10**
Crime in India	HREACC	95.14	89.84	92.41	92.49	0.11
	HANNCC	91.03	86.27	80.58	88.65	0.14
	HMNCC	83.41	79.18	81.23	81.29	0.21
	GSCA	86.63	85.48	87.02	86.05	0.15
	Proposed work	**97.14**	**92.87**	**94.95**	**95.00**	**0.08**

considered for evaluation. For each dataset, all the comparison works were considered and the respective graphs were also plotted. The average values of the metrics were calculated and also tabulated for each dataset along with their corresponding comparison works.

From Fig. 6, one can infer the metrics for the UCI Crime and Community Dataset. The proposed method performed better than HREACC with an increase of 2.75% in precision, 4.05% increase in recall, 3.44% increase in F-Measure, and an increase of 3.4% in accuracy. FNR was lower for the proposed work by a factor of 0.04. Considering HANNCC for UCI Community and Crime dataset, the proposed work had an increase in precision, recall, F-Measure, and accuracy with increases of 6.1, 7.03, 6.6, and 6.57 in terms of percentage respectively. In terms of FNR, there was a decrease of 0.07 for the proposed work. The comparison of HMNCC and the proposed work yielded that the latter had higher metrics, to be precise, 14.61% in precision, 16.36% in the recall, 15.56% in F-Measure, 15.49% in accuracy. There was a drastic decrease in FNR for the proposed work when compared to HMNCC by a factor of 0.19. The proposed method performed better than the GSCA method, with an increase of 10.81% in precision, 13.00% in recall, 11.99% in F-Measure, 11.91% in accuracy. There was a decrease of 0.13 in FNR for the proposed work when compared to the GSCA method. The UCI Community and Crime dataset consist of 128 different attributes and by using GloVe vectors, it could be used for the Bi-LSTM model. The HREACC method which uses extreme learning machine was not able to use the semantic relations in the data. HANNCC, due to its incomplete and unexplained behavior, could not give high metrics. The HMNCC method which uses a hierarchal matching network for matching the labels of articles, gave lower metrics when baselined with the proposed work. The GSCA method which uses a Bayesian classifier also gave lower metrics due to the volume of the data.

From Fig. 7, the performance measures for the Fraud and Civil Action Dataset can be interpreted. When the HREACC method is baselined with the proposed work, one can infer that there was an increase in Precision, Recall, F-Measure, and Accuracy by factors of 3.96,

