Design and Analysis of Machine Learning Algorithms for Predicting Pattern and Trends of Crime in India

Thesis submitted by

Surajit Giri

Doctor of Philosophy

Department of Computer Science
Ramakrishna Mission Residential College (Autonomous)
VIVEKANANDA CENTRE FOR RESEARCH
P.O. Narendrapur, Dist. 24 Parganas (South), West Bengal, Pin 700103, India

2024

VIVEKANANDA CENTRE FOR RESEARCH

A University of Calcutta recognised Research Centre

P.O. Narendrapur, Dist. 24 Parganas (South), West Bengal, Pin 700103, India

1. Title of the thesis : **Design and Analysis of Machine Learning Algorithms for Predicting Pattern and Trends of Crime in India.**

2. Name, Designation & Institution of the Supervisors : **DR. Siddhartha Banerjee, Associate Professor, Department of Computer Science, Ramakrishna Mission Residential College (Autonomous).**

3. List of Publication :

   (a) Surajit Giri and Siddhartha Banerjee, *Ensemble Learning Approach for Phishing Website Detection using Greedy Stacking Model,*Journal of The Institution of Engineers (India): Series B, Communicated after 2^{nd} revision, March, 2024.

   (b) Surajit Giri and Siddhartha Banerjee, *Performance analysis of annotation detection techniques for cyber-bullying messages using word-embedded deep neural networks,* Social Network Analysis and Mining, volume 13(1) , pages 1-12, 2023.

   (c) Surajit Giri, Sayak Das, Sutirtha Bharati Das, and Siddhartha Banerjee, *SMS Spam Classification–Simple Deep Learning Models With Higher Accuracy Using BUNOW And GloVe Word Embedding,* Journal of Applied Science and Engineering, volume 26(10) , pages 1501-1511, 2022.

   (d) Surajit Giri, Siddhartha Banerjee, Kunal Bag and Dipanjan Maiti, *Comparative Study of Content-Based Phishing Email Detection Using Global Vector (GloVe) and Bidirectional Encoder Representation from Transformer (BERT) Word Embedding Models,* First International Conference on Electrical, Electronics, Information and Communication Technologies (ICEEICT),Pages 1-6, 2022.

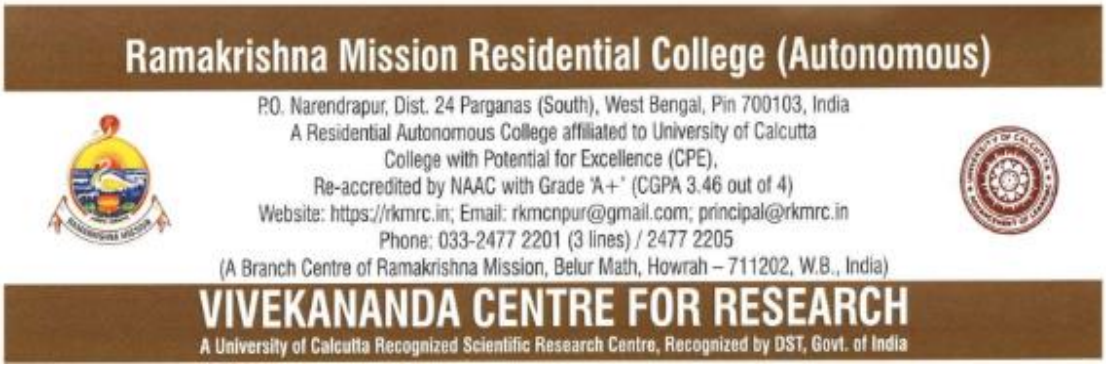
   (e) Siddhartha Banerjee ,Surajit Giri ,Debayan Das and Pramit Kumar Mandal, *An Approach to Predict the Location of Crime Using Machine Learning.* Advances in Intelligent Systems and Computing book series (AISC,volume 1397), pages 943–951, 2021.

4. List of Patents : None.

5. List of Presentations in National/International/Conferences/Workshops :

   (a) Surajit Giri, Siddhartha Banerjee, Kunal Bag and Dipanjan Maiti, *Comparative Study of Content-Based Phishing Email Detection Using Global Vector (GloVe) and Bidirectional Encoder Representation from Transformer (BERT) Word Embedding Models,* First International Conference on Electrical, Electronics, Information and Communication Technologies (ICEEICT),Pages 1-6, 2022.

   (b) Siddhartha Banerjee, Surajit Giri, Debayan Das, Pramit Kumar Mandal, *Soft Computing for Security Applications*, Advances in Intelligent Systems and Computing book series (AISC,volume 1397), pages 943–951, 2021.



Ramakrishna Mission Residential College (Autonomous)

P.O. Narendrapur, Dist. 24 Parganas (South), West Bengal, Pin 700103, India
A Residential Autonomous College affiliated to University of Calcutta
College with Potential for Excellence (CPE).
Re-accredited by NAAC with Grade 'A+' (CGPA 3.46 out of 4)
Website: https://rkmrc.in; Email: rkmcnpur@gmail.com; principal@rkmrc.in
Phone: 033-2477 2201 (3 lines) / 2477 2205
(A Branch Centre of Ramakrishna Mission, Belur Math, Howrah – 711202, W.B., India)

VIVEKANANDA CENTRE FOR RESEARCH
A University of Calcutta Recognized Scientific Research Centre, Recognized by DST, Govt. of India

DECLARATION OF SUPERVISOR

This is to certify that Sri Surajit Giri ***(Registration No. A03/Ph.D./COM/01/2020)*** has worked under my guidance and supervision for the thesis entitled ***"Design and Analysis of Machine Learning Algorithms for Predicting Pattern and Trends of Crime in India"*** which is being submitted to Vivekananda Centre for Research (A University of Calcutta recognized Research Centre) at Ramakrishna Mission Residential College (Autonomous) for the degree of Doctor of Philosophy.

To the best of my knowledge the thesis is the result of his own investigation into the subject, and no part of the thesis was submitted to any other University/Institutions for any research degree. Sri Surajit Giri has fulfilled all the requirements under the Ph.D. regulations of Vivekananda Centre for Research (A University of Calcutta Recognized Research Centre).

Date: ……………………

Place: …………………

……………………………….

(Signature of the Supervisor)

(DR. Siddhartha Banerjee)

Acknowledgements

At the outset I would acknowledge the patient efforts of my supervisor **DR. Siddhartha Banerjee** in deciding upon the problem which finally has led to this thesis. Without his iterative feedback and suggestions it would not have been possible to come up with this direction of research. I am grateful to him for his painstaking review of my work and meaningful suggestions which ultimately has given shape to this thesis.

____

Signature

Abstract

Crime is very a old concept which transmitted in our society from generation to generation. No one is safe today. Crime is a social evil. Our society suffers a lot because of crimes committed by its members. The world is full of criminals and criminal activities. The rising wave of crime nowadays has caused alarm to anyone. Due to the constant and continuous operation of criminals, the peace and happiness of our society have been hampered. In this internet era, with the development of technology, criminals are using scientific techniques while committing crimes. They use SMS (Short Message Service), Email, websites, etc. for their criminal activities. The main objective of the present work is to design and analyze different machine learning algorithms for predicting patterns and trends of crimes in India. The focus is given both cyber crime and traditional crimes. In this respect, six tasks are performed - i) the pattern of crime rate and different factors influencing the traditional crimes against children in India are analyzed and algorithms are designed to predict the geological location of a crime, ii) algorithms are designed to identify spam SMS correctly, iii) efficient algorithms are developed for annotation detection of cyber-bullying messages, iv) algorithms are developed to detect phishing emails, v) algorithms are designed for phishing websites detection, and finally vi) trends and prediction of cyber-crime in India are analysed. To achieve the said objectives different statistical, machine learning, and deep learning models are proposed.

In the first part, traditional crimes are considered. Children are a valuable strength of a nation. The growth rate of crimes against children in different states and union territories of India, during the period 2001 to 2020 is studied. It has been found that Delhi, A&N Islands, Chandigarh, Sikkim, and Madhya Pradesh have a higher growth rate. While investigating the role of different socio-economic factors, it has been observed that the digitization rate and urbanization rate have a strong influence on increasing the rate of crimes against children. It has also been observed that the rate of crime decreases as the literacy rate increases. In order to prevent crimes, it is very important to recognize the patterns of the criminal activities. If the crime patterns of different geological points of a city are known to the Police and Detective agencies, then they can work more efficiently in order to solve the problem. The proposed methodology provides an automated technique to predict the geological location of a crime depending upon the date, time, and type of the crime from past crime behavior. Linear Regression and Support vector Regression algorithms are used for this purpose. The proposed methodology is

tested on a dataset containing the crime records of Indore city in the month of February and March 2018. The results obtained using Linear Regression and Support vector Regression algorithms are compared and it is found that Support vector Regression provides better results.

In the second part of the thesis, different aspects of cyber crimes are taken into account and different solutions are proposed to reduce cyber crimes. At first, the problem of Spam SMSs (Short Message Service) are considered. Unwanted text messages are called Spam SMSs. It has been proven that Machine Learning Models can categorize spam messages efficiently with great accuracy. However, the lack of proper spam filtering software or mis-classification of genuine SMS as spam by existing software, the use of spam detection applications has not become popular. In this work, we propose multiple deep neural network models to classify spam messages. Tiago's Dataset is used for this investigation. Initially, the pre-processing step is applied to the messages in the data set, which involves lower casing the text, tokenization, lemmatization of the text, and removal of numbers, punctuations, and stop words. These pre-processed messages are fed in two different deep learning models with simpler architectures, namely the Convolution Neural Network and a hybrid Convolution Neural Network with a Long Short-Term Memory Network for classification. To increase the accuracy of these two simple architectures, BUNOW (Binary Unique Number of Word) and GloVe (Global Vectors) word embedding techniques are incorporated with deep learning models. BUNOW and GloVe are popular choices in sentiment analysis, but in this work, these two word embedding techniques are tried in the context of text classification to improve accuracy. The best accuracy of 98.44% is achieved by the Convolution Neural Network with a Long Short-Term Memory (CNN-LSTM) BUNOW model after 15 epochs on a 70% - 30% train-test split. The proposed model can be used in many practical applications like real-time SMS spam detection, email spam detection, sentiment analysis, text categorization, etc.

In recent times, online harassment due to cyber bullying has significantly increased with the growth of social media users. Cyber bullying is a technique to harass users using electronic messages. Many researchers attack this problem using natural language processing. Most of them detect whether a message is a bully or not. In this paper, multiple deep learning models are introduced to detect not only bullying messages but also the annotation of cyber bullying. Annotation detection of cyber bullying message, assigns a proper description in which category a message belongs to. The advantage of annotation detection is to warn the user by giving an alert message with proper annotation when the user sends or posts a message on social media. If this feature is combined with popular social network sites like Facebook, Twitter, WhatsApp, etc, this can be an additional filter to alert the user that they are going to post or send a bullied message of which type. Social media messages are unstructured as it includes text, URL links, emojis, abbreviations, etc. Most of the previous works are conducted to detect bullying messages only considering important words in the text, neglecting the other attributes in the message like URL links, emojis, and abbreviations. In this paper, an advanced pre-processing technique is proposed by considering some of the attributes in the messages like URL, abbreviation, number, emojis, etc. to detect bullying messages. In this work, six models i.e. three deep learning models combined with two different word-embedding techniques have been employed for annotation detection. The performances of each of these six models are measured twice, by employing traditional pre-processing and proposed advanced pre-processing. The experimental results show that the advanced pre-processing works better in the case of all six models.

In our daily life, among different types of cyber crime, one of the major threats is a Phishing attack. Due to the COVID-19 pandemic, the growth of internet users in the last two years has increased enormously. At the same time, Phishing attacks have also increased. Many internet users are suffering financially and their personal information is being misused for different crime purposes. Using a forged URL the attacker attempts to access the user's credential information like bank details, credit card details, personal information, etc. In the modern digitization era, Phishing website detection algorithm is a challenging task to safeguard internet users from these attacks. In this paper, a new methodology has been proposed to detect phishing websites using an ensemble learning approach. Initially, important features are collected from different website URL addresses using the Random Forest Regressor model. Then different supervised machine learning algorithms namely Naïve Bayes (NB), K-Nearest Neighbour (K-NN), Support Vector Machine (SVM), Random forest (RF), Bagging, Logistic Regression (LR), and Neural Network (NN) are used to detect phishing website depending on collected features. Finally, to improve the accuracy of phishing website detection, an optimal stacking model with a greedy approach has been employed. The widely used data set from the UCI Machine Learning Repository has been used to evaluate the performance of the proposed model. The proposed method achieves 96.73% accuracy in the detection of a phishing website.

The use of the internet in India is growing rapidly and in the near future, the country will be in the digital era. But with this growth, the rate of cyber crime is also increasing and threatening the capabilities of the investigation system. Physical, economic, social, and political security are under challenge due to Cyber attacks. Different sectors like government, industry, and academic institutions give extensive effort to successful anticipation and forecasting of such cyber attacks. Recently Government of India has banned 59 Chinese apps to ensure cyber security. The data generation regarding cyber crimes is also increasing nowadays which are mostly digital in nature. In the last part of this work, cyber crimes data during the years 2011 to 2018 are analyzed to investigate the trends of cyber crime in India. Two different data-processing approaches namely the linear regression model and polynomial regression model are used for this analysis. It is observed that the polynomial regression of degree 4 best suits the crime pattern in India with a R^2 value of approximately 0.98. These observations are used to predict the cyber crime events in 2019 and 2020. To test the accuracy of the prediction, we compare our prediction with actual cyber crime records of 2019. It is observed that prediction suits the actual records with tolerable accuracy.

Table of contents

List of figures

List of tables

Chapter 1

Introduction

1.1 Background

“Everyone has a right to live in a community that is safe. Just as it should not pose threats to the health of residents, people should not have to fear for their personal safety and/or the safety of their belongings” [1].

The design and analysis of patterns of criminal behavior over a particular crime is a major element for law enforcement agencies either in capturing the criminals of a crime or in curtailment and prevention of crime. This approach towards the problem of crime is clearly governed by a study called **"crime analysis"**. According to Gottlieb, Arenberg, and R Singh in their study titled "Crime Analysis: From First Report to Final Arrest" [2], crime analysis is a collection of processes that are systematic and analytically aimed. Crime analysis provides relevant data regarding trends and patterns in crime commission. It is directly related to the improvement in the operational aid in the deployment of resources, the process of investigation, and the clearance of cases. It also helps in the research and planning required for the functioning of tactical units and administrative services. Crime analysis involves a systematic approach or research of the data related to the crimes. The data are of two types namely, qualitative and quantitative. **Qualitative data** are measures of values or counts and are expressed as numbers. Using qualitative data we recognize the characteristics of data. **Quantitative data** are data about numeric variables. Both types of data collection and interpretation are engaged by crime analysis to analyze, interpret, and find crime patterns and trends by using statistical measurement, machine learning models, or deep learning models.

It is universal truth that, today's children are the future of tomorrow's country. But looking at today's scenario, the safety of children has become a major problem. An increasing number of crimes against children is proving that the condition of the country is not secure. According to the National Crime Record Bureau (NCRB), 109 children were sexually abused every day in India in 2018. Innocent children are falling victim to crimes such as murder, rape, child pornography, physical and emotional abuse, sexual abuse, buying and selling, etc. There are various laws have been passed to prevent crimes, and many kinds of policies have also been made for the welfare of children in the country. Yet the crimes are still growing. There is a need to go into the scientific study of every state and union territory of India to visualize the actual growth rate. Not only finding the crime rate, but it is also urgently required to find out the root of the reasons for crimes. To achieve that goal, different socio-economic factors like literacy rate, unemployment rate, digitization rate, and urbanization rate must be analyzed. In recent years, the Government of India published a large amount of crime data on the web. The researchers analyzed those data in a scientific manner and proposed different models to prevent crimes [3, 4]. Researchers have also started to propose various crime prediction methods [5, 6]. Our goal is to find the location of a crime by studying the crime patterns and behaviors of the crimes using a machine learning approach. The outcomes of predicting the location of a crime can be used in various ways. Firstly, cities could use such data to better plan police patrols, targeting the locations and times at which crimes are more likely to have occurred. Secondly, the residents and tourists can be alerted to ignore such locations and times when scheduling their outdoor activities.

With the advancement of technology, almost everyone depends on the Internet for their day to day activities. In India, the threat of cyber crime is also growing rapidly. The investigation of cyber crime in India started by launching an "FIR" (First Information Report), then different investigating agencies investigated and penalized the criminals under the Indian Penal Code, 1860 ("IPC") and the Information Technology Act, 2000 ("IT Act"). National Crime Records Bureau (NCRB) defines that cyber criminals perform cyber crimes to earn money, to become famous, to just have fun, to sexually exploit someone, to blackmail someone, for developing own business, computer hacking, forgery, fraud, for selling/purchasing illegal contents, to take a revenge of someone, or to do a prank with someone, and so on. By using actual cyber crime data, available on different government websites as an input, prediction of cyber crime is urgently necessary in India. Some techniques should also be needed to reduce the occurrence of cyber crime.