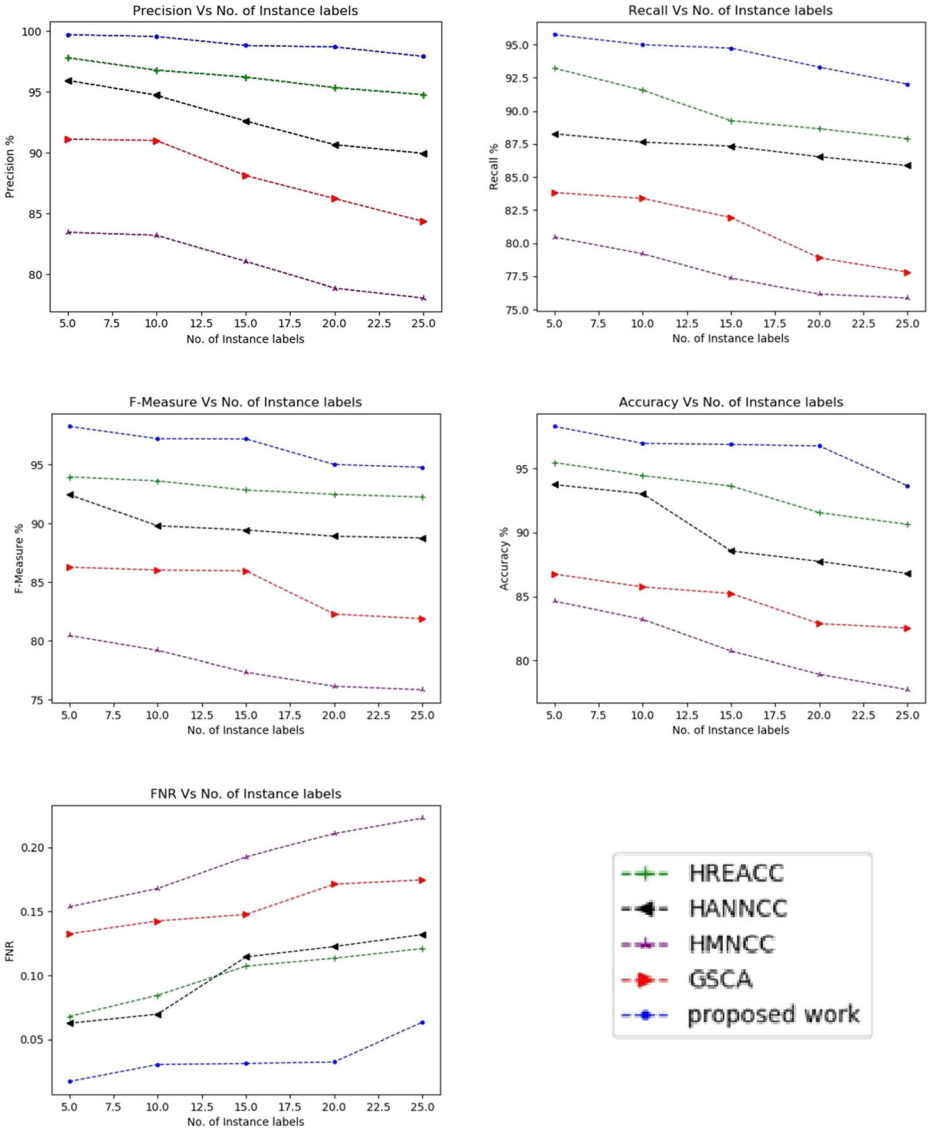

Fig. 6 Graphical representations of the metrics of methods for Dataset, UCI Community and Crime

5.99, 5.05, and 4.97 in terms of percentage respectively. Considering FNR, a decrease of 0.06 was observed. The proposed work performed better than HANNCC with an increase of 9.08% in Precision, 8% in the Recall, 3.66% in F-Measure, 8.54% in Accuracy. A decrease of 0.08 FNR was observed for the proposed work.

There were higher metrics when using Bi-LSTM when compared to the HMNCC method; with increases in Precision, Recall, F-Measure, and Accuracy by factors of 15.8, 14.76, 15.27, and 15.28 in terms of percentage respectively. The FNR is lower by a factor of 0.14. When the proposed work was compared with the GSCA method, observations lead to the conclusion that the proposed work outperformed the latter drastically in terms of the metrics. Precision, Recall,