Short Message Service (SMS) is the most commonly and extensively used communication medium. SMS is a text communication platform that allows mobile phone users to exchange short text messages for the purpose of advertisement and promotion of products, banking updates, agricultural information, flight updates, and internet offers. As the popularity of the platform has increased, the number of unwanted messages known as spam is also increased. The purposes of SMS spammer are spreading unwanted messages for commercial or financial gain such as the purchase of lottery tickets, the disclosure of credit card information, advertisement and marketing of various products, sending political issues, loan offers, spreading inappropriate adult content and internet offers. SMS spamming has become a major irritation for mobile subscribers. At worst, it can cause a significant financial impact. Thus, a proper SMS spam detection technique is a significant necessity.

Spam email is unsolicited and unwanted junk email sent by spammers mostly for commercial purposes. Spam email can be dangerous because nowadays it is also widely used for transferring harmful malware and electronic viruses. There are several problems due to spam emails such as communications overload because it uses network bandwidth and traffic, fake emails that attempt to trick us into giving out sensitive personal information, and fraudulent messages from hacked email accounts. Spam emails are also responsible for waste of valuable time, because the recipient deletes spam emails manually, and the loss of an important email that accidentally gets deleted along with the embarrassment of spam email. It goes without saying that spam is an irritation for all of us. Therefore, spam email classifications are required urgently.

In this internet era, the rate of internet users is growing rapidly. Mostly they use the Internet for the purpose of communication, entertainment, education, shopping, and so on. With the progression of online life, criminals view the Internet as an opportunity to transfer their physical crimes into a virtual environment. Phishing attacks are becoming successful due to a lack of user awareness, not all users can identify fake websites that mimic of original websites. There are various types of web threats - such as identity theft, stealing of personal information, financial loss, etc. The world is suffering several economic damage due to Phishing attacks. So, detecting these types of emails and websites is an important task.

Cyber-bullying is the medium of digital technology of posting personal information, rude texts, pictures, or videos designed to bully someone. Nowadays, different social media sites are very popular and users can easily share their thoughts and feelings. Cyber-bullying is becoming increasingly common among young people. The effects of cyber-bullying are very serious and include mental health problems, growing stress and anxiety, unhappiness, nervousness, depression, and low self-confidence, which affects in long-term future of society. Cyber-bullying annotation detection techniques can help users while they are going to posting a text or messages to understand the type of messages like racism, sexism, or normal.

1.2 Past Works

The criminal cases in India are increasing rapidly due to which the number of pending cases is also growing. This continuous increase in criminal cases is moving our society towards a worse situation and also proving to be difficult to classify and solve. Despite the high number of law enforcement agencies in the country, the ratio of law enforcement agencies to citizens is highly outnumbered with many flaws at various levels of the system. Computers can do background work and make decisions intelligently and automatically [7]. The crime-solving agencies can do a better job if they have a good idea of the pattern of criminal activities. Many researchers proposed different statistical / mathematical / machine learning / deep learning models to analyze the patterns, and trends and predict the activities of criminals. An outline of these past works is described in the following sections.

1.2.1 Crime against children

During the COVID-19 pandemic, violence against children is described by Cappa et al. [8], and during this pandemic period psycho-social risks for children due to the economic crisis are described by Ramaswamy et al. [9]. Various aspects of child abuse, and neglects, and their impacts are also studied [10, 11].

To understand the real picture of crime against children in India, different statistical parameters are analyzed [12–15]. They have collected data from NCRB (National Crime Records Bureau) for different periods on crime against children in India and analyzed the crime rate, and growth rate of different states in India using different statistical measurements. They also studied different factors like female education, Scheduled Caste (SC), Scheduled Tribe (ST), Gross Domestic Product, Urban population, and the unemployment rate which influence the rate of crime against children. A correlation between the crime rate and the literacy rate was determined by Bodhgire et al. [16] using a machine learning regression model. Different factors like the police force, arrest rates, charge sheet rates, conviction rates, and quick disposal of cases in India were also studied by Dutta et al. [17].

Analyzing demographic information and the usage of mobile network infrastructure to predict the hot spot area of criminal activities are studied by Bogomolov et al. [5]. To predict the geological location of the crime, different supervised machine learning approaches are proposed [6, 18, 19]. The unsupervised machine learning clustering method used by Yadav et al. [20], to study the patterns of criminal behaviour and geographic criminal history. To predict crime in real-time, a sentiment analysis has been carried out on Twitter data [21].

1.2.2 Spam SMS Classification

To classify spam SMS, most of the researchers use either traditional machine learning approaches or deep learning approaches. Different performance measurement tools are used to evaluate their proposed model. Some of them use different word embedding techniques to increase the performance of the proposed model.

Traditional machine learning approaches are applied heavily for spam SMS classification [22–25]. Mostly they used Naïve Bayes, Random Forest, Logistic Regression, Support Vector Machine, Gaussian Naïve Bayes, Boosted Naïve Bayes, and Boosted C4.5. Sethi et al. [23] uses frequencies of the words as the input vector. Navaney et al. [24] uses the document term matrix as an input vector. Furthermore, a hidden Markov model is used by Xia et al. [26], to detect spam SMS where every word is processed directly. They extend their research [27] by assigning weight for each word that indicates the probability of the message being detected as spam or ham SMS. Diallo et al. [28] tried to find a similarity matrix in the multi-view document clustering algorithm and further enhanced their proposed model [29] by considering Cosine similarity, Euclidean distance, and magnitude difference.

In the traditional machine learning approach feature engineering is a time-consuming process. To overcome that problem many researchers [30–39] studied deep neural networks to classify spam SMS. Mostly they used CNN, LSTM, BLSTM, and RNN. Some of the researchers use a hybrid model by merging multiple deep neural networks. Furthermore, Shaaban et al. [40] proposed a deep ensemble approach for spam detection. For feature extraction, they used convolution and pooling layers, and their base classifier consists of random forests and extremely randomized trees.

1.2.3 Annotation Detection for Cyber-bullying Messages

Many researchers used supervised machine learning models for cyber-bullying detection. They used logistic regression [41, 42], Random forest [41–44], Support vector machine [41, 42, 44], Naïve Bayes [41, 43], and Decision Tree [44] for that purpose. The disadvantage of the supervised machine learning approach is that, it does not consider the context of words. To overcome this advantage, deep neural networks are introduced. Pronunciation-based cyber-bullying message detection technique [45] using Convolutional Neural Network (CNN) has been proposed. Word embedded convolution neural network was also tried by some researchers [46, 47]. Different deep learning models like CNN, Long Short-Term Memory (LSTM), etc. are used by combining GloVe and SSWE (Sentiment Specific Word Embedding) word embedding techniques [48–51]. Many researchers also used Recurrent Neural Networks (RNN), and Gated Recurrent Units (GRU) for detecting cyber-bullying [52, 53]. Locality Sensitive Hashing with Similarity-Based Word Embedding (LSHWE) [54] has been used for the detection of cyber-bullying messages.

1.2.4 Phishing Email Classification

We have studied three approaches, namely the blacklist mechanism, the machine learning approach, and deep neural networks for detection of phishing emails. In the blacklist mechanism, back-listed senders are stored on the database depending on their past activity. In this approach, the search process is time-consuming. To overcome that problem, many researchers tried machine learning approaches for phishing email classification. Support Vector Machine (SVM) [55, 56], K-Nearest Neighbour (KNN) [57], Naïve Bayes (NB) [58], and Decision Tree (DT) [57, 58] have been tried in this context. Many researchers also used an ensemble model [59–61] which is a combination of multiple base models for detecting phishing emails. Traditional machine-learning approaches cannot find the optimal features of the data set. To overcome that limitation of the machine learning approach, many researchers used deep learning networks. Srinivasan et al. [62] proposed distributed word embedding method with Deep Learning for spam email detection. Sumathi et al. [63] used random forest and deep neural network classifiers for spam email classification. Advanced deep convolution neural network algorithms was proposed by Soni [64] for Spam e-mail detection. Using a distinct neural network, Castillo et al. [65] detected email threats. Fang et al. [66] used an improved Recurrent Convolution Neural Networks (RCNN) model and their proposed model consists attention mechanism. Manaswini et al. [67] used the THEMIS model proposed by Fang et al. [66] and to improve the accuracy, they used Recurrent Convolutional Neural Networks (RCNN). Vinaya Kumar et al. [68] have proposed a new model named Deep Spam Phish Email Net (DSPEN) over a deep Learning framework. Hiransha et al. [69] proposed a convolution neural network(CNN) deep learning model for phishing email detection. Chetty et al. [70] classified phishing emails using a deep learning model.

1.2.5 Phishing Website Detection

There exist several approaches for the detection of phishing websites, namely the black list and white list approach, heuristic-based approach, content-based approach, visual similarity-based approach, and machine learning-based approach. Google safe browsing and Phish Tank are used to store malicious URLs or IP addresses of blacklisted websites. The limitation of that technique is, it cannot detect newly generated phishing websites. In the heuristic approach [71], by scanning web pages of known attacks, a signature database is created. It is time-consuming and attackers easily bypass it through obfuscation. In the content-based approach [72], a detailed analysis of page content is required. It relies on third-party services. In the visual similarity approach [73], phishing websites look similar to non-phishing websites by embedding objects like images, scripts, etc. In this approach stored snapshots of different legitimate websites are compared with newly generated websites. It is time, and space-consuming. Many researchers use different machine learning models like Random

Forest (RF) [74, 75], Naïve Bayes [76, 77], Support Vector Machine (SVM) [77, 78], etc. for classifying phishing or legitimate websites. Many researchers [79–81] found that the performance of the ensemble predictive methods gives better accuracy compared to the individual machine learning algorithm. The ensemble machine learning approach combines multiple base models to increase the accuracy. Ensemble learning can be classified into three categories, namely Bagging, Boosting, and Stacking. The Bagging and Boosting ensemble method is based on voting but in Stacking, lower-level base learners are combined to produce high-level learners. Many researchers [82–85] use ensemble machine learning techniques to classify phishing websites.

1.2.6 Analyzing Trends and Prediction of Cyber Crimes

Researchers use different models to analyze and predict cyber crime offenses. The features of cyber crime incidents as a classification system for related offenses and a schema that binds together the various elements are studied by Tsakalidis et al. [86]. They proposed a comprehensive list of cyber crime related offenses which are ordered in a two-level classification system based on specific criteria. Unpredicted patterns of cyber crimes are discovered by Ganesan et al. [87]. They have considered a database that includes cyber-bullying, stalking, scams, robbery, identity theft, defamation, and harassment. Pravakaran et al. [88] studied five types of crime namely - fraud detection, traffic violence, violent crime, web crime, and sexual offense. They used various machine learning algorithms to analyze crimes. K-means clustering unsupervised machine learning algorithms studied by Kiger et al. [89]. They captured fraud, malware, spam, and digital piracy. Some cyber crime prevention techniques are also recommended by Soomro et al. [90]. They discussed different types of cyber crime such as Burglary via Social Networking, Social Engineering, Phishing, Identity Theft, Cyber-Stalking, etc., and their prevention techniques correspondingly.

1.3 Present Works

Objectives:

- The Objectives of the present work are to analyze different types of crimes and their trends and also to design new methodologies that can work better, from the existing models. The objective includes the following

- Analyzing the role of different socio-economic factors over crime against children in India using statistical and machine learning approaches and predicting the location of a crime from past crime incidents.
- True Classification of spam SMSs using different deep learning and word embedding models with high accuracy.
- Annotation detection of Cyber bullying messages using word-embedded deep neural networks.
- Identification of Phishing email using Global Vector and Bidirectional Encoder Representation from Transformer Word Embedding Models.
- Detection of Phishing websites using ensemble learning approach.
- Finding trends of Cyber crimes in India and its prediction method.

1.3.1 Contributions of the Work

Major contributions of the present work are as follows:

- Development of methodologies for finding the growth rate of crimes against children in different states and union territories of India by considering the past twenty years of data. The study also analyzes the role of different socioeconomic factors which influence the growth rate. An automated technique is suggested to predict the geological location of a crime depending upon the date, time, and type of the crime from past crime events [91].
- Designing multiple deep neural network models to classify spam messages with high accuracy. The proposed model can also be used in many other practical applications like real-time SMS spam detection, email spam detection, sentiment analysis, text categorization, etc. [92].
- Development of an annotation detection technique for Cyber bullying messages. If this feature is combined with popular social network sites like Facebook, Twitter, WhatsApp, etc. This can be an additional filter to alert the user that they are going to post or send a bullied message of which type [93].
- Development of two word embedded deep neural network models to detect Phishing emails by analyzing the content of the email, and performing a comparative study between them [94].
- A new methodology has been proposed to detect phishing websites using an ensemble learning approach.

- Analyzing the cyber crimes data in India during the years 2011 to 2018, using different machine learning approaches to investigate the trends of cyber crimes, and developing a prediction model to predict trends of cyber crimes in the future.

1.3.2 Overview of the Dissertation

To analyze the pattern and trends of the crime incidents and to develop different methodologies for reducing the crime events, the overall work is shown in Figure 1.1. The dissertation includes both traditional and cyber crimes. It includes analysis of crimes against children in India and predicting the location of the crime, classification of spam messages, Annotation detection of cyber bullying messages, Phishing email classification, Phishing website detection, and Trends and prediction of cyber crime incidents in India.

In this dissertation, a very small segment of the traditional crimes, namely the crimes against children are considered. Children are being forcefully indulged in many activities like trafficking and begging, they are being sold, enslaved, exploited, physically abused, and killed too. We consider children to be the future of our nation but it would not be incorrect to say that they have been neglected a lot. The crimes that are committed against children happen because of their incapability to appreciate the nature of the offenses. Most of the children are victimized by their relatives, caretakers, parents, guardians, or any other who are being employed to look after them. A statistical approach has been introduced in chapter 2, to understand the actual growth rate of crime against children in different states, and union territories in India. We have also analyzed different socio-economic factors which influence the rate of crime against children. Due to the increasing rate of crimes, it is a challenge how to reduce crime events. An automated technique has been proposed that predicts the geological location of a future crime event depending upon the date, time, and type of the crimes from past crime records. If the crime patterns of different geological points of a city are known in advance to the law and order agencies, such as police or detectives then they can work more efficiently in order to solve the problem.

Due to the lack of traditional crimes data, rest of the dissertation is devoted to handling cyber crime. Nowadays, almost everyone has a mobile. SMS spam attacks arrive from National and International networks without proper filtering capabilities creating massive problems for subscribers. Rogue banking application alerts are sent to subscribers leading to bank fraud. There are several objectives of spam SMS. It can be data theft, identity theft, or system manipulation. The ones who lead them are criminal workforces who organize and coordinate a plan targeting people to gain maximum profit. In chapter 3, initially text pre-processing methods have been discussed. Two word embedding models are used to convert text messages to numeric word vectors and deep learning methods are

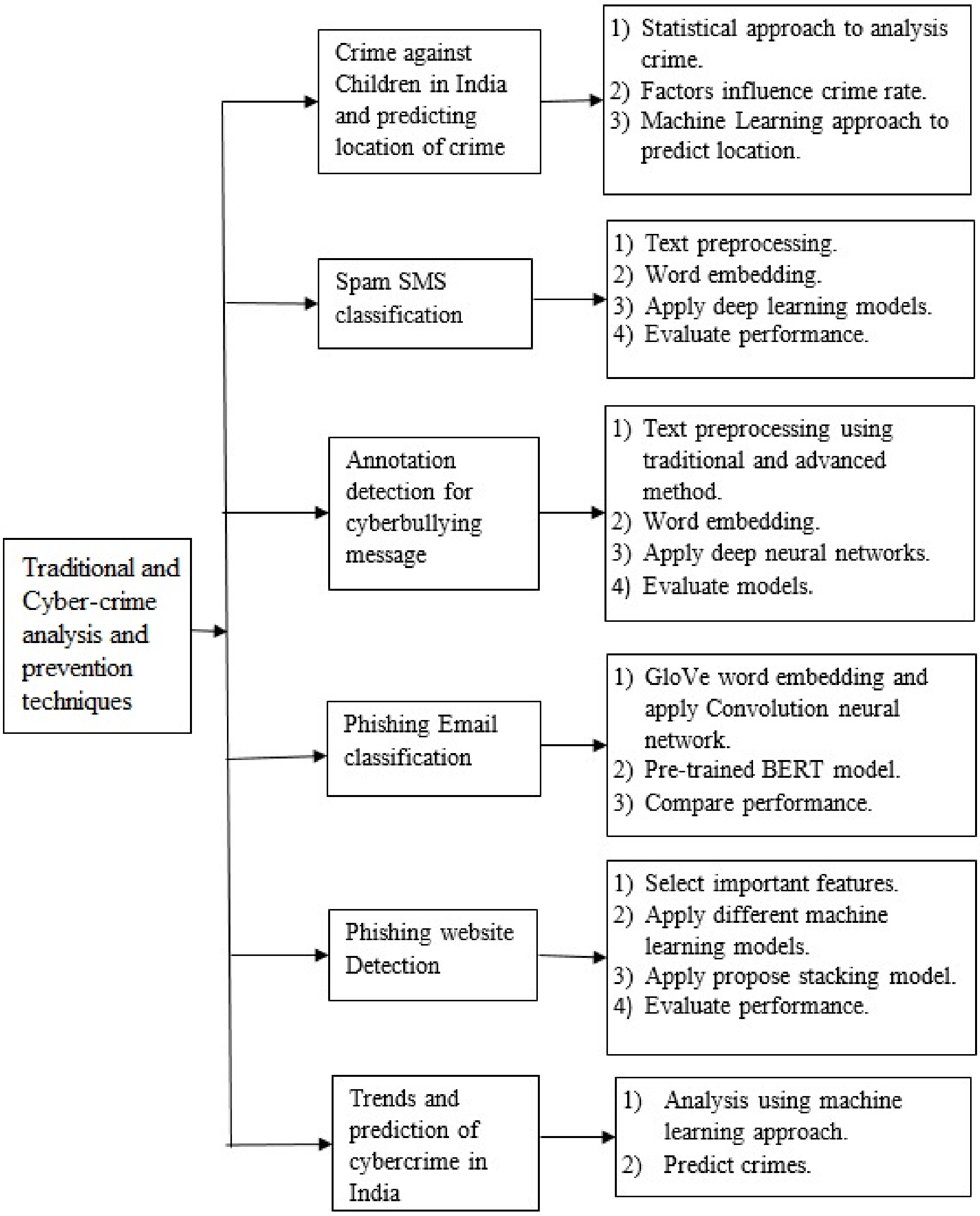

Traditional and Cyber-crime analysis and prevention techniques
Crime against Children in India and predicting location of crime
1) Statistical approach to analysis crime.
2) Factors influence crime rate.
3) Machine Learning approach to predict location.
Spam SMS classification
1) Text preprocessing.
2) Word embedding.
3) Apply deep learning models.
4) Evaluate performance.
Annotation detection for cyberbullying message
1) Text preprocessing using traditional and advanced method.
2) Word embedding.
3) Apply deep neural networks.
4) Evaluate models.
Phishing Email classification
1) GloVe word embedding and apply Convolution neural network.
2) Pre-trained BERT model.
3) Compare performance.
Phishing website Detection
1) Select important features.
2) Apply different machine learning models.
3) Apply propose stacking model.
4) Evaluate performance.
Trends and prediction of cybercrime in India
1) Analysis using machine learning approach.
2) Predict crimes.