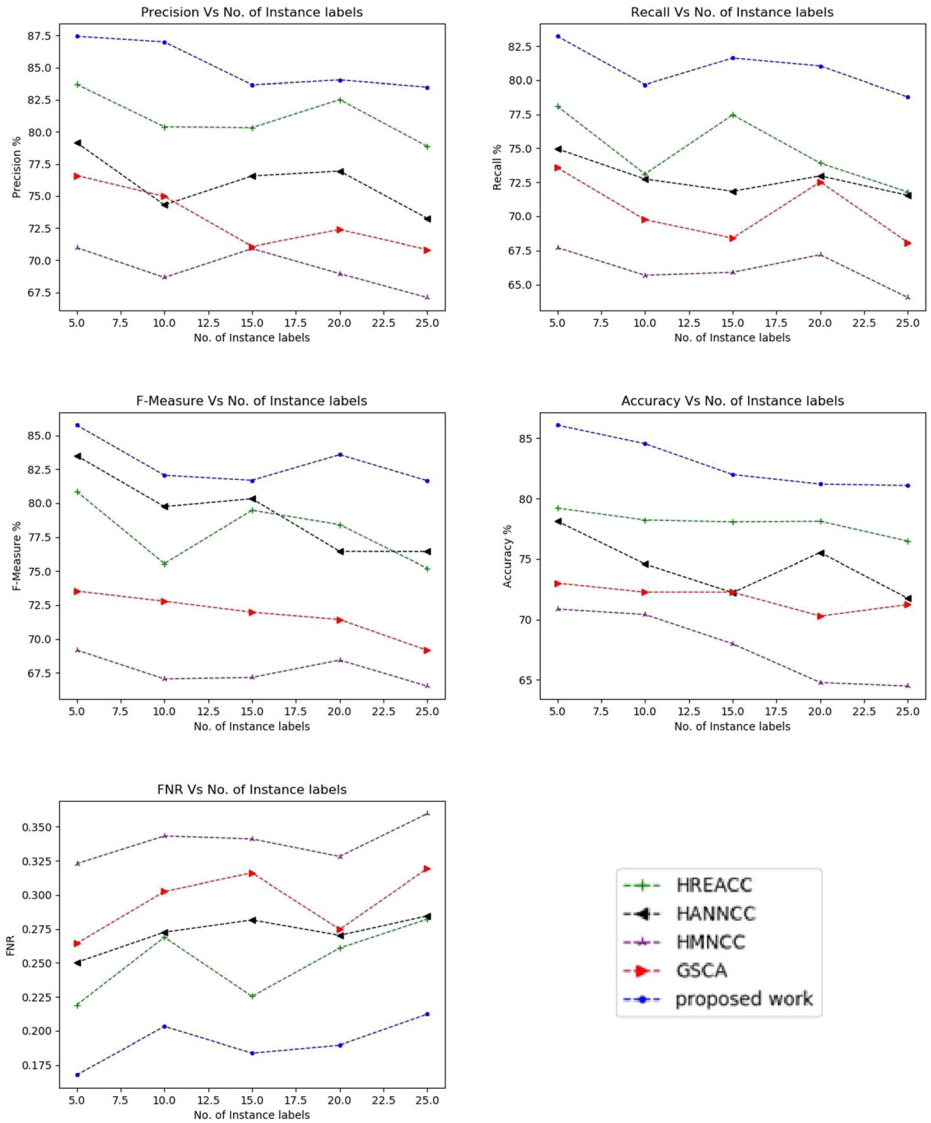

Fig. 7 Graphical representations of the metrics of methods for Dataset Fraud and Civil Action

F-Measure, Accuracy had an increase of 11.96%, 10.4%, 11.16%, 11.18% for the proposed work respectively. There was a decrease of 0.1 in FNR.

The HREACC method initially clusters the data into three hotspots. For the Fraud and Civil Action dataset, the proposed method works better than the HREACC method. The HANNCC method uses an Artificial Neural network which is not very good in keeping prior information and thus when baselined with the Bi-LSTM method, had lower metrics. The proposed method faces the semantic relations between words better than the HMNCC method and thus has better metrics. Similarly, the Bi-LSTM method when baselined with the GSCA method works better in identifying semantic relations.