Fig. 1.1 Overall Organization of the Dissertation.

proposed to classify spam SMS. Finally, we have discussed the performance of the proposed model with existing models.

Cyber bullying has become very major problem in today's digital era. It is not a simple matter, it can destroy the image of a person, demoralize a person, a person can go into depression, loss of self-confidence, and have a lot of harmful effects on the victim. Most of the researchers use traditional methods to check whether a message is bullied or not. In chapter 4, an annotation detection technique for cyber bullying messages has been proposed. An advanced text pre-processing technique has been introduced to increase the accuracy. Two word embedded models namely BUNOW and GloVe are used for converting text messages to numeric word vectors. Three deep neural networks namely, CNN, LSTM, and BLSTM models are applied over word vectors. Finally, different performance evaluation tools are used to measure the performance of proposed models.

Phishing emails are unwanted junk emails. Because of its low cost, most of the spammer uses it for commercial purposes. It can be very harmful if criminals send malicious links that can infect our computers with viruses and malware. In chapter 5, two deep learning models have been proposed for the detection of spam emails by analyzing the content of emails. For identifying the context of a word, GloVe, and Bidirectional Encoder Representations from Transformers (BERT) models are used separately. Finally, the performance of both models is compared.

Ensemble methods are the combination of multiple machine learning approaches. In chapter 6, an ensemble approach has been proposed for detecting phishing websites. Important feature selection is a crucial task for phishing website detection. For this context, we have used the Random Forest Regressor model. After finding important features, seven machine learning models namely Naïve Bayes(NB), k-Nearest Neighbour (k-NN), Support Vector Machine (SVM), Random forest (RF), Bagging, Logistic Regression (LR) and Artificial Neural Network (ANN) are used as base models to evaluate the performance of these individual base models. Finally, using the optimal stacking model with a greedy approach, we have aggregated individual models one by one to get better accuracy.

In recent years the Government of India published cyber crime data publicly. The researchers use those data and analyze the behavior of crime, and the nature of crime, and derive some anti-crime strategies. The prediction of crime is a growing field of research. In chapter 7, the cyber crime records for 28 States and 8 Union Territories of India published by the National Crime Records Bureau during the year 2011 to 2018 have been analyzed. The trends of cyber crimes in India as a nation and also the trends for individual states and union territories are observed in different sections under the IT Act 2000. These observations are used to predict the cyber crime events in India as well as individual states and territories for the years 2019 and 2020.

Finally, the work is summarized and the scope for further research is outlined in Chapter 8.

Chapter 2

Crime against Children

2.1 Introduction

India is the fastest-growing developing country in the world. Crime against children is a major issue, because today's children will play an important role in the future nation. The Government of India delivers different laws to prevent crime against children. Some of these laws are the Child Labour Prohibition and Regulation (1986) [95], the Juvenile Justice Act/Care and Protection (2000) [96], the Child Marriage Prohibition Act (2006) [97] and the Protection of Children from Sexual Offences Act (POSCO-2012) [98]. Recently, on March 15th, 2021, Lok Sabha passed the Juvenile Justice (Care and Protection of Children) [99], but the rate of crime against children increased day by day. Figure 2.1 shows year-wise total crime against children between the years 2001 and 2020. The data are collected from the NCRB (National Crime Records Bureau) [100]. A steady upward line signifies that the crime percentage is very high. During the COVID-19 pandemic time in the year 2020, it is observed that the crime against children is slightly reduced but the number of crime incidents is 128531 which is also a huge number. If we consider 2019 it increased by more than 511% over the past decade (148090 in 2019 over 24201 in 2009).

Socio-economic conditions, as well as the lifestyle of people, are affected by criminal activities that are very familiar to every part of the world today. The criminal activities include different types of events like pick pocketing, robberies, rapes, murders, smuggling, etc. The lives of common people are at risk due to these criminal activities. People feel very insecure about all the anti-social activities happening around them. The works of the criminals are very organized now and they hardly leave any clue behind to trace them. It is also observed that in some cases the criminals have linked up with

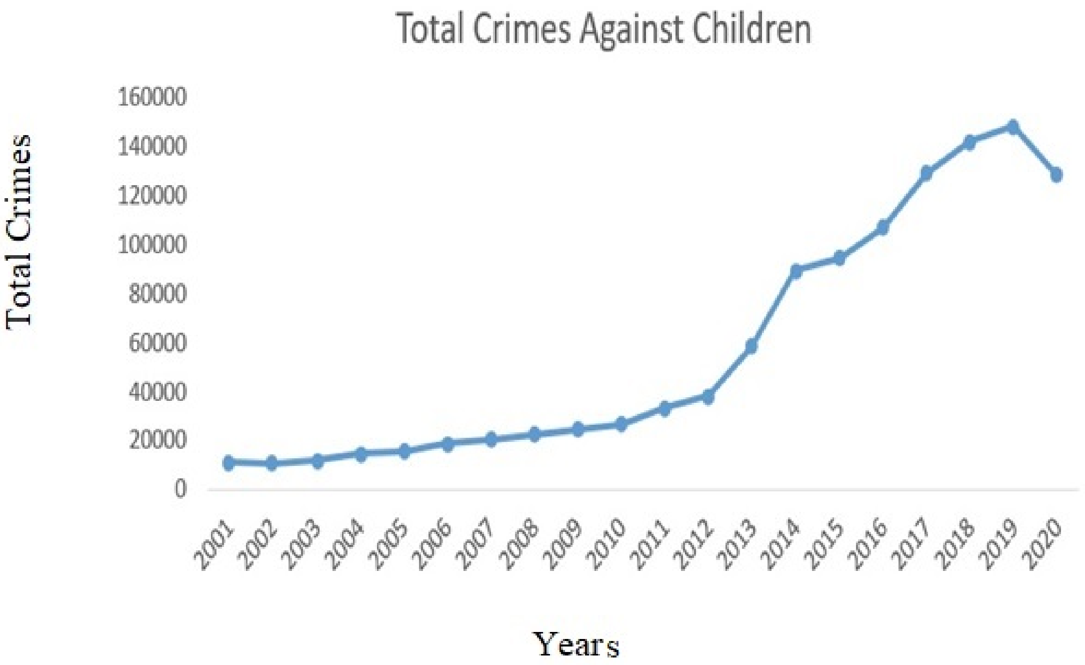


Fig. 2.1 Total crime against children during the year 2001 to 2020.

national and international agencies. Thus dealing with such criminal activities becomes a big issue for the government.

The scientific study for understanding the behaviour of crime, the nature of the crime, and deriving some anti-crime strategies by identifying the characteristics of crime is known as **criminology. Crime analysis** is a sub-branch of criminology. It studies the pattern of the crimes and tries to find indicators of the events. But the presence of a huge amount of data and due to increasing crime rate, make it impossible for security authority and their personnel to manually analyze these data and find the hidden secrets within these data [18, 101–103]. The application of data science technology appeared as a promising and reliable solution in the last few years to handle business strategies, smart decision-making, marketing, and weather and environment forecasting [104, 105]. The fact is that every incident holds a piece of valuable information that can be used for forecasting the occurrence of the incident in the future, i.e., the past is the true providence of the future [106]. Crime is also predictable since human nature is not reversible [107]. Human nature generally evokes periodically to perform the same event. With the advancement of big data and easy-to-implement algorithms for the analysis of data, the prediction of crime is a growing field of study.

This chapter is divided into two parts. The first part (PART - I) analyzes the growth rate of crime against children in different states and union territories in India from the past crime events during the period 2001 to 2020. The influences of four socio-economic factors on the crime rates are also determined. The second part of this chapter (PART - II) proposes an automated Machine Learning based approach to predict the the geological location of crime. To predict the geological location of the crime it is necessary to provide the date, time, and type of crime in the proposed model. The proposed methodology used Linear Regression and Support Vector Regression algorithms to predict future crime patterns from past criminal behavior. Testing has been done on a data set containing the crime records of Indore City in the months of February and March 2018. The data set contains the following types of crimes:

- Robbery (Act 379)
- Gambling (Act 13)
- Accident (Act 279)
- Violence (Act 323)
- Murder (Act 302)
- Kidnapping (Act 363)

The result obtained using Linear Regression and Support Vector Regression algorithms are compared and it is found that Support vector Regression provides a better result. This concept can also be applied to the data set of past crime events against children to predict the geological location of future crime in advance.

2.2 PART - I: Analysis of Crime Incidents against Children in India

2.2.1 Past Works

Day by day the crime rate of children is increasing considerably. Many researchers have studied the different impacts of crime on children. The study by Cappa et al. [8] based on reports and articles published in the middle of 1st March 2020 and 31st December 2020, their study mainly focused on violence against children during the COVID-19 duration. They found that during COVID-19, family violence and child abuse-related injuries increased and decreased in police reports. Various aspects of child abuse and negligence are discussed by Zeanah et al. [10] and their suggestion is to develop more legal and child protective service systems so that maltreated child can lead their normal life.

Hillis et al. [11] describe the different impacts of violence against children. They discussed many consequences of violence against children like injury, HIV and other infectious diseases, mental health, reproductive consequences, and impact on special populations. During the COVID-19 pandemic due to the economic crisis psycho-social risks for children are described by Ramaswamy et al. [9]. They have considered different issues like child labor, child trafficking, and child marriage, and their impact on child mental health. Jain et al. [108] conducted a survey of over 256 students before the COVID-19 pandemic and 118 students during the lockdown in India and concluded that the pandemic has affected our vulnerability to cyber bullying.

Different statistical measures are studied by the researchers to understand the behaviour of crimes. Maity et al. [12] considered 19 Indian states over the period 2001-2015 and analyzed the rate of crime against women. They have calculated the growth rate of different states and also concluded that female education, Scheduled Caste (SC), and Scheduled Tribe (ST) directly influence the rate of crime against women. Another study has been conducted by Mavi [13] during the period 2001-2011 on crime against children in India and concludes that murder, kidnapping and abduction, and rape of children highly contribute to crime against children. She also considered the different socio-economic factors like Gross Domestic Product (GDP), Urban population, and the unemployment rate and their influence on crime against children. Internet access and sexual offense against children are investigated by Shaik et al. [14]. They collected data from NCRB (National Crime Records Bureau) during 2000-2012 and did not find any correlation between the increases in sexual offensive crime with the growth rate of internet access. A correlation between the crime rate and the literacy rate was determined by Bodhgire et al. [16]. They used a regression model and predicted that if the literacy rate increased in different states then the crime rate may be decreased. Different factors like population density, sex ratio, minority population, poverty, and per capita income were studied by Gupta et al. [15] with the crime rate in India over the period of 2011. They investigate how these factors influence the rate of crime in India. Different factors also were studied by Dutta et al. [17] like the police force, arrest rates, charge sheet rates, conviction rates, and quick disposal of cases in India. They are studied based on the duration of 1999 to 2005.

By observing the above study, our investigation of crime against children consists of two parts, first, we have tried to find out the actual growth rate in different union territories and states in India over the period 2001 to 2020, Second, we have considered four factors, namely digitization rate, urbanization rate, literacy rate, and unemployment rate and find out the relationship with the rate of crime against children over the same period. The originality of this paper is, it tries to find the actual growth rate of crime against children with a novel examination in different states and union territories of India by considering twenty years of the data set and also investigating the factors that influence the rate of crime against children.

2.2.2 Materials and Methods

Our investigation is based on secondary data publicly available on different Government websites. The number of crime incidents against children in different states and union territories over the period 2001 to 2020 has been collected from the National Crime Records Bureau (NCRB) website [109]. The digitization rate, literacy rate, unemployment rate, and urbanization rate over the same period have been collected from the Open Government Data (OGD) Platform in India [110].

Basic Statistics: Initially to understand the rate of crime events against children over the period 2001 to 2020 we examined **Mean** (Eq. 2.1), **Standard Deviation** (Eq. 2.2), **Coefficient of Variation** (Eq. 2.3), **Max Rate**,(Eq. 2.4) and **Min Rate** (Eq. 2.5).

$$Mean(\mu) = \frac{\sum R_i}{N} \tag{2.1}$$

Where R_i denotes the rate of crime against children over the years 2001 to 2020 and N denotes the total number of years, In our study N=20.

$$Std.dev. = \sqrt{\frac{\sum(R_i-\mu)^2}{N}} \tag{2.2}$$

$$Coefficientofvariation(CV) = \frac{Std.dev.}{mean} * 100 \tag{2.3}$$

$$MaxRate = Max(R_i) \tag{2.4}$$

$$MinRate = Min(R_i) \tag{2.5}$$

Stationary: We have collected the rate of crime against children over the period 2001 to 2020 from NCRB which is a time-series data set. To check stationary we have used Augmented Dickey-Fuller Test (ADF). The equation for the ADF test is presented as 2.6.

$$\triangle Y_t = \alpha Y_{t-1} + \sum_{i=1}^{n} \beta_i \triangle Y_{t-1} + \varepsilon_t \tag{2.6}$$

where, $\triangle$ is the difference operator, y_t is the variable of interest at time t. α is the coefficient of the endogenous variable of y on difference form and is used for the actual test. The β coefficients are for the n lags of y on difference form. ε_t is an independent identically distributed residual term, that is the error term. The null hypothesis (H_0) is that the series has a unit root, so that is equal to 0, that is $H_0 : \alpha = 0$. The alternative hypothesis (H_1) is that the series has no unit root, so that is less than 0, that is $H_1 : \alpha < 0$. The evaluation is done with t-values using the corresponding test statistics is $t = \frac{\tilde{\alpha}-\alpha}{SE(\tilde{\alpha})}$. After Ordinary Least Square (OLS) estimation (SE), it can be compared to the relevant critical value for the Dickey-Fuller test. The p-value obtained by the test should be less than the significance level, in our study we have considered 5% significance, therefore if the p-value is less than 0.05, we reject the null hypothesis, that is the series is stationary.

Growth Rate: The growth rate of crime against children in the different states was calculated after performing the stationary test. We have used linear regression to calculate the growth rate by the equation 2.7.

$$lnY_(t) = \alpha + \beta_i t + \varepsilon_t \tag{2.7}$$

$$Growthrate = (e^{\beta_i - 1}) * 100 \tag{2.8}$$

Equation 2.8 denotes the percentage of the growth rate of crime against children over the period 2001 to 2020.

Instability Measurement:

To understand the true picture of the growth rate of crime against children we have used the Cuddy–Della Valle index for measurement of instability. It is calculated as equation 2.9.

$$I_x = CV * \sqrt{(1-\bar{R}^2)} \tag{2.9}$$

I_x denotes the Cuddy-Della Valle index. CV is the coefficient of variation and $\bar{R}^2$ is the adjusted coefficient. In this experiment, if $\bar{R}^2$<0 then we have considered an unadjusted R-squared value.

Multiple Regression model: In the second part of our study, we have considered different socioeconomic factors that directly influence the rate of crime against children in different states in India. We have considered a multiple regression model for this purpose. Our dependent variable is the rate of crime against children and considered four independent variables. The mathematical equation is presented in Eq. 2.10.

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \beta_4 X_4 + \varepsilon \tag{2.10}$$

Where, Y = Rate of crime against children

β_i = Regression coefficient i=0,1,2,3,4

X_1= Percentage rate of unemployment

X_2= Percentage rate of literacy

X_3= Percentage rate of digitization

X_4= Percentage rate of urbanization

ε = Denotes the error term

We have used the Ordinary Least Square (OLS) technique for estimation. Multi-collinearity among independent variables (X_1, X_2, X_3, and X_4) has been examined.