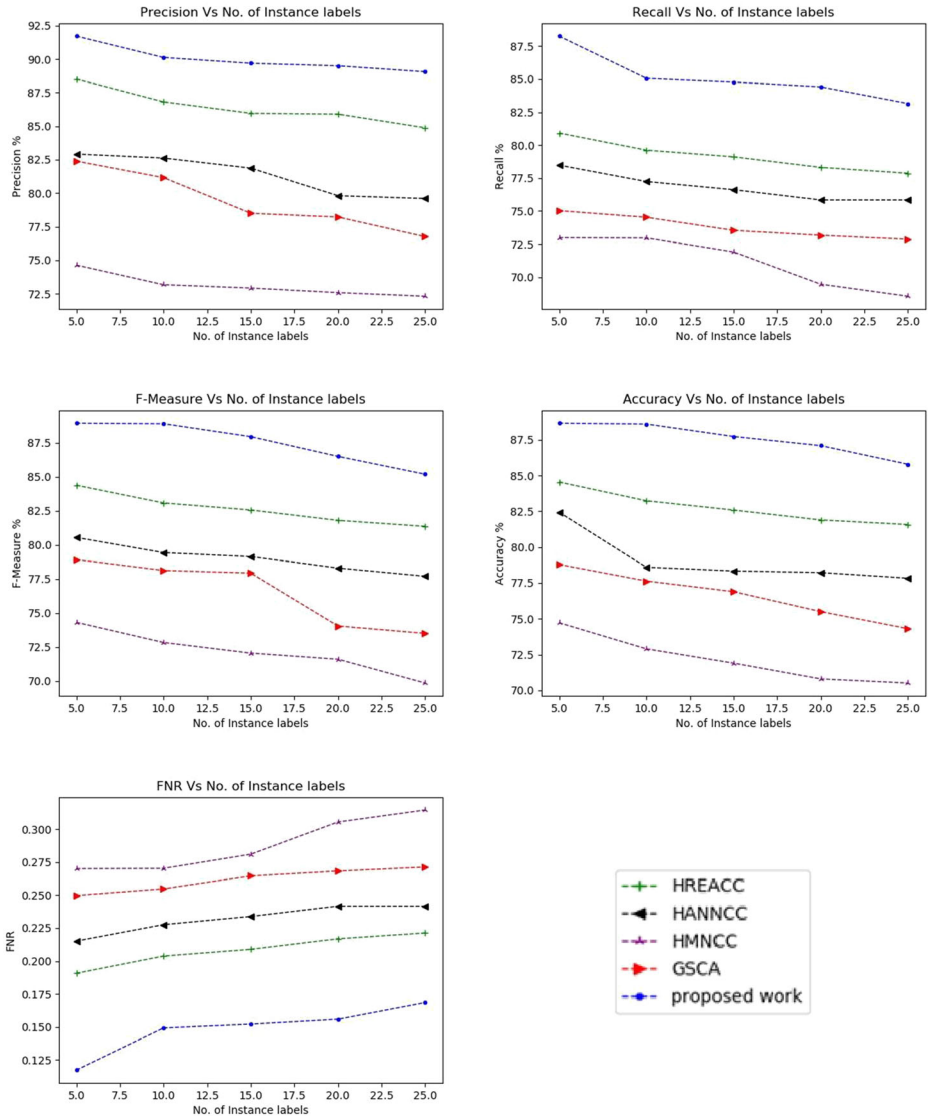

Fig. 8 Graphical representations of the metrics of methods for Dataset CAIL

From Fig. 8, the performance comparison for the CAIL dataset can be inferred. It was evident that the proposed work outperformed the HREACC method, as there was an increase of 3.6% in Precision, 5.97% in the Recall, 4.87% in F-Measure, and 4.81% in Accuracy. FNR for the proposed work was comparatively 0.1 lower than HREACC. HANNCC, when compared with the proposed work, had lower metrics, particularly the Bi-LSTM work had increases in Precision, Recall, F-Measure, and Accuracy of 8.64%, 8.33%, 8.48%, and 8.48% respectively. A decrease of 0.9 FNR was observed for the proposed work when baselined with the HANNCC method.

The proposed work outperformed the HMNCC method by an increase in Precision, Recall, F-Measure, and Accuracy by a factor of 16.87%, 13.96%, 15.36%, and 15.41% respectively.

In terms of FNR, there was a drastic decrease of 0.17. When baselined with the GSCA method, the Bi-LSTM method works better as there were better metrics in Precision, Recall, F-Measure, and Accuracy, and FNR. An increase of 10.6% in Precision, 11.31% in the Recall, 10.99% in F-Measure, 10.95% in Accuracy, and a decrease of 0.11 in FNR was observed.

For the CAIL dataset, the HREACC method initially uses hotspot analysis, leading to relatively more false positives and thus, the Bi-LSTM method works better. The proposed method, which uses the Adam optimizer, when baselined with the HANNCC method, which uses the Artifical Bee Colony optimization algorithm, performs better as Adam is more efficient than the Artifical Bee Colony algorithm. The HMNCC method matches semantic relations based on the parent fact, which leads to more false positives when compared to the Bi-LSTM method. The proposed method, when baselined with the GSCA method, performs better as the CAIL dataset consists of fact description and relevant judgments and thus Bayesian classifier does not perform well in deriving the semantic relations in the dataset.