2.2.3 Results and Discussions

The number of crime incidents against children in different states and union territories over the period 2001 to 2020 has been collected from the National Crime Records Bureau (NCRB) website [109]. The digitization rate, literacy rate, unemployment rate, and urbanization rate over the same period have been collected from the Open Government Data (OGD) Platform in India [110].

To understand the overall rate of crime against children in different states or union territories, three statistical parameters - mean, standard deviation, and coefficient of variation (CV) have been examined. Table 2.1 shows the basic statistics of the rate of crime against children in different states and union territories in India over the period 2001 to 2020. It is observed that a higher mean rate consists of the states and union territories are Delhi (79.2) followed by A&N Islands (43.7), Chandigarh (35.8), Sikkim (32.3), and Madhya Pradesh (30.6). All of the above states or union territories consist mean rate of more than 30, which is arbitrarily chosen. In Jammu and Kashmir (3.7), Nagaland (3.7), and Jharkhand (3.3), the rate of crime is less than 5. Only considering the mean rate is not enough to visualize the actual scenario of the rate of crime against children. The Coefficient of variation (CV) along with the mean has also been studied. If a state or union territory acquires a lower CV value indicates that it maintains the rate of crime against children in the near future. In Table 2.1 we observed that only Delhi's CV value is 72.4 which is less than the mean of 79.2, which indicates the rate of crime against children will not increase acutely in the upcoming year. Except for Delhi, all other states or union territories achieved a much higher value of CV than the mean rate, thus we can conclude that the present rate of crime against children is not sustainable in the long run. Further study is required to investigate the visualization of the rate of crime against children in different states or union territories in India.

The mean rate and coefficient of variation are not enough to analyze the actual growth rate of crime against children in different states and union territories in India. Before calculating the growth rate and instability it is necessary to test the stationary of the rate of crime against children dataset. The statistical stationary test is conducted to determine whether the time series data is capable of performing experiential analysis. Our stationary test has been performed using the Augmented Dickey-Fuller Test (ADF) as mentioned earlier. Table 2.2 shows the results of the stationary test. It has been observed that after applying a stationary test over the actual dataset, the results of the t-statistic and P-value do not meet the permission level in most of the states and union territories. That infers the existence of a unit root hence it is non-stationary. After considering the first-order difference over the actual dataset and applying the stationary test, it has been observed that the value of t-statistic and P-value in almost all states except Gujarat satisfied the requirement of stationary. To calculate the growth rate and instability of the rate of crime against children in different states and union territories in India, first-order difference data have been considered.

After conducting a stationary test, Growth rate and instability have been calculated to measure the rate of crime against children in different states and union territories in India over the period 2002 to 2020. Table 2.3 shows the results of coefficient, probability (P), adjusted R-squared ($\bar{R}^2$), Growth Rate (GR), Coefficient of variation (CV), and Cuddy–Della Valle Index (I_X). It has been observed that the rate of crime against children achieves the highest growth rate in Delhi (4.0290), followed by A&N Islands (2.2039), Chandigarh (1.7959), and Sikkim (1.6230). The overall growth rate of India is calculated as 0.6622. Figure 2.2 shows the top 10 states and union territories in India which incur higher growth rates. To measure the stability of the growth rate we have used the Cuddy-Della Valle Index (I_X). A lower value in growth rate but a higher value in the Cuddy-Della Valle Index (I_X), indicates that the growth rate may not persist in the upcoming year. The growth rate of Andhra Pradesh is 0.4309 with the lowest I_X value of 29.9030, indicating that the growth rate will persist in the future. In the state of Assam, the growth rate is 0.6823 but a large I_X value indicates the growth rate does not persist in the long run. The highest growth rate of 4.0290 occurred in Union territories Delhi with moderate I_X value, indicating that the growth rate in the upcoming year has not drastically changed.

Table 2.1 Basic statistics of the rate of crime against children in different states and union territories in India over the period 2001 to 2020.

State	Mean	Std. Dev	CV%	Min	Max
Andhra Pradesh	8.7	5.1	58.9	1.2	17.5
Arunachal Pradesh	14.6	12.9	88.4	0.0	38.7
Assam	13.6	18.6	136.6	0.3	55.6
Bihar	5.7	6.0	104.7	0.3	20.8
Chhattisgarh	29.4	20.2	68.8	8.4	68.9
Goa	24.8	17.0	68.5	6.6	63.5
Gujarat	10.3	7.2	69.6	3.3	23.8
Haryana	18.5	19.2	103.9	2.1	55.2
Jharkhand	3.3	4.2	126.4	0.3	13.0
Karnataka	10.4	12.3	119.0	0.3	32.2
Kerala	17.2	16.4	95.6	1.5	50.9
Madhya Pradesh	30.6	19.7	64.6	7.2	63.3
Maharashtra	19.4	17.0	87.2	5.1	51.8
Mizoram	22.8	20.4	89.8	0.0	59.3
Nagaland	3.7	4.4	119.9	0.0	13.9
Odisha	11.7	15.8	134.9	0.3	49.9
Punjab	11.9	9.7	81.5	1.5	29.9
Sikkim	32.3	32.3	99.8	2.7	108.9
Tamil Nadu	7.6	7.5	97.8	0.3	21.0
Tripura	12.2	9.8	80.1	0.0	29.8
Uttar Pradesh	9.9	7.2	72.7	3.0	22.5
West Bengal	9.4	10.5	111.4	0.6	34.2
A&N &Islands	43.7	41.1	94.0	0.0	125.5
Chandigarh	35.8	23.1	64.7	12.6	72.0
D&N Haveli and Daman & Diu	28.2	23.3	82.6	0.0	89.0
Delhi	79.2	57.3	72.4	10.5	169.4
Jammu & Kashmir	3.7	4.3	114.4	0.3	12.7
Puducherry	8.4	5.6	67.1	1.5	19.8
India	13.3	10.9	81.8	3.0	33.2

Table 2.2 ADF test for stationary.

state	**First Difference**		**Original Data**	
	t-statistic	P-Value	t-statistic	P-Value
Andhra Pradesh	-6.927	0.000	1.065	0.995
Arunachal Pradesh	-5.974	0.000	-0.175	0.941
Assam	-3.216	0.019	-0.056	0.954
Bihar	-4.147	0.001	0.755	1.000
Chhattisgarh	-2.113	0.024	0.246	0.996
Goa	-3.174	0.022	-1.760	0.401
Gujarat	-2.301	0.093	0.063	0.999
Haryana	-2.498	0.012	0.333	0.979
Jharkhand	-6.520	0.000	-0.967	0.765
Karnataka	-2.682	0.008	0.902	1.000
Kerala	-3.626	0.005	-2.493	0.117
Madhya Pradesh	-3.835	0.003	-0.526	0.887
Maharashtra	-2.685	0.044	-2.173	0.999
Mizoram	-6.988	0.000	-1.817	0.372
Nagaland	-2.872	0.049	-1.182	0.996
Odisha	-2.708	0.084	0.491	0.985
Punjab	-3.286	0.016	-0.337	0.920
Sikkim	-2.763	0.040	0.378	0.981
Tamil Nadu	-3.653	0.005	0.969	0.994
Tripura	-4.491	0.000	-4.002	0.001
Uttar Pradesh	-4.719	0.000	-3.352	0.013
West Bengal	-2.125	0.010	-3.490	1.000
A&N Islands	-7.171	0.000	0.737	0.991
Chandigarh	-2.630	0.047	-0.701	0.847
D&N Haveli and Daman & Diu	-4.789	0.000	1.032	0.995
Delhi	-2.339	0.016	0.542	0.986
Jammu & Kashmir	-4.132	0.001	2.265	0.999
Puducherry	-7.075	0.000	-0.220	0.936
India	-0.057	0.099	2.142	0.999

Table 2.3 Growth rate and instability.

State	**Coefficient**	P	$\bar{R}^2$	GR	CV	I_X
Andhra Pradesh	0.0043	0.0000	0.7420	0.4309	58.8715	29.9030
Arunachal Pradesh	0.0073	0.0001	0.5540	0.7327	88.4379	59.0617
Assam	0.0068	0.0039	0.3290	0.6823	136.6251	111.9159
Bihar	0.0028	0.0004	0.4660	0.2804	104.6950	76.5062
Chhattisgarh	0.0146	0.0000	0.6760	1.4707	68.8138	39.1695
Goa	0.0124	0.0000	0.6770	1.2477	68.5399	38.9533
Gujarat	0.0051	0.0000	0.6710	0.5113	69.6283	39.9378
Haryana	0.0092	0.0004	0.4700	0.9242	103.9343	75.6653
Jharkhand	0.0016	0.0021	0.3680	0.1601	126.3830	100.4725
Karnataka	0.0052	0.0013	0.3990	0.5214	119.0029	92.2561
Kerala	0.0086	0.0002	0.5140	0.8637	95.6067	66.6509
Madhya Pradesh	0.0152	0.0000	0.7040	1.5316	64.5607	35.1248
Maharasht	0.0097	0.0001	0.5610	0.9747	87.2035	57.7785
Mizoram	0.0114	0.0001	0.5460	1.1465	89.7604	60.4801
Nagaland	0.0018	0.0014	0.3950	0.1802	119.8677	93.2352
Odisha	0.0059	0.0035	0.3360	0.5917	134.8964	109.9220
Punjab	0.0059	0.0000	0.5960	0.5917	81.4579	51.7754
Sikkim	0.0161	0.0002	0.4910	1.6230	99.8392	71.2295
Tamil Nadu	0.0038	0.0002	0.5020	0.3807	97.7983	69.0154
Tripura	0.0061	0.0000	0.6040	0.6119	80.1036	50.4080
Uttar Pradesh	0.0049	0.0000	0.6510	0.4912	72.6548	42.9217
West Bengal	0.0047	0.0007	0.4330	0.4711	111.3944	83.8794
A&N Islands	0.0218	0.0001	0.5220	2.2039	94.0343	65.0130
Chandigarh	0.0178	0.0000	0.7030	1.7959	64.7028	35.2616
D&N Haveli and Daman & Diu	0.0141	0.0000	0.5880	1.4200	82.5917	53.0133
Delhi	0.0395	0.0000	0.6530	4.0290	72.3581	42.6237
Jammu & Kashmir	0.0019	0.0009	0.4190	0.1902	114.3741	87.1798
Puducherry	0.0042	0.0000	0.6870	0.4209	67.1492	37.5676
India	0.0066	0.0000	0.5940	0.6622	81.8137	52.1302

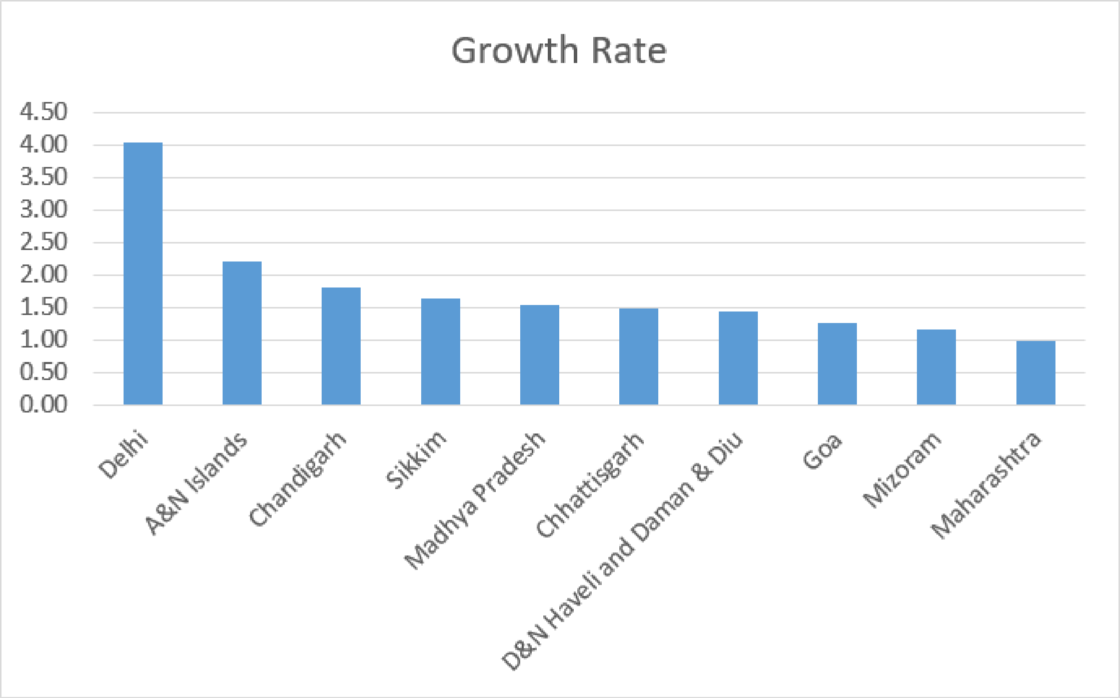


Fig. 2.2 Top ten states of India with respect to the growth rate of crime against children.

To determine different socio-economic factors that influence the rate of crime against children in different states and union territories in India, we have considered four factors namely rate of unemployment, rate of literacy, rate of digitization, and rate of urbanization. Table 2.4 shows the correlation matrix of different factors. The correlation values lie between -1 to +1. It has been observed in Table 2.4, that all values are positively correlated. The highest correlation (0.944478) is observed between urbanization and digitization which indicates that when urbanization increases digitization also increases.

Table 2.4 Correlation matrix between different socio-economic factors.

	Unemployment (X_1)	Literacy (X_2)	Digitization (X_3)	Urbanization (X_4)
Unemployment (X_1)	1.000000			
Literacy (X_2)	0.169227	1.000000		
Digitization (X_3)	0.256244	0.691257	1.000000	
Urbanization (X_4)	0.128541	0.713648	0.944478	1.000000

Table 2.5 Factors influence rate of crime against children in India.

Variable	Coefficient	Standard Error	t	Standardized Coefficient (β)
Rate of unemployment(X_1)	-3.928	1.232	-3.189	-1.317
Rate of literacy(X_2)	-0.466	0.181	-2.575	-0.082
Rate of digitization(X_3)	0.643	0.046	14.160	0.744
Rate of urbanization(X_4)	1.815	0.424	4.283	2.714
No Of Observation		**20**		
R^2		**0.986**		
$\bar{R}^2$		**0.982**		

After conducting a multi collinearity test we have applied a multiple regression model by considering the rate of crime against children (Y) as an independent variable and rate of unemployment (X_1), rate of literacy (X_2), rate of digitization (X_3), and rate of urbanization (X_4) as a dependent variable. For this estimation, we have used the Ordinary least square technique. The details of the results are shown in Table 2.5. We obtained R^2 and adjustment R^2 (i.e. $\bar{R}^2$) values as 0.986 and 0.982 respectively.

Unemployment and a crime against children: From Table 2.5, it can be observed that the rate of unemployment achieves a negative coefficient. The negative value indicates that the unemployment rate in India does not influence the rate of crime against children. A similar observation is found in the study of Beland et al. [111] in Canada. They found that unemployment is not related to crime or violence in the family, and parents who work from home or are unemployed are less involved in domestic violence.

Literacy and a crime against children: Literacy rate and the rate of crime against children were negatively correlated as found in Table 2.5. But Rahul Amin [112] has designed a mathematical model on the rate of crime and the literacy rate over different states in India and concluded that the volume of crime decreases as the literacy rate increases.

Digitization and crime against children: The rate of crime against children and the rate of digitization are positively correlated in Table 2.5. The coefficient value is 0.643 and is more than 0.5. It indicates that they are strongly correlated, i.e. the rate of crime against children increases as digitization increases. India is the fastest developing country where the growth of the internet increases day by day and subsequently, the rate of crime against children is increasing in the form of cyber bullying, cyber stalking, identity theft, online abuse, and online gaming threats. According to NCRB reports, cyber crime against children in India was reported at 164 in 2019 whereas in 2020 it was 842, an increasing rate is 413 percent. Out of 842 instances 738, nearly 87 percent depicted minors involved in the sexual act. It is an alarming call for all of us to develop strategies and measures for cyber worlds for our youths. The usage mannerisms of technology must be taught to our adolescents.

Urbanization and crime against children: The urbanization rate is also positively correlated to the rate of crime against children, which indicates that the rate of crime against children increases as the rate of urbanization increases. Malik [113] discussed different aspects by which crime increases in an urban area in India. They found various reasons, people are less integrated, cultural conflicts, loss of moral values, capital accumulation, and the gap between rich and poor people which increase the rate of crime in an urban area. A state with proper law and order can reduce the rate of crime to counter the crime.

2.3 PART - II: Predicting Geological Location of Crime

2.3.1 Past Works

Much research has been carried out to predict the occurrence of crime before its commencement. To deal with criminal activities, the prediction of crime is essential. The government across the globe provides priority for crime prediction. To identify the hotspot of crimes, different projects are carried out by different government, non-government, and academic organizations. Research has also been carried out to study the relationship between socioeconomic variables (e.g. education level, income level, unemployment) and criminal activities.