From Fig. 9, it is observed that the Bi-LSTM method works better when baselined with the HREACC method, mainly since there was an increase of 2% in Precision, 3.03% in the Recall, 2.54% in F-Measure and 2.51% in Accuracy with a decrease in FNR by a factor of 0.03. For the HANNCC method, the metrics were comparatively lower than the proposed work; 6.11% in Precision, 6.6% in the Recall, 14.37% in F-Measure, 6.35% in Accuracy, and 0.06 in FNR. HMNCC method when compared with the Bi-LSTM method yielded that the proposed work performs better. There was an increase of 13.73% in Precision, 13.69% in Recall, 13.72% in F-Measure, 13.71% in Accuracy. In terms of FNR, a lower factor of 0.14 was observed. The proposed work outperformed the GSCA method in terms of Precision, Recall, F-Measure, Accuracy by an increase of 10.51, 7.39, 7.93, 8.95 respectively in terms of percentage. There was a decrease of 0.07 in FNR. The HREACC with the Crime in India dataset has lower metrics than the proposed method as the data is not sequential and thus the extreme learning algorithm does not perform well. Similarly, the Bi-LSTM method when compared with the HANNCC method performs better because ANN cannot keep prior information. The HMNCC method uses semantic matching with facts but is not as efficient as the Bi-LSTM method. The GSCA method also doesn't perform well when baselined with the proposed method due to the vast categorical data.

The reason for the proposed approach to perform much better than the baseline approaches is due to several factors which are subsequently discussed. HREACC made use of the Extreme Learning Machine learning algorithm for the classification of crime. They have used RNN for the feature extraction and learning process and while this might work well for short sentences or sequences when it comes to news data, RNNs will not be able to extract features efficiently as it cannot keep the information of context entities from far off attributes in a sequence. The proposed work overcomes this problem by using Bi-LSTM as it can remember attributes from far off point in a sequence. The RNN solves the memory problem by introducing a feedback look that plays the role of memory. So the previous input value to the RNN model leaves a trace. Bi-LSTM extends that idea and by creating both a short-term and a long-term memory component. The model inputs the sequential data as it is and also reversed, as the meaning of a word might depend on the words that precede it as well as the words that succeed it. And thus for long sequential data, Bi-LSTM can perform better than simple RNNs.

The HMNCC incorporates the Hierarchical Matching Network and doesn't perform well comparatively. This might because of the robustness of the news articles as it is rapidly changing and that HMN makes use of law relations for classification. Since the Bi-LSTM

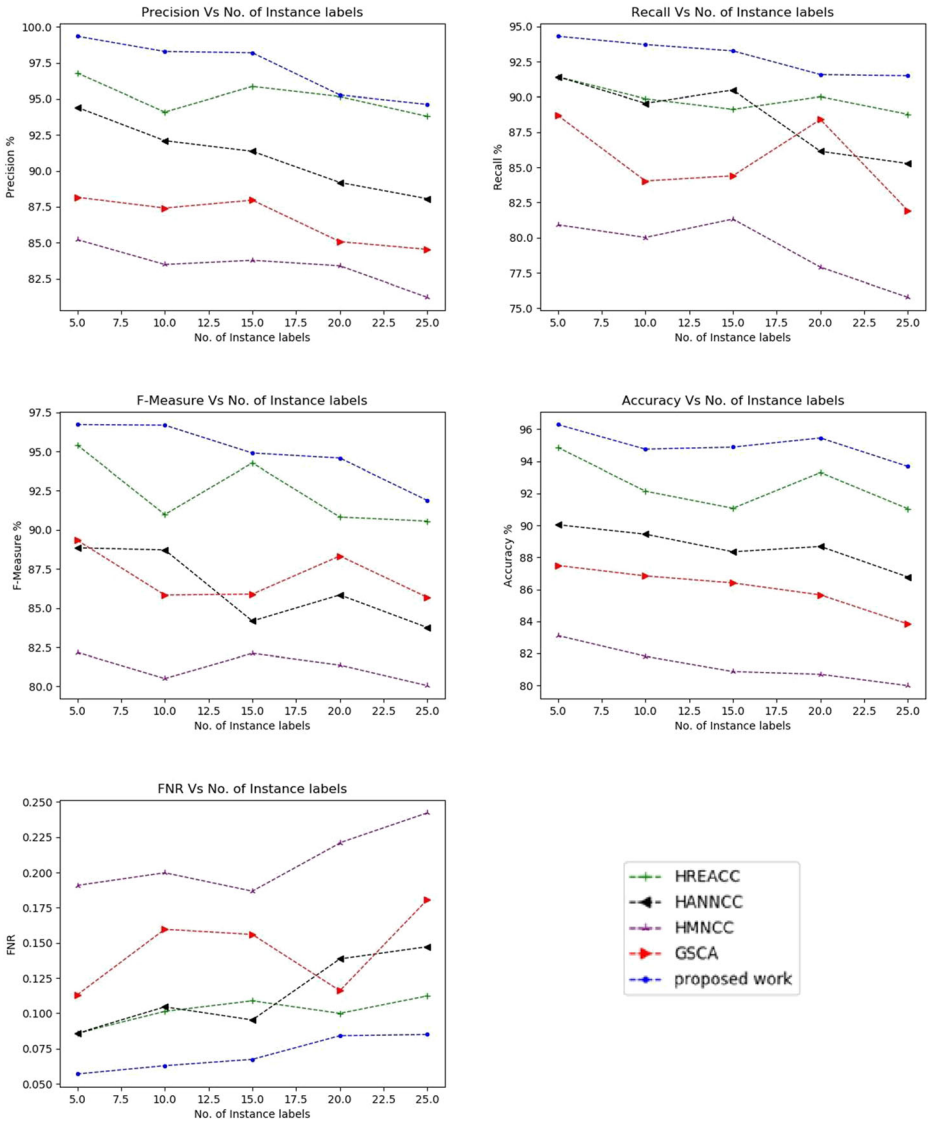

Fig. 9 Graphical representations of the metrics of methods for Dataset Crime in India

model, with the help of the GloVe vectors, can consider the semantic meaning between different words efficiently, this problem is overcome. HANCC makes use of a hybrid ANN with the artificial bee colony algorithm for optimizing the ANNs local optima problem and have used this method for crime classification. In ANNs, the number of parameters required for an efficient model is quite high and this is a challenge both in terms of computation and tuning. ANNs cannot efficiently generalize for sequences with different lengths or correlate the semantic relationship between words and sentences. The proposed work can overcome these obstacles as the number of parameters needed are comparatively lesser and it can take different length sequences.

The GSCA method classifies the types of crime using the Bayesian classifier after feature extraction using statistical algorithms. A Bayesian classifier has a considerable chance of loss of accuracy because of its assumption that features are independent of one another when conditioned upon class labels which are not so accurate. The Bi-LSTM method takes a discriminative approach to classification by trying to differentiate between positive and negative examples.

The proposed approach is efficient in differentiating between news relevant to crime and the ones which are not and also in classifying the types of crime from news or text data. The usage of a Bi-LSTM neural network is a useful method for the classification of new articles of crime as it will take considerably lesser time with good precision than other methods because the model is an efficient choice and had been trained on a large dataset with great variety. Furthermore, the construction of the ontological explanation can further help researchers have greater insights and analyze crime patterns, advancing the field of crime prediction and studies.

6 Conclusions

A knowledge driven hybridized approach to classify crime data for facilitating crime prediction analytics has been proposed. The approach is centred on an Ontological Model which is dynamically modelled based data from Google News and the crime related events from Twitter which is a major portion of the Social Web. The encompassment of Fuzzy c-means clustering with the TF-IDF facilitated initial labelling. A Deep Bi-LSTM neural network was trained on the four distinct datasets along with the dynamically modeled Knowledge Model, for classification of crime related data accurately, and effectively. The amalgamation of auxiliary knowledge and semantic infused learning by facilitating dynamic modelling of Ontologies from the metadata has not only enriched the performance of classification of crimes but also is the first of its kind approach to facilitate crime classification using a semantic infused learning paradigm. The knowledge centric approach for classification of crime based on the events on Google News and the Social Web, served as a dorminant vantage point for crime related analyses and has also promoted toward eXplainable Artificial Intelligence in crime classification. The proposed method outperforms the other methods with an average increase of 9.2% in accuracy and with a very low FNR value of 0.11, for all the four distinct standard datasets.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Abbass Z, Ali Z, Ali M, Akbar B, Saleem A (2020) "a framework to predict social crime through twitter tweets by using machine learning," *2020 IEEE 14th International Conference on Semantic Computing (ICSC)*, San Diego, CA, USA, 2020, pp. 363–368, https://doi.org/10.1109/ICSC.2020.00073.
2. Alatrista-Salas H, Morzán-Samamé J, Nunez-del-Prado M (2020). “Crime Alert! Crime Typification in News Based on Text Mining”. In: Arai K., Bhatia R. (eds) Advances in Information and Communication. FICC 2019. Lecture notes in networks and systems, vol 69. Springer, Cham.
3. Anuar S, Selamat A, Sallehuddin R (2015) “Hybrid artificial neural network with artificial bee Colony algorithm for crime classification”. In: Phon-Amnuaisuk S., au T. (eds) computational intelligence in information systems. Advances in intelligent systems and computing, vol 331. Springer, Cham.