Bogomolov et al. [5] analyze demographic information and the usage of mobile network infrastructure to predict the hotspot area of criminal activities. A Decision Tree and Naive Bayesian classifier are used by Rizwan [6] on two data sets for predicting crime category. The study achieves 83.95% accuracy. Different machine learning algorithms have been applied by Shojaee et al. [18] to predict the crime category. Among these techniques, the K-Nearest Neighbor algorithm achieves the best performance with 89.50% accuracy. The feature selection process is improved by using the Chi-square method. The crime pattern is studied by Wang et al. [19] by proposing a machine learning agent known as “Series Finder”, which tries to find a crime committed by the same offender or groups of offenders. Yadav et al. [20] use the clustering method to study the patterns of criminal behaviour and geographic criminal history. The roles of social networks have also been studied to

Table 2.6 Dataset Attributes Description.

Attributes	Description
TIMESTAMP	A timestamp when the given crime occurred
ACT 379	Signifies if the crime is under act 379 i.e. robbery
ACT 13	Signifies if the crime is under act 13 i.e. gambling
ACT 279	Signifies if the crime is under act 279 i.e. accident
ACT 323	Signifies if the crime is under act 323 i.e. violence
ACT 363	Signifies if the crime is under act 363 i.e. murder
ACT 202	Signifies if the crime is under act 202 i.e. kidnapping
LATITUDE	It signifies the latitude of the location of the crime
LONGITUDE	It signifies the longitude of the location of the crime

predict criminal activities. Sentiment analysis has been carried out on Twitter data by Chen et al. [21] to predict crime in real time. The concentration of crime occurrences and large-scale hotspot locations are also identified by analyzing these Twitter data.

2.3.2 Materials and Methods

To analyze the crime pattern and to predict the location of the crime, we use the criminal records of Indore city which are available publicly on the website of the Indore city police. It provides information on crime incidents reported in Indore in the months of February and March of 2018. This data set contains 2091 crime information with different attributes as shown in Table 2.6. Before performing the analysis and prediction on crime data, a pre-processing step is performed. In this step, the Timestamp attribute is processed to obtain ten different attributes i.e. ‘year’, ‘month’, ‘day’, ‘hour’, ‘day of year’, ‘week’, ‘week of year’, ‘day of week’, ‘weekday’, and ‘quarter’. The location of the crime (i.e. latitude and longitude of the crime) is predicted from these ten different attributes and the type of crime. Linear regression [114] and support vector regression [115] is used as prediction model.

Linear regression is the simplest approach to statistical learning. In this model, the variable which is to be estimated is called the dependent variable and the other variables are called the independent variable. There are two types of Linear Regression models, namely, Single variable and Multivariable linear Regression. In this experiment, the multivariable linear regression model is used where dependent variable Y is estimated from a set of independent variables $X_1, X_2, ..., X_p$ using the equation 2.11.

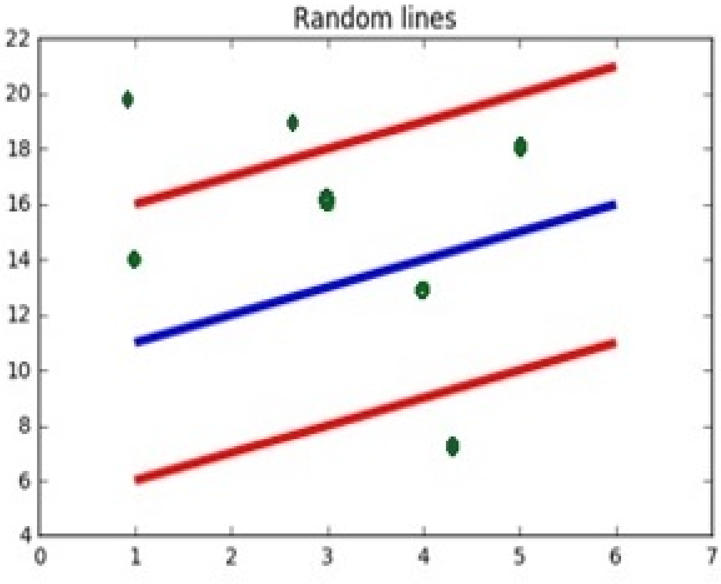


Fig. 2.3 SVR boundary with data points.

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_p X_p \tag{2.11}$$

where β_i indicates the coefficient of independent variable X_i.

The Support Vector regression (SVR) algorithm works with continuous values. In the simple regression model, the error rate is minimized. But in SVR the error is fitted within a certain threshold limit. As shown in Fig. 2.3, SVR considers the points that are within a boundary line (shown in red color line). In simple terms, SVR takes only those points which have the least error rate. Thus it gives a better-fitting model.

In these two models, the latitude and longitude of the crime are considered as dependent variables whereas ten attributes are obtained from Timestamps, and the type of crime is considered as an independent variable.

2.3.3 Results and Discussions

In this study, we analyzed the total number of crimes at different times of the day for the months of February and March of 2018. In this plot (Fig. 2.4) y-axis shows the total number of crimes and

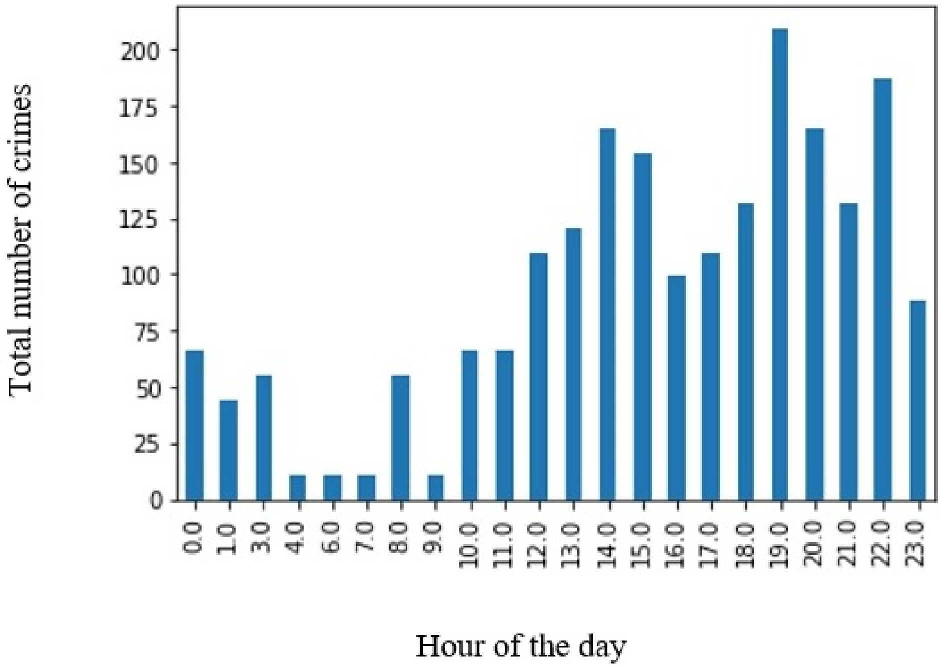


Fig. 2.4 Criminal activities occur in the different hour of the day.

the x-axis on the graph shows the hour of the day in 24-hour format. It is clear from the plot that most of the crime occurs after evening (after 6 p.m. onwards).

We also study the rate of different kinds of crimes occur in this two months. The plot is shown in Fig. 2.5. In the plot y axis represents different types of crimes and x axis shows the total number of crimes during these two months. From this plot we can conclude that the violence, accident and robbery occur with high frequency but kidnapping event occurs with low rate.

Crime Location Prediction

The linear regression and support vector regression models are used to predict the geological locations i.e. latitude and longitude of the crimes. The locations of the crime in a particular month of a year at a specific time are computed for different types of crime using linear regression and support vector regression. Figure 2.6 shows the predicted locations for different types of crime for the month of March, 2018 at 22 hours using linear regression as a representative. The same investigation is also done using support vector regression and shown in Fig. 2.7.

To measure the accuracy of the prediction, the data set is divided into two portions: training data set and testing data set. The training data set contains all features along with the target label.

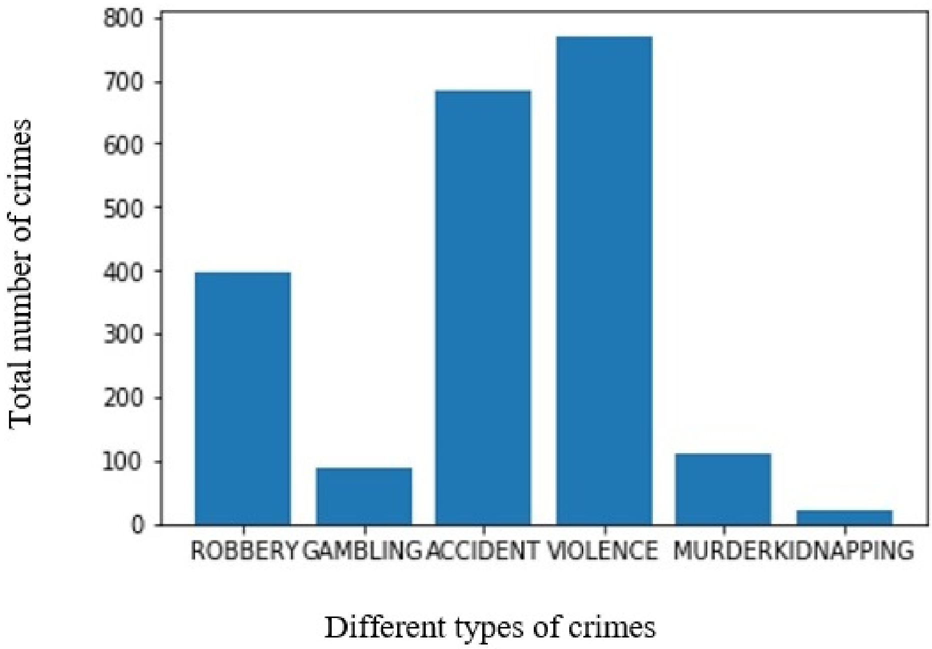


Fig. 2.5 Rate of different types of criminal activities.

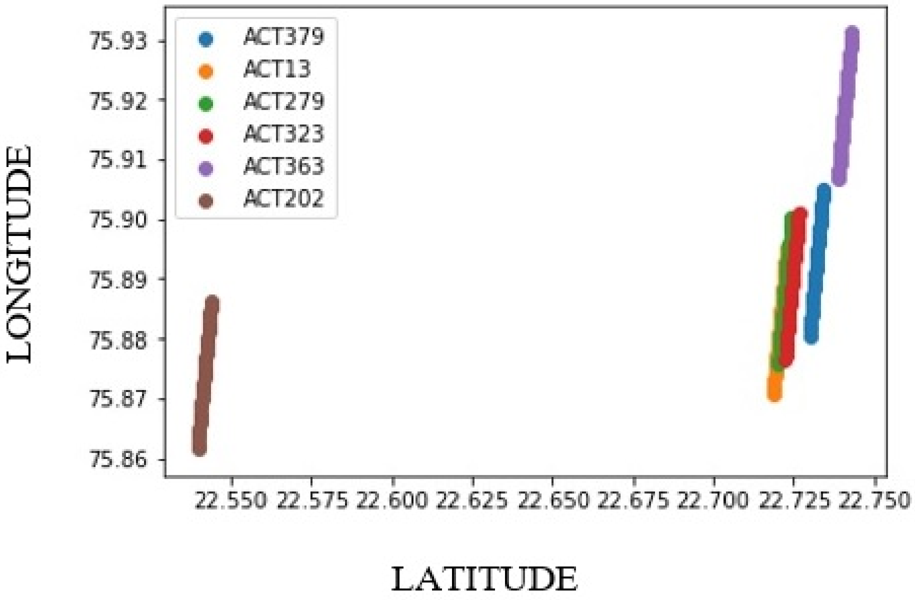


Fig. 2.6 Crime area prediction by Linear Regression.

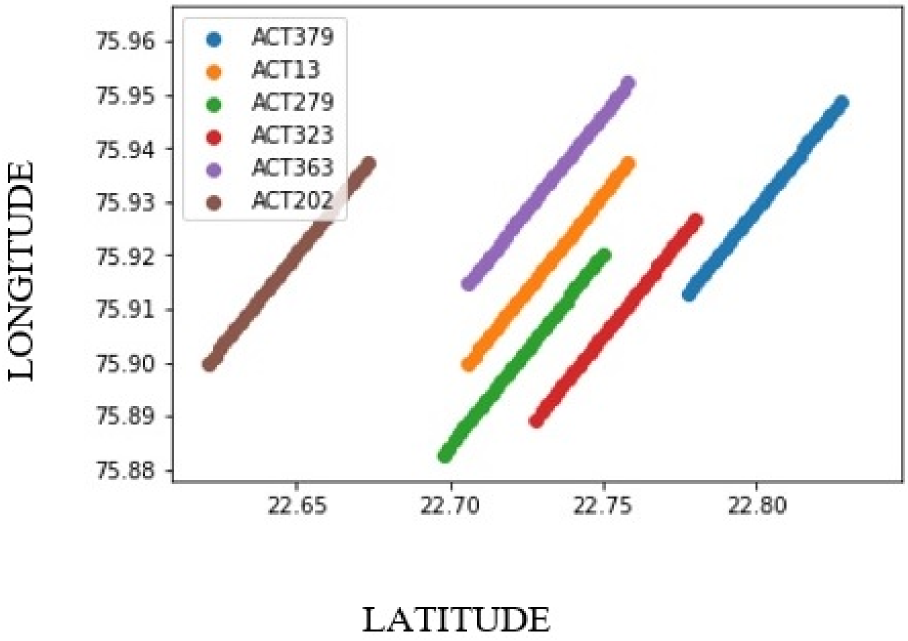


Fig. 2.7 Crime area prediction by Support Vector Regression.

Table 2.7 The error estimation on latitude and longitude.

Model	Latitude		Longitude	
	RMS Error	MAE Error	RMS Error	MAE Error
Linear Regression	0.0024	0.0493	0.0367	0.1917
Support Vector Regression	0.0007	0.0274	0.0004	0.0207

However, the testing data set contains only the features. A machine learning model is used to predict the target label. The test data set contains 20% of the original data set and the rest is used for training purposes. In the testing phase, the location for a particular crime is predicted using linear regression and support vector regression and compared with the original information. Figure 2.8 shows the predicted locations using linear regression and actual locations for robbery (ACT 379) as an example. Similarly, Fig. 2.9 shows the same information using support vector regression.

Error estimation in the predicted location of crime (i.e. Latitude and Longitude) is done for the testing data set for these two models. The Root Mean Square Error (RMS) and Mean Absolute Error (MAE) are calculated using predicted and original values on latitude and longitude. The error estimation is shown in Table 2.7. It can be observed from the table, that support vector regression better estimates the crime location over the linear regression model.

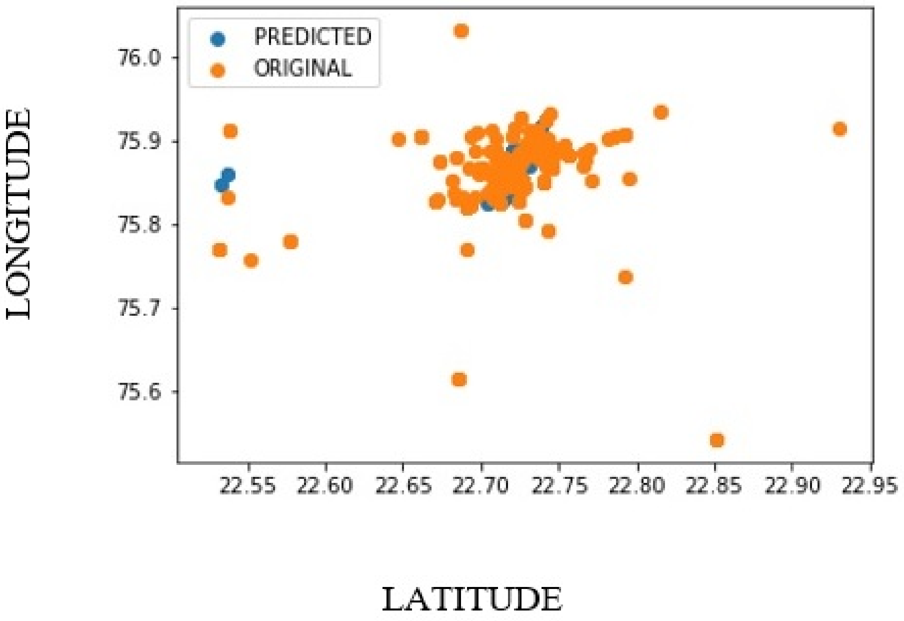


Fig. 2.8 Original and Predicted location for robbery (ACT 379) using Linear Regression.