4. Ashagrie M, Tekli J, Taddesse FG, Chbeir R, Tekli G (2019) Generic metadata representation framework for social-based event detection, description, and linkage. Knowledge-Based Systems 188. https://doi.org/10.1016/j.knosys.2019.06.025
5. Bhalla A, Pawar RP (2019) Crime in India 2018, National Crime Records Bureau (Ministry of Home Affairs) Government of India
6. Bhati S, Vikramaditya and Tiwari S, Mandloi J, (2019). "Machine Learning and Deep Learning Integrated Model to Predict, Classify and Analyze Crime in Indore City". Proceedings of Recent Advances in Interdisciplinary Trends in Engineering & Applications (RAITEA) 2019. Available at SSRN: https://ssrn.com/abstract=3364984 or https://doi.org/10.2139/ssrn.3364984.
7. Boppuru PR, Ramesha K (2019) Geo-spatial crime analysis using newsfeed data in Indian context. International Journal of Web-Based Learning and Teaching Technologies (IJWLTT) 14(4):49–64. https://doi.org/10.4018/IJWLTT.2019100103
8. Chen H, Chung W, Xu J, Wang G, Qin Y, Chau M (2004) Crime data mining: a general framework and some examples. IEEE Explore-Computer 37(4):50–56
9. Das P, Das AK (2020). "Graph-based crime reports clustering using relations extracted from named entities". In: Behera H., Nayak J., Naik B., Pelusi D. (eds) computational intelligence in data mining. Advances in intelligent systems and computing, vol 990. Springer, Singapore
10. Das P, Das A, Nayak J, Pelusi D, Ding W (2019) Group incremental adaptive clustering based on neural network and rough set theory for crime report categorization. Neurocomputing. https://doi.org/10.1016/j.neucom.2019.10.109
11. Fares M, Moufarrej A, Jreij E, Tekli J, Grosky W (2019) Unsupervised word-level affect analysis and propagation in a lexical knowledge graph. Knowl Based Syst 165:432–459
12. Gerber M (2014) Predicting crime using twitter and kernel density estimation. Decis Support Syst 61. https://doi.org/10.1016/j.dss.2014.02.003
13. Ghankutkar S, Sarkar N, Gajbhiye P, Yadav S, Kalbande D, Bakereywala N (2019) "modelling machine learning for Analysing crime news", *2019 International Conference on Advances in Computing, Communication and Control (ICAC3)*, Mumbai, India, pp. 1–5, https://doi.org/10.1109/ICAC347590.2019.9036769.
14. Hardy J, Bell P, Allan D (2020) A crime script analysis of the Madoff investment scheme. Crime Prev Community Saf 22:68–97
15. Hochreiter S, Schmidhuber J (1997) Long Short-term Memory. Neural computation 9:1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735
16. Jurafsky D, Martin J. (2008). Speech and language processing: an introduction to natural language processing, Computational Linguistics, and Speech Recognition
17. Kumar A, Verma A, Shinde G, Sukhdeve Y, Lal N (2020). "crime prediction using K-nearest neighboring algorithm," *2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE)*, Vellore, India, 2020, pp. 1–4, https://doi.org/10.1109/ic-ETITE47903.2020.155
18. Lal, Sangeeta & Tiwari, Lipika & Ranjan, Ravi & Verma, Ayushi & Sardana, Neetu & Mourya, Rahul. (2020). "Analysis and Classification of Crime Tweets". Procedia Computer Science. 167. 1911–1919. https://doi.org/10.1016/j.procs.2020.03.211.
19. Mikolov T, Chen K, Corrado G, Dean J (2013) "Efficient estimation of word representations in vector space", CoRR (2013) 1–12 abs/ 1301.3781.
20. Mikolov T, Sutskever I, Chen K, Corrado G, Dean J (2013) "Distributed representations of words and phrases and their compositionality", in: Proceedings of the 26th International Conference on Neural Information Processing Systems, Se- ries = NIPS'13, Vol. 2, 2013, pp. 3111–3119. abs/ 1310.4546.
21. Munir K, Anjum MS (2018) The use of ontologies for effective knowledge modelling and information retrieval. Applied Computing and Informatics 14:116–126
22. Nair S, Soniminde S, Sureshbabu S, Tamhankar A, Kulkarni S, (2019). "Assist Crime Prevention Using Machine Learning". Proceedings 2019: Conference on Technologies for Future Cities (CTFC).
23. Noormanshah WMU, Nohuddin PNE, Zainol Z (2020) "Document content analysis based on random Forest algorithm". In: Peng SL., son L., Suseendran G., Balaganesh D. (eds) intelligent computing and innovation on data science. Lecture notes in networks and systems, vol 118. Springer, Singapore
24. Pangestuti D, Herdiani A, Selviandro N (2019) "Analysis and implementation of ontology based text classification on criminality digital news". IOP conference series: materials science and engineering. 662. 022135. https://doi.org/10.1088/1757-899X/662/2/022135.
25. Pennington J, Socher R, Manning C (2014). "Glove: global vectors for word representation", in: proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), 2014, pp. 1532–1543
26. Priandini N, Zaman B, Purwanti E (2017). Categorizing document by fuzzy C-Means and K-nearest neighbors approach. AIP Conference Proceedings. 1867. 020012. https://doi.org/10.1063/1.4994415.