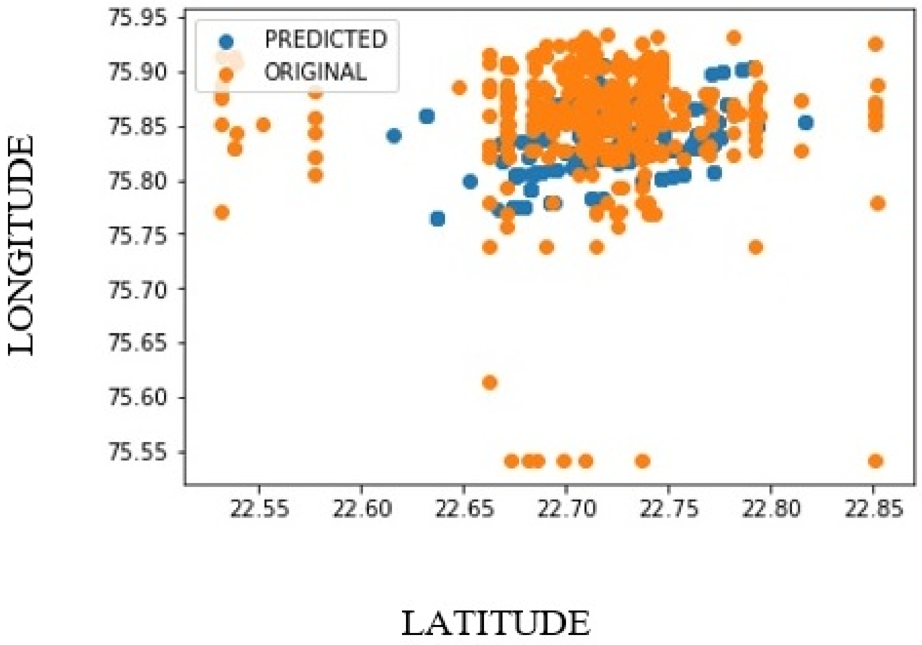


Fig. 2.9 . Original and Predicted location for robbery (ACT 379) using Support Vector Regression.

2.4 Summary

In the first part of this chapter, we have focused on the rate of crime against children in different states and union territories in India over the period 2001 and 2020. The main hypothesis of this study is based on statistical analysis. Initially, we collected the rate of crime against data from NCRB, after that, we examined it using different statistical measures like mean, standard deviation, and coefficient of variance. We have found that the union territories Delhi have the highest crime rate 79.2 followed by A&N Islands (43.7), Chandigarh (35.8), Sikkim (32.3), and Madhya Pradesh (30.6). We have also measured the mean along with the coefficient of variation. Since the data set consists of time series data, the mean, and coefficient of variations are not capable of determining the overall rate of crime against children in India. To analyze time series data, the stationary test is required to be performed. Thus, we have performed an ADF stationary test over the rate of crime data in different states and union territories in India. We observed that first-order differences support the permission level of stationary. Using first-order difference data we have calculated the growth rate of different states and union territories of India. Delhi has achieved the highest growth rate 4.029 followed by A&N Islands (2.2039), Chandigarh (2.2039), Sikkim (1.623), and Madhya Pradesh (1.5316), and the overall growth rate throughout India is 0.6622. We have analyzed the stability of the growth rate by using the Cuddy–Della Valle Index (I_x). The unemployment rate, literacy rate, digitization rate, and urbanization rate are analyzed to determine their influence on the rate of crime against children. We have conducted a multi collinearity test over these four factors using the ordinary least square technique. The unemployment rate and literacy rate have a negative impact whereas the digitization rate and urbanization rate directly influence the rate of crime against children.

In the second part of this chapter, an automated methodology has been proposed to predict the location of the crime to reduce the occurrence of crime. Machine learning algorithms have been used to predict the location of crime in advance, from past crime records. Data mining and machine learning have become a vital part of crime detection and prevention nowadays. The effectiveness and accuracy of two machine learning algorithms, namely, linear regression and support vector machines are evaluated using a data set containing the crime records of Indore city. From the experimental results, it can be concluded that support vector regression works well and has more accuracy if the data points lie within the clustering region as it has a low RMS error value. However, the Linear Regression works well if the points are located near the boundaries.

Chapter 3

Spam SMS Classification

3.1 Introduction

Short Message Service (SMS) is one of the most popular forms of telecommunication service around the world because of its affordability. According to [116], around 5 billion people can send and receive SMS, and more than 200 thousand SMS are sent every second. 83% of SMS messages are read within 90 seconds. On average, SMS messages have a 98% open rate compared to 20% of emails. Text messages have a 209% higher response rate than emails. Because of these statistics, marketers prefer SMS as the primary form of advertising which leads to spamming. SMS spamming has become a serious problem for mobile subscribers. Spam SMS refers to unwanted text messages that are usually either someone promoting a product or service or someone attempting to scam a subscriber into providing personal information. It incurs substantial costs in terms of lost productivity, network bandwidth usage, management, and raid of personal privacy. The main reason to stop spamming is that it costs a lot more to its receivers than its senders. Because of these reasons and many others, SMS spam detection is very necessary.

SMS spam is a kind of problem that does not have an algorithmic definite solution. Existing SMS spam filtering methods are not very robust. The machine learning method comes to be the most popular choice for spam classification; several researchers have utilized supervised machine learning methods for comparative results of spam classification. Plenty of research has been carried out in this direction making use of machine learning techniques such as Naïve Bayes, Random Forest, Support Vector Machine, and Decision Trees etc. Using the traditional machine learning methods mentioned above, feature engineering is a time-consuming process with an extra computational expense. It is also difficult to extract all the information from the short length of the text.

Among the recent solutions that have proven to be effective in solving these kinds of problems is the use of deep neural networks. Deep neural network-based architecture, such as Convolution Neural Networks (CNN), Recurrent Neural Networks (RNN), and Long Short-Term Memory (LSTM) has been used for many classification problems for images, videos, and texts. Deep learning models are empowered with automatic pattern recognition as well as traditional clustering machine learning techniques (unsupervised techniques).

It has been seen from the existing work, that deep learning models achieve better accuracy than traditional machine learning models. The researchers use the complex architecture of deep learning models to increase the performance of the model. In this chapter, deep learning models with simpler architectures have been tried in search of better accuracy by using word embedding techniques. The proposed method classifies spam and non-spam (ham) messages using two deep learning models, namely Convolution Neural Network (CNN) and a Convolution Neural Network with Long Short-Term Memory (CNN-LSTM). These two models are embedded with two indexed vocabularies – BUNOW [117] and GloVe [118] as input vectors. The models are tested with Tiago's data set [119] and the accuracy of these models is compared with existing methods. The performances of the models are also evaluated on the Youtube spam collection data set [120] to judge the versatility of the proposed method.

3.2 Past Works

Machine learning techniques have been used extensively for SMS spam detection. Some researchers use traditional machine learning approaches while others use deep learning models for this purpose. Among these, which use Tiago's data set for testing and performance evaluation has been discussed here.

3.2.1 Traditional Machine Learning Approaches

Almeida et al. [22], the creators of Tiago's data set, tested several machine-learning algorithms and laid the groundwork for further research. In this work, two tokenizers have been used, one for preserving domain names, and mail addresses and another one to preserve symbols that can have a crucial part in classification. SVM, Boosted NB, Boosted C4.5 and PART achieved the best accuracies. SVM got the highest accuracy of 97.64%. Three machine learning algorithms, viz., Naïve Bayes, Random Forest, and Logistic Regression are used by Sethi et al. [23]. Frequencies of the words are used as the input vector. Naïve Bayes achieved the best accuracy of 98.445% while Random

Forest and Logistic Regression got 97.009% and 94.312% accuracy respectively. Navaney et al. [24] compared the results of Naïve Bayes, SVM, and Maximum Entropy algorithm for SMS spam detection. A document term matrix is used as an input vector. In this research, Naïve Bayes and the Maximum Entropy Algorithm achieved accuracies of 94.55% and 91.95% respectively while SVM achieved the maximum accuracy of 97.4%. Alzahrani et al. [25] compared the performance of spam detection using a neural network model, Gaussian NB, Logistic Regression, and SVM. The neural network had three dense layers and one output layer with 137,153 total parameters. After training it for 30 epochs, the neural network model achieved the best accuracy of 97.67% among these four models. Gaussian NB got the worst accuracy of 88.16% while Logistic Regression and SVM both achieved an accuracy of 94.26%. Xia et al. [26] used a hidden Markov model to detect spam SMS where every word is processed directly. They extended their research [27] by assigning weight for each word that indicates the probability of the message being detected as spam or ham SMS. The experimental results show that the weighted feature gives better accuracy. The weighted feature model achieves 96.9% accuracy. Initially, Diallo et al. [28] tried to find a similarity matrix in the multi-view document clustering algorithm and further enhanced their proposed model [29] by considering Cosine similarity, Euclidean distance, and magnitude difference. Their proposed model gives better performance for small dimensional but for larger dimensions, it consumes more space and time.

3.2.2 Deep Learning Models

In the deep neural category, Taheri et al. [30] implemented an RNN-based classification model on Tiago's data set. The data set is divided into training and testing purposes which includes 70% and 30% respectively. The size of the RNN was 100 and each word in a training vector got a size of 50. The batch size for each epoch and word sequence was 25 for both. After training the model for 200 epochs the best accuracy achieved was 98.11%. The semantic LSTM model has been tried for spam detection by Jain et al. [31]. For vectorization of the vocabulary, the missing words are searched from WordNet and ConceptNet, instead of passing each word through Google Word2Vec. If the missing word is not found on WordNet and ConceptNet, a random value is assigned to that word. The LSTM model consisted of 100 units, the dropout rate was 0.1 and the Sigmoid activation function was used. The model was trained with 10 epochs. One interesting observation from this research was that as the feature count went up from 5000 to 6000, the accuracy increased but increasing the features even more resulted in lesser accuracy. The best accuracy, 98.92% was achieved when the number of features was 6000. Popovac et al. [32] implemented a CNN model for spam detection. For pre-processing, several steps had been carried out like lowercasing, tokenizing, stop word removing, and finally converting the messages into a matrix of TF-IDF features. The CNN model was composed of two convolution layers with a filter size of 32, and a kernel size of 3 with a ReLU activation function. After that Max pooling of pool size 2 were added. After a flattened layer, a fully connected dense layer of 128 units with ReLU activation was used. Finally, the output layer consisted of one unit with a sigmoid activation function. Due to the nature of the problem Adam optimizer and binary cross-entropy loss were used. After training the model for 10 epochs an accuracy of 98.4% was achieved. CNN and RNN models are tried and their performances are compared by Annareddy et al. [33]. The list_to_sequence( ) function of the Keras library had been used to assign a unique integer value to every word in the vocabulary for the input vector. For both models, the ReLU activation function was used and the dropout was initialized at 0.2. After training two models for 10 epochs, the accuracy of the testing data for the CNN model was 96.4% with a loss of 0.090 while the RNN model achieved an accuracy of 97.8% with a loss of 0.143. Roy et al. [34] implemented several deep learning models on Tiago's dataset. The best accuracy was achieved on a 3-channel CNN with different dropout values for each CNN. After doing a 10-fold cross-validation on the multi-channel CNN, 99.44% accuracy was achieved. Chandra et al. [35] proposed an RNN-LSTM model for classification. As preprocessing steps stop words were removed from the corpus and the words were converted into a TF-IDF vector. The data set was divided into 70% training and 30% testing data sets. RMSprop optimizer, ReLU, and Sigmoid activation function are used in the model and 98% accuracy was achieved after 8 epochs. On the same TF-IDF vector, Naïve Bayes and SVM models had been applied and achieved accuracies of 80.54% and 97.81% respectively. Konti et al. [36] use the different versions of BERT models and analyze the perfor-mance of these models for detecting SMS spam. The highest accuracy (95.02%) is achieved by the DistilBERT model. Abayomi-Alli et al. [37] used different machine learning algorithms like Naïve Bayes, BayesNet, Self Organizing Maps, decision tree, and a deep learning model BiLSTM to detect spam SMS. The experiment achieves 98.6% highest accuracy using the BiLSTM model. Diallo et al. [38] introduce a deep learning based document clustering technique that can work better than the existing clustering algorithms. Their proposed model can learn document representation in a better manner and they proved it by considering images and text data sets. Shaaban et al. [40] proposed a deep ensemble approach for spam detection. For feature extraction, they used convolution and pooling layers, and their base classifier consists of random forests and extremely randomized trees. Their model achieves 98.38% accuracy. Ghourabi et al. [39] proposed a hybrid CNN-LSTM model for the classification of spam messages. They merge the two data sets collected from the UCI repository and the Arabic messages collected from a different source. Their proposed model achieved 98.3% accuracy.

Although all these mentioned works achieved great accuracy using existing techniques, it is clearly visible that deep learning models achieved better accuracies than traditional machine learning models. Some of these deep learning models have a very complex architecture. In this work, deep learning models with simpler architectures have been tried in search of better accuracy. For this purpose, BUNOW and GloVe word embedding techniques are incorporated with deep learning models. BUNOW and GloVe are the popular choices in sentiment analysis [121, 122], but in this work, these two word embedding techniques are tried in the context of text classification to improve accuracy.

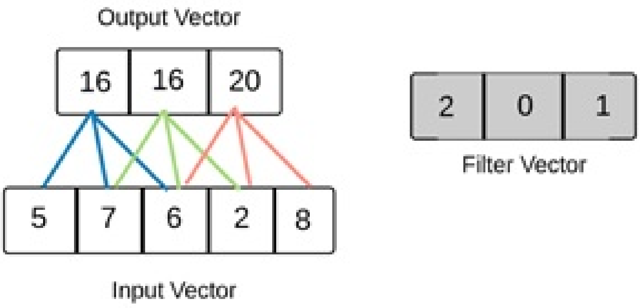


Fig. 3.1 1-Dimensional Convolution.

3.3 Deep Learning Background

3.3.1 1-Dimensional Convolution Layer

CNN has been a crucial component in artificial neural networks for solving complex computer vision problems. The main component of a CNN is the convolution operation. For most tasks, a 2-dimensional convolution is conducted as most of the CNNs are applied on 2D images. A 1-dimensional convolution works on a similar principle with one major difference. Instead of a matrix of tunable parameters scanning an image from top left to bottom right, it has a vector of tunable parameters that scans the input from top to bottom.

For example, in Fig 3.1, a 1D convolution is shown on a 5 × 1 input vector, that is being altered by a 3 × 1 filter vector. Moving the filter vector by unit stride, it produces a 3 × 1 output vector.

Formally, given a weight vector $\overrightarrow{W}$, of dimensions r × 1, acting on the input vector $\overrightarrow{T}$, a convolution with output z is defined in Eq. 3.1.

$$z = \overrightarrow{W} * \overrightarrow{T}[p : p + r] \tag{3.1}$$

Where p is the current position of the weight vector as it strides along the input to produce a vector of output values. In Eq. 3.1 the * operation is the element wise multiplication of the weight vector with the input followed by the obtained values summation [123].

A convolution layer creates a convolution filter that is convoluted with a single spatial dimension to produce a tensor of outputs.

3.3.2 LSTM

Recurrent Neural Network (RNN) is well known for critical thinking of time management. But, in practice, RNN has two problems – vanishing gradient and exploding gradient. Thus, RNN fails to deal with long-term dependencies in natural language processing.

Long Short-Term Memory (LSTM) is capable of remembering information in its memory cell for a long period of time. There are four main elements in an LSTM architecture: an input gate, a forget gate, an output gate and the last one is a self-recurrent connection along with neurons respectively. Together they are called the memory cell or four interacting layers [124]. The architecture of an LSTM cell is clarified with the equations below: Mathematically, suppose there are h hidden units and the number of input units is d.The *input gate layer* decides which value will update using Eq. 3.2,

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i) \tag{3.2}$$

where, $i_t \in R^h$ is the input gate's activation vector, matrices W_{xi} and W_{hi} contain the weights of the input and recurrent connections respectively. $x_t \in R^d$ is the input vector of the LSTM unit and $h_t \in R^h$ is the hidden state vector also known as the output vector of the LSTM unit, b_i is the bias for input gate and σ is the sigmoid function.

A *tanh layer* creates a vector of the new input value using Eq. 3.3,

$$c\_in_t = \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_{c\_in}) \tag{3.3}$$

where, $c\_in_t \in R^h$ is the cell input activation vector, matrices W_{xc} and W_{hc} contain, respectively, the weights of the input and recurrent connections. $x_t \in R^d$ is the input vector of the LSTM unit and $h_t \in R^h$ is the hidden state vector also known as the output vector of the LSTM unit, b_{c_in} is bias and tanh is the hyperbolic tangent function

The *forget gate layer* decides which values to keep by producing a value between 0 and 1, The equation of forget gate is given by Eq. 3.4,

$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + b_f) \tag{3.4}$$

where, $f_t \in R^h$ is the forget gate's activation vector, matrices W_{xf} and W_{hf} contain, respectively, the weights of the input and recurrent connections. $x_t \in R^d$ is the input vector of the LSTM unit and $h_t \in R^h$ is the hidden state vector also known as the output vector of the LSTM unit, b_f is the bias for input gate and σ is the sigmoid function.

$$c_t = f_t * c_{t-1} + i_t * c\_in_t \tag{3.5}$$

Where, c_t is the cell state vector in Eq. 3.5 and * denotes the element-wise product.

Finally, the *output gate layer* walks through the *tanh layer*, and decides the output, the equation of *output gate layer* is given in Eq. 3.6

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + b_o) \tag{3.6}$$

where, $o_t \in R^h$ is the output gate's activation vector, matrices W_{xo} and W_{ho} contain, respectively, the weights of the input and recurrent connections. $x_t \in R^d$ is the input vector of the LSTM unit and $h_t \in R^h$ is the hidden state vector also known as the output vector of the LSTM unit, b_o is the bias for input gate and σ is the sigmoid function

$$h_t = o_t * \tanh(c_t) \tag{3.7}$$

Where, h_t is the hidden state vector also known as the output vector in Eq. 3.7. “*" denotes the element-wise product and tanh is the hyperbolic tangent function.

Figure 3.2 displays the architecture of LSTM.

3.3.3 Pooling Layer

The pooling layer is generally added after a convolution layer. The purpose of the pooling layer is to achieve spatial invariance by reducing the resolution of the feature maps. Each pooled feature map corresponds to the input from a small n X n patch of units. The size of the pooling window can be arbitrary and the windows can be overlapping. The max pooling function in Eq. 3.8,

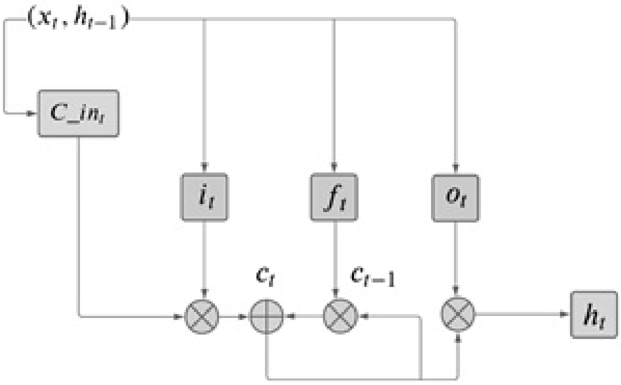


Fig. 3.2 LSTM Model Architecture.

$$a_j = \max_{NXN}(a_i^{nXn} u(n,n)) \tag{3.8}$$

applies a window function u(x,y) to the input patch and computes the maximum in the neighborhood. The result of a max pooling function is a lower resolution feature map [125].

3.3.4 Dropout Layer

Deep neural networks with a large number of parameters are very powerful machine learning systems. However, with limited training data, many of these complicated relationships in deep neural networks will be the result of sampling noise, which means they will exist in the training set but not in the testing set even if it is drawn from the same distribution. This leads to overfitting.

The model combination is one of the best methods that nearly always improves the machine learning model's performance. However, training different large model architecture and finding the optimal hyper-parameters is hard because, it needs a lot of training data and requires a lot of computation.

Dropout is a technique that addresses both of these issues. Dropout means temporarily removing a neural network's unit (hidden and visible) from the network along with all of its incoming and outgoing connections. The choice of picking the dropping units is random [126].

3.3.5 Fully Connected Layer

A fully connected layer is a function from R_m to R_n. Each output dimension depends on each input dimension. Mathematically, a fully connected layer is defined as follows:

Let x∈ R_m represent the input to a fully connected layer. Let $y_i \in$ R be the $i^t h$ output from the fully connected layer. Then $y_i \in$ R is computed using Eq. 3.9:

$$y_i = \sigma(w_{1\times1} + ... + w_{m\times m}) \tag{3.9}$$

where, σ is a nonlinear function and w_i are the learnable parameters in the network. The full output of y is calculated using Eq. 3.10

$$y = \sigma(w_{1,11} ... + w_{n,mm}) | \sigma(w_{n,11} + ... + w_{n,mm}) \tag{3.10}$$

3.3.6 Swish and Sigmoid Activation Function

The choice of activation function in deep networks has a significant effect on models' performance. Rectified Linear Unit (ReLU) is the most successful and widely used activation function. A ReLU function is defined in Eq. 3.11

$$f(x) = max(x,0)\{ \begin{array}{ll} x_i, & \text{if } x_i \geq 0 \\ 0, & \text{if } x_i < 0 \end{array} \tag{3.11}$$

Swish is a fairly new activation function, developed by researchers at Google, that constantly matches or outperforms ReLU. Swish is a smooth, non-monotonic function defined using Eq. 3.12

$$f(x) = x * \sigma(x) \tag{3.12}$$

and the sigmoid function $\sigma(x)$ is defined as Eq. 3.13

$$\sigma(x) = (1 + \exp(-x))^{(-1)} \tag{3.13}$$

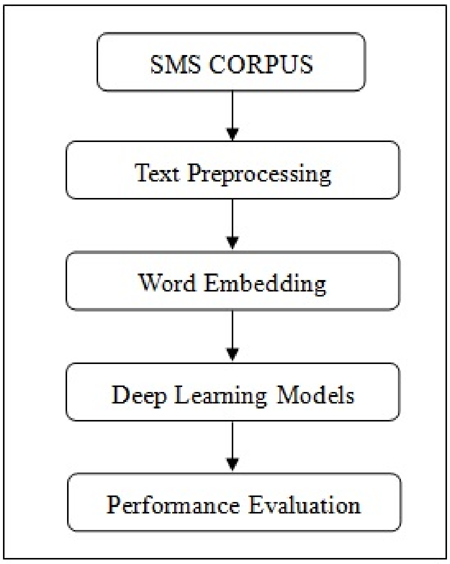


Fig. 3.3 Proposed methodology for SMS spam classification.

3.4 Materials and Methods

The proposed methodology for SMS spam classification is given in Fig. 3.3. Initially, pre-processing is done on the messages in the SMS corpus. After pre-processing, two different word embedding techniques are applied to extract the context of each word in the messages. This context information is used in the training phase of two deep learning models. Finally, testing is applied to evaluate the performance of the proposed model.

3.4.1 SMS Corpus

The dataset used in this research is known as Tiago's data set [119]. It consists of 5,574 English, real, and non-encoded messages labeled as ham (non-spam) or spam. The data set is a collection of 4 subsets:

- A subset extracted from the Grumble text Web site. It contains 425 spam SMS messages.
- A collection of 3,375 SMS non-spam messages was provided by NUS SMS Corpus (NSC) and collected for research at the Department of Computer Science at the National University of Singapore.
- SMS Spam Corpus v.01 Big subsets. It consists of 322 spam and 1,002 ham messages.

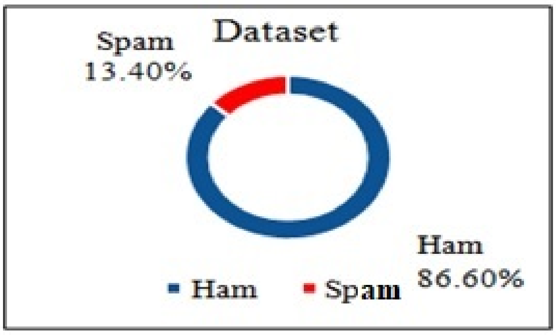
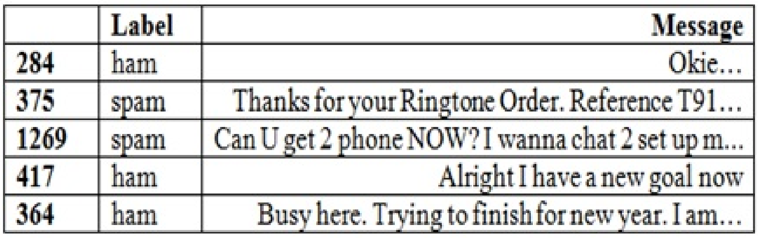


Fig. 3.4 Distribution of Spam and Ham messages in the data set.



	Label	Message
284	ham	Okie...
375	spam	Thanks for your Ringtone Order. Reference T91...
1269	spam	Can U get 2 phone NOW? I wanna chat 2 set up m...
417	ham	Alright I have a new goal now
364	ham	Busy here. Trying to finish for new year. I am...

Fig. 3.5 A Snapshot of the data set.

- A collection of 450 SMS ham messages from Caroline Tag's Ph.D. Thesis.

The data set contains 86.60% ham messages and the remaining 13.40% spam messages as shown in Fig. 3.4. A snapshot of the data set is shown in Fig. 3.5.

3.4.2 Pre-processing

One of the essential steps for creating a good machine-learning model for classification is to pre-process the text data [127, 128]. Pre-processing converts the raw data to something understandable by an algorithm. Pre-processing is a complex process that requires a lot of steps. In the proposed model, the following steps are performed in pre-processing.

In the first stage, every word in the data set is converted into lowercase. Generally, text messages often have a variety of capitalizations, which have no big impact on the final model. So, every word is reduced to lowercase for simplicity. In the next step, tokenization was applied. A word tokenizer splits each lowercase text message into a set of words. For example, the tokenizer breaks the message "alright i have a new goal now" into a set of seven words 'alright', 'i', 'have', 'a', 'new', 'goal', 'now'. Thus it produces a data frame of individual words for further processing. After tokenization, all the

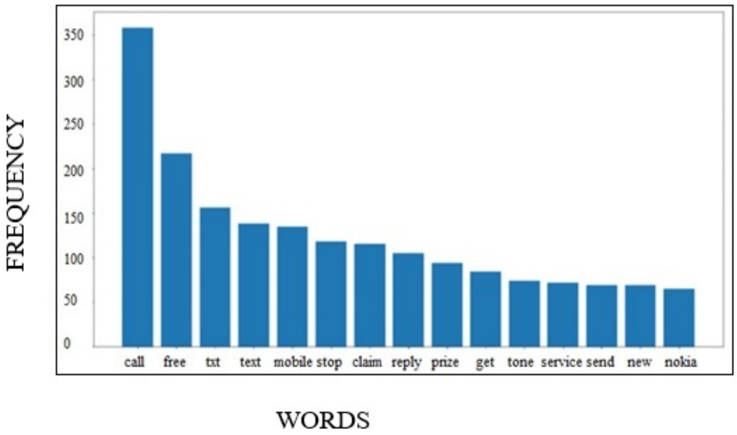


Fig. 3.6 Most Frequent Words in Spam messages with their frequencies

words with lengths less than two are removed. The alphanumeric words i.e. words with the mixing of letters and digits are also removed from the data set as numbers, symbols and punctuations did not contribute much to the learning. In the next step, stop words were removed. Stop words are frequently used common words like ‘or’, ‘and’, ‘this’, ‘that’ etc. These stop words do not contribute to the knowledge source and can be removed from the textual data. The final step of preprocessing was lemmatization. Lemmatization is the process of reducing a word to its base form by removing inflectional endings only, which is also known as a lemma. It was done to get the simplest form of each word. For example, a lemmatization reduces the words car, cars, car’s, cars’ to the car. Lemmatizer uses vocabulary and morphological analysis of words and returns the base or dictionary form of a word.

The most frequent words in the spam messages in the data set with their frequencies are shown in Fig. 3.6.

3.4.3 Word Embedding

The context of a word in a document, the words with similar semantics, and the relation of words in a text can be captured using word embedding. In word embedding, words that have the same meaning have similar representation. Each word is represented as real valued vectors in a predefined vector space. In this work, two types of word embedding techniques have been used.

The first one is the Binary Unique Number of Word (BUNOW) method [117]. In this technique, each distinct word (W_i) in the training corpus (T) is assigned a unique integer ID (ID_{Wi}). For this purpose, a vocabulary of distinct words is created from the training data set. After that, a unique serial integer ID has been assigned for each distinct word. The word (W_i) is represented by a fixed dimensional vector of size k where (2^k=number of distinct words in Vocabulary) by converting the assigned unique integer ID (ID_{Wi}) into its binary equivalent(B_{Wi}). Maximum message length (LW_{max}) is set with the number of words in the longest message in the training data set. The input feature vector (IV) of each message (M) is created by concatenating the binary vector of each word that exists in that message as given in Eq. 3.14.

$$IV_M = B_{W1} + B_{W2} + B_{W3} + B_{W4} + ... + B_{LWmax} \tag{3.14}$$

Where (+) is the concatenation operator. Any message with less than LW_{max} words is padded with all-zero vectors.

The second one is Stanford's Global Vectors (GloVe) word embedding technique [118]. GloVe word embedding considers the global context of the word rather than considering only local meaning. It is a pre-trained word embedding model that combines the advantages of two major model families: global matrix factorization and local context window methods. The model was trained on five corpora of varying sizes: 2010 Wikipedia dump with 1 billion tokens; 2014 Wikipedia dump with 1.6 billion tokens; Gigaword5 which has 4.3 billion tokens; the combination Gigaword5 and Wikipedia2014, which has 6 billion tokens; and 42 billion tokens of web data, from Common Crawl. Each corpus was tokenized and lowercased. A co-occurrence matrix X was constructed with a vocabulary of 400,000 most frequent words. In the construction of X, a context window was initialized to distinguish the left context from the right context, and a decreasing weighting function was used in the context window so that word pairs that are **d** words apart contribute **1/d** to the total count. This way, much distant word pairs contributed less to the relevant information about word relationships. The cost function for the experiment is described in the Eq. 3.15,

$$J = \sum_{i=1}^{V} \sum_{j=1}^{V} f(X_{ij})(\widetilde{W}_i^T \widetilde{W}_j + b_i + \tilde{b}_j - logX_{ij})^2 \tag{3.15}$$

where V is the size of the vocabulary, X_{ij} is the co-occurrence matrix. W_i represents a particular word in the message and W_j represents the set of context words of W_i.$\widetilde{W}_i$ and$\widetilde{W}_j$ are the vector representation of W_i and W_j respectively. b_i,$\tilde{b}_j$ are biases and f(X_{ij}) is a weighting function that should follow the following properties,

- f(0) = 0. If f is viewed as a continuous function, it should vanish as x $\rightarrow$ 0 fast enough that the $\lim_{x \rightarrow 0} f(x) log^2 x$ is finite.

- f(x) should be non-decreasing so that rare co-occurrences are not overweighted.
- f(x) should be relatively small for large values of x, so that frequent co-occurrences are not over weighted.

Although a large number of functions satisfy these properties, one class of functions are found to work well can be parameterized as,

$$f(x) = \{ \begin{array}{ll} (\frac{x}{x_{max}})^{\alpha} & \text{if } x < x_{max} \\ 0 & \text{otherwise} \end{array} \tag{3.16}$$

The performance of the model depends weakly on the cutoff, which was fixed to x_{max} = 100 for all experiments.

3.4.4 Deep Learning Models

Two deep learning models namely Convolution Neural Network (CNN) [129] and Convolution Neural Network – Long Short Term Memory (CNN-LSTM) [129] are used for the classification of messages. In this proposed method, CNN model consists of two 1-dimensional convolution layers with filter size 32, kernel size 3 and swish activation function. Between these two convolution layers, a Max Pooling layer of pool size 3, and Dropout layer with a rate of 0.2 are used. A Global Max Pooling layer and another Dropout layer with a rate of 0.2 followed the second convolution layer. A fully connected dense layer of 128 units with swish activation function and another Dropout layer with a rate of 0.2 preceded the output layer. The output layer consists of two units with sigmoid function. Weights of CNN network are randomly initialized. The model is shown in Fig. 3.7

Due to imbalanced class distribution, cost-sensitive classification is executed and class weights are assigned accordingly, 0 for ham messages and 1 for spam messages. Hyper parameters were assigned based on tests of multiple models with different values. Due to its computational and space efficiency, Adam optimizer was used with a learning rate of 0.001 along with binary cross entropy loss for compiling the model. Models were trained through 20 epochs while one of them having BUNOW embedding and the other one having GloVe embedding.

The CNN-LSTM model has exactly the same architecture as the CNN model, except the fully connected layer was preceded by a LSTM layer with 256 units with recurrent dropout rate of 0.3, and the Global Max Pooling layer was replaced with a Max Pooling layer of pool size 3. The proposed model is shown in Fig. 3.8.

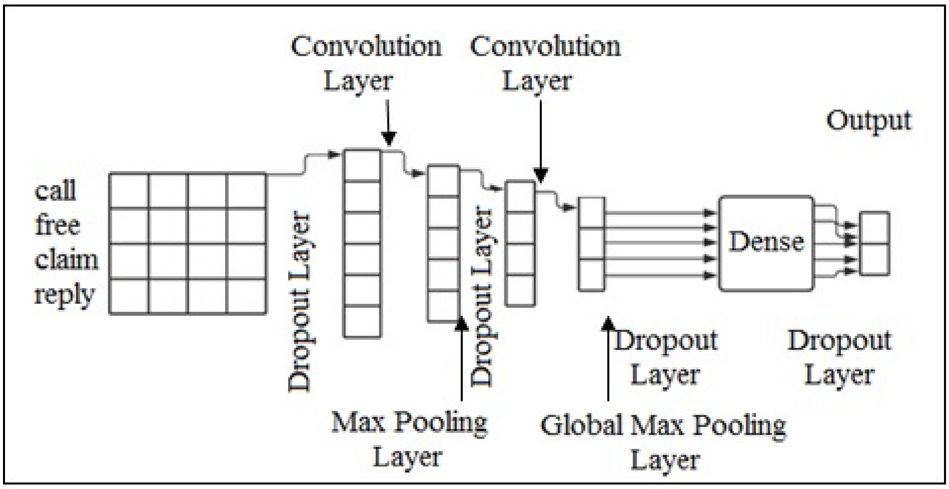


Fig. 3.7 Proposed CNN Architecture.