27. Ramasubbareddy S, Aditya Sai Srinivas T, Govinda K, Manivannan SS (2020). Crime prediction system. In: Saini H., Sayal R., Buyya R., Aliseri G. (eds) innovations in computer science and engineering. Lecture notes in networks and systems, vol 103. Springer, Singapore
28. Saha R, Naskar A, Dasgupta T, and Dey L (2020) "A System for Analysis, Visualization and Retrieval of Crime Documents". In Proceedings of the 7th ACM IKDD CoDS and 25th COMAD (CoDS COMAD 2020). Association for Computing Machinery, New York, NY, USA, 317–321.
29. Soleimanian Gharehchopogh F, Haggi S (2020) An optimization K-modes clustering algorithm with elephant herding optimization algorithm for crime clustering. Journal of Advances in Computer Engineering and Technology 6(2):78–87
30. Sreejith AG, Lansy A, Krishna KSA, Haran VJ, Rakhee M (2020). Crime analysis and prediction using graph mining. In: Ranganathan G., Chen J., Rocha Á. (eds) inventive communication and computational technologies. Lecture notes in networks and systems, vol 89. Springer, Singapore
31. Sundhara Kumar KB, Bhalaji N. (2020) A Novel Hybrid RNN-ELM Architecture for Crime Classification. In: Smys S., Senjyu T., Lafata P. (eds) Second International Conference on Computer Networks and Communication Technologies. ICCNCT 2019. Lecture notes on data engineering and communications technologies, vol 44. Springer, Cham
32. Thilagam P, Karur S (2019) Crime base: Towards building a knowledge base for crime entities and their relationships from online news papers. Information Processing & Management:56. https://doi.org/10.1016/j.ipm.2019.102059
33. Wang P, Yu F, Niu S, Yang Z, Zhang Y, Guo J. 2019. Hierarchical matching network for crime classification. In proceedings of the 42nd international ACM SIGIR conference on Research and Development in information retrieval (SIGIR'19). Association for Computing Machinery, New York, NY, USA, 325–334.
34. Wang M, Cai Q, Wang L, Li J, Wang X. (2020) "Chinese news text classification based on attention-based CNN-BiLSTM", proc. SPIE 11430, MIPPR 2019: pattern recognition and computer vision
35. Zaidi NAS, Mustapha A, Mostafa SA, Razali MN (2020) "A classification approach for crime prediction". Communications in Computer and Information Science, 68–78.
36. Zhang Z, Huang J, Hao J et al (2020) Extracting relations of crime rates through fuzzy association rules mining. Appl Intell 50:448–467
37. Haoxi Zhong, Guo Zhipeng, Cunchao Tu, Chaojun Xiao, Zhiyuan Liu, and Maosong Sun. 2018. Legal Judgment Prediction via Topological Learning. In Proceedings of the 2018 Conference on empirical methods in natural language processing. Association for Computational Linguistics, 3540–3549.