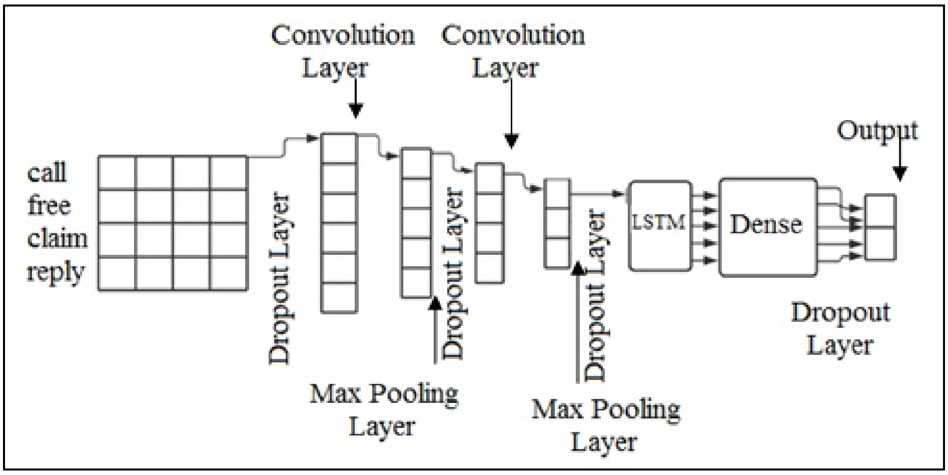


Fig. 3.8 Proposed CNN LSTM Architecture.

3.4.5 Working Principle

Tiago's data set [119] consists of 5,574 English, real and non-encoded messages labeled as ham (non-spam) or spam. The assigned labels of the messages are represented by numeric values 0 and 1 for indicating ham and spam messages respectively. These messages are pre-processed by the steps described in section 3.4.2.

Pre-processed messages are split into two parts for training and testing purposes. In this experiment, different training testing ratios have been used. The distinct words (W_d) within the messages in the training set are identified. In the case of BUNOW word embedding, each of these distinct words is represented by a unique vector by the method described in Section 3.4.3. On the other hand, in the case of GloVe word embedding, each distinct word is represented by a vector following the method described in Section 3.4.3. Next, the longest message from the training set is identified and the number of words in that message (LW_{max}) is calculated. Each message in the training set is represented by a fixed length vector of size LW_{max}. The vector representation of the message is created by concatenating the vector representation of the words that constitute the message. If the number of words in the message is less than LW_{max}, the remaining components in the fixed length vector are set to zero. After representing the messages in the training set by fixed length vector, two different deep learning models, namely the CNN model and CNN LSTM model are trained. One dimension convolution neural network has been applied in this experiment since each word is represented as one dimensional input vector. The dimension of the input layer in both the models is equal to the number of distinct words (Wd) and the output dimension is set to 300. A polling layer has been applied to decrease the dimension of the features without losing essential features. Finally, a dropout layer has been applied to prevent over fitting. The detailed structures of the models have been described in section 3.4.4.

For detecting a message as spam or ham, initially, the message is represented as fix length vector of size LW_{max} by a similar approach described above. If the length of the message is more than LW_{max}, the first LW_{max} words are considered. If the message consists of new words which are not present in the vocabulary of the training data set, the word is eliminated from the message before representing it in the vector. The vector representation is fed into the trained model to classify the message as spam or ham.

3.4.6 Performance Evaluation

To evaluate the accuracies of the models, several performance measures exist in the literature. True positive, false positive, true negative and false negative are calculated to build the confusion matrix. Accuracy, precision, recall, specificity and f1 score are used to evaluate the performance of each model.

- *True positive (TP):* It measures the number of spam messages classified as spam messages in the entire SMS corpus.
- *False positive (FP):* It measures the number of ham messages classified as spam in the entire SMS corpus.
- *False negative (FN):* It measures the number of spam messages classified as ham messages in the entire SMS corpus.
- *True negative (TN):* It measures the number of ham messages classified as ham messages in the entire SMS corpus.
- *Accuracy (A):* It measures the overall rate of correct prediction. It is defined by Eq. 3.17.

$$Accuracy = \frac{(TP+TN)}{(TP+FP+TN+FN)} \tag{3.17}$$

- *Precision:* It measures the rate of instances correctly detected as spam messages concerning all instances detected as spam. It is defined by Eq. 3.18.

$$Precision = \frac{TP}{(TP+FP)} \tag{3.18}$$

- *Recall or True Positive Rate (TPR):* It measures the proportion of spam messages which are identified correctly, as defined by Eq. 3.19

$$Recall = \frac{TP}{(TP+FN)} \tag{3.19}$$

- *Specificity or True Negative Rate (TNR):* It measures the proportion of ham messages that are identified correctly, defined by Eq. 3.20

$$TNR = \frac{TP}{(TP+FN)} \tag{3.20}$$

- *f1 Score:* It is the harmonic mean of Precision and Recall. It is defined by Eq. 3.21.

$$f1Score = \frac{(2*Precision*Recall)}{(Precision+Recall)} \tag{3.21}$$

Table 3.1 Confusion Matrices for all models.

			Predicted							
	Train-Test Split		Ham	Spam	Ham	Spam	Ham	Spam	Ham	Spam
Actual	**90% – 10%**	Ham	489	0	487	2	489	0	487	2
		Spam	3	66	2	67	3	66	3	66
	85% – 15%	Ham	722	1	720	3	721	2	722	1
		Spam	9	104	5	108	8	105	9	104
	80% – 20%	Ham	966	2	968	0	962	6	962	6
		Spam	9	138	11	136	8	139	6	141
	75% – 25%	Ham	1207	1	1206	2	1199	9	1202	6
		Spam	16	169	13	172	14	171	12	173
	70% – 30%	Ham	1436	6	1437	5	1438	4	1436	6
		Spam	26	204	21	209	28	202	24	206
			CNN BUNOW		CNN-LSTM BUNOW		CNN GloVe		CNN-LSTM GloVe	

3.5 Results and Discussions

As mentioned earlier, four different models are proposed for spam message classification. The performance of these four models, namely, CNN with BUNOW word embedding, CNN with GloVe Embedding, CNN-LSTM with BUNOW word embedding, and CNN-LSTM with GloVe Embedding are evaluated using Tiago's Dataset [119]. All the models are implemented using Python and evaluated in Colab platform. To measure the accuracy, four models are trained with five different train-test splits (90% – 10%, 85% – 15%, 80% – 20%, 75% – 25% and 70% – 30%). The confusion matrix and the different performance measures are shown in Table 3.1 and Table 3.2 respectively for these five train-test splits.

From Table 3.2, it can be observed that both CNN BUNOW and CNN GloVe perform best for 90% - 10% train-test with 99.46% accuracy. These two models classify all 489 ham messages correctly. Out of 69 spam messages, these two models predict 66 messages as spam, hence having the highest sensitivity (TPR) of 100%. But as the training percentage is decreased, CNN-LSTM BUNOW perform best among these four models with accuracy 99.04%, 99.01%, 98.92% and 98.44% for 85% – 15%, 80% – 20%, 75% – 25% and 70% – 30% train-test splits respectively. Since Long Short Term Memory (LSTM) is capable of remembering information in its memory cell for a long period, the CNN-LSTM model achieves better accuracy as expected. Though GloVe word embedding considers the global context of words, but fails to achieve the best performance with the CNN-LSTM model in this context.

Table 3.2 Performance Measures for four models.

Train-Test Split		CNN BUNOW	CNN-LSTM BUNOW	CNN GloVe	CNN-LSTM GloVe
	TPR	0.9565	0.9710	0.9565	0.9565
	TNR	1.000	0.9959	1.000	0.9959
90% – 10%	Precession	1.000	0.9710	1.000	0.9705
	f1 score	0.9778	0.9710	0.9778	0.9635
	Accuracy	0.9946	0.9928	0.9946	0.9910
	TPR	0.9204	0.9558	0.9292	0.9204
	TNR	0.9986	0.9959	0.9972	0.9986
85% – 15%	Precession	0.9905	0.9730	0.9813	0.9905
	f1 score	0.9541	0.9643	0.9545	0.9541
	Accuracy	0.9880	0.9904	0.9880	0.9880
	TPR	0.9388	0.9251	0.9456	0.9592
	TNR	0.9979	1.000	0.9938	0.9938
80% – 20%	Precession	0.9857	1.000	0.9586	0.9592
	f1 score	0.9617	0.9611	0.9521	0.9592
	Accuracy	0.9901	0.9901	0.9874	0.9892
	TPR	0.9135	0.9297	0.9243	0.9351
	TNR	0.9992	0.9983	0.9925	0.9950
75% – 25%	Precession	0.9941	0.9885	0.9500	0.9665
	f1 score	0.9521	0.9582	0.9370	0.9505
	Accuracy	0.9878	0.9892	0.9835	0.9871
	TPR	0.8870	0.9087	0.8783	0.8956
	TNR	0.9958	0.9965	0.9972	0.9958
70% – 30%	Precession	0.9714	0.9766	0.9806	0.9717
	f1 score	0.9273	0.9414	0.9266	0.9321
	Accuracy	0.9809	0.9844	0.9809	0.9821

Table 3.3 Best Accuracies achieved by four models.

Train-Test Split	**Model**	**Accuracy**	**Epoch**
90% – 10%	CNN BUNOW	0.9946	11
	CNN-LSTM BUNOW	0.9928	7
	CNN GloVe	0.9946	8
	CNN-LSTM GloVe	0.9910	15
85% – 15%	CNN BUNOW	0.9880	6
	CNN-LSTM BUNOW	0.9904	12
	CNN GloVe	0.9880	9
	CNN-LSTM GloVe	0.9880	11
80% – 20%	CNN BUNOW	0.9901	11
	CNN-LSTM BUNOW	0.9901	10
	CNN GloVe	0.9874	7
	CNN-LSTM GloVe	0.9892	10
75% – 25%	CNN BUNOW	0.9878	11
	CNN-LSTM BUNOW	0.9892	5
	CNN GloVe	0.9835	14
	CNN-LSTM GloVe	0.9871	11
70% – 30%	CNN BUNOW	0.9809	7
	CNN-LSTM BUNOW	0.9844	5
	CNN GloVe	0.9809	14
	CNN-LSTM GloVe	0.9821	10

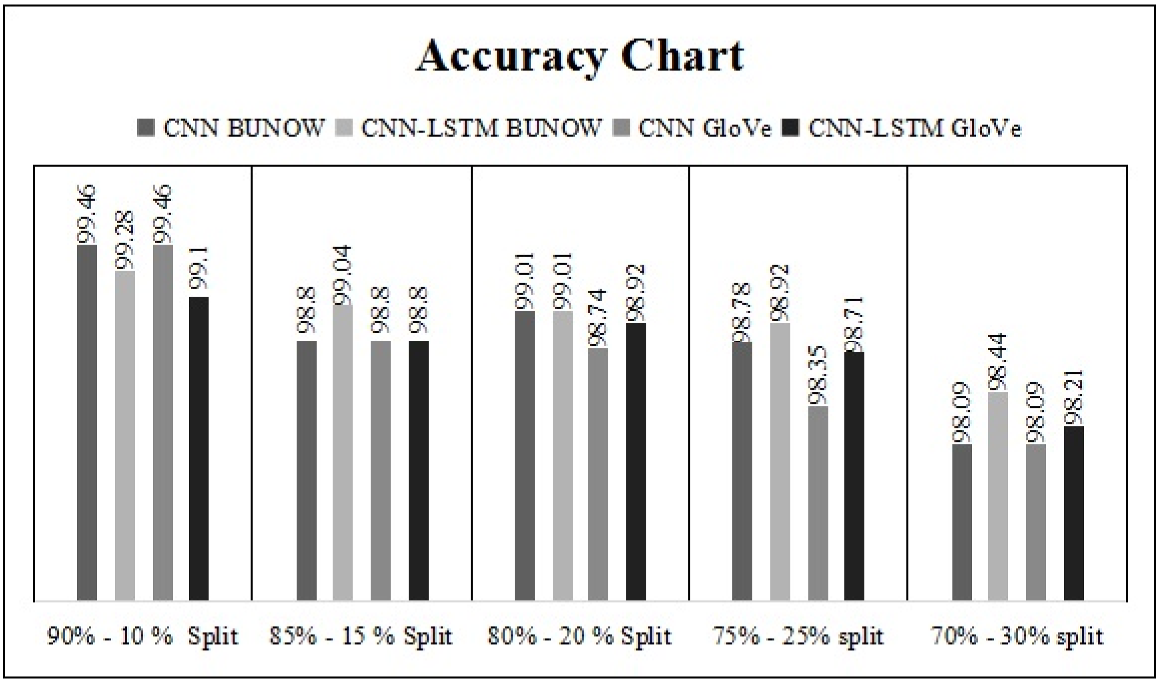


Fig. 3.9 Best Accuracies achieved by four models.

Table 3.4 Results of Paired comparisons.

	CNN BUNOW	CNN-LSTM BUNOW	CNN GloVe
CNN-LSTM BUNOW	0.300	-	-
CNN GloVe	0.192	0.110	-
CNN-LSTM GloVe	0.369	0.002	0.643

All of these accuracies and confusion matrices are measured after the 15^{th} epoch. Although, all of the models achieved a better accuracy in earlier epochs and because of over-fitting their accuracies decreased. The best accuracy of each model and the epoch at which this best accuracy is achieved are listed in Table 3.3 for each train-test split. The best accuracies achieved by each of these four models are compared in Fig. 3.9.

For statistical evaluation of the results among four proposed models, paired comparison [130] has been carried out. The p-values of the paired comparison are given in Table 3.4. Since all the pairs except pair CNN-LSTM BUNOW and CNN-LSTM GloVe exhibit p-value more than 5% significance level, all the proposed models significantly equal in terms of statistical measures. Since the pair CNN-LSTM BUNOW and CNN-LSTM GloVe shows p-value 0.002, which is less than 5% significance, it can be concluded CNN-LSTM BUNOW perform better with respect to CNN-LSTM GloVe model.

Table 3.5 Performance measures of the Youtube data set for 80% – 20% train-test split.

Train-Test Split		CNN BUNOW	CNN-LSTM BUNOW	CNN GloVe	CNN-LSTM GloVe
80% – 20%	TPR	0.9121	0.9024	0.9073	0.8585
	TNR	0.9679	0.9786	0.9733	0.9947
	Precession	0.9689	0.9788	0.9738	0.9944
	f1 score	0.9396	0.9390	0.9393	0.9215
	Accuracy	0.9387	0.9388	0.9388	0.9234

Table 3.6 Result of ablation test on the Youtube data set.

Train-Test Split		CNN	CNN-LSTM
90% – 10%	TPR	0.3902	0.5024
	TNR	0.8396	0.8930
	Precession	0.7273	0.8374
	f1 score	0.5079	0.6280
	Accuracy	0.6046	0.6888

To judge the versatility of the proposed method, the experiment has also been applied on Youtube spam collection dataset [120]. It consists of 1956 different data items, out of these 1005 (51.38%) data items are spam and 951 (48.62%) data items are ham. The experimental results are shown in Table 3.5 for 80% - 20% train-test split as a representative. To show the importance of BUNOW and GloVe Word embedding techniques, the ablation test has been carried out on the Youtube dataset by eliminating word embedding process and the performance metrics of CNN and CNN-LSTM models are shown in Table 3.6 for same train-test split. It can be observed from the results, by eliminating word embedding process, CNN and CNN-LSTM models achieve 60.46% and 68.88% accuracy whereas all the models achieve above 92% accuracies when com-bined with word embeddings.

To measure the performance, the accuracies of the proposed CNN-LSTM BUNOW model achieved after 15 epochs (Table 3.3) are compared with the accuracies of other exiting models. For comparing the performance, the accuracies achieved in a given train-test split on Tiago's Data set [119] are considered. It can be concluded from Table 3.7, the proposed CNN-LSTM BUNOW model outperforms all of the state-of-the-art machine learning algorithms mentioned in [[22], [30], [24], [32], [25] [33], [34], and [35]] by achieving better accuracy. The model proposed by Roy et al. [35] achieves

Table 3.7 Comparison of CNN-LSTM BUNOW model with other existing models.

Existing Models	Year of Publication	Train-Test Split	Accuracy achieved by Existing Models	Accuracy of Proposed CNN-LSTM BUNOW model
Almeida et al. [22]	2011	70% - 30%	97.64%	98.44%
Taheri et al. [30]	2017	70% - 30%	98.11%	98.44%
Navaney et al. [24]	2018	75% - 25%	97.4%	98.92%
Popovac et al. [32]	2018	70% - 30%	98.4%	98.44%
Alzahrani et al. [25]	2019	80% - 20%	94.26%	99.01%
Annareddy et al. [33]	2019	80% - 20%	97.8%	99.01%
Roy et al. [34]	2019	66.7% - 33.3%	99.44%	98.44%
Chandra et al. [35]	2019	70% - 30%	97.81%	98.44%

better accuracy than the proposed CNN-LSTM BUNOW model. But Roy et al. proposed a complex multichannel CNN architecture and perform a 10-fold cross-validation which takes huge time on training the model. The CNN-LSTM BUNOW model is much simpler and achieves acceptable accuracy in less training time.

3.6 Summary

In this chapter four different neural network models viz., CNN BUNOW, CNN-LSTM BUNOW, CNN GloVe, and CNN-LSTM GloVe are proposed. The models were applied to Tiago's data set to distinguish spam from non-spam messages. The data set is mostly composed of ham messages. To obtain a good model, pre-processing of the data is performed. Pre-processing steps include lowercasing the text, tokenization, lemmatization, and removal of stop words, symbols, numbers, and words with lengths less than 2. To achieve better accuracy, the texts were converted into two different types of embeddings, BUNOW embedding, and GloVe embedding. Proposed models are trained and tested on different train-test splits. The accuracies of the proposed models are calculated on these different train-test splits. It should be noted that though both CNN BUNOW and CNN GloVe achieve best accuracy on 90% - 10% train-test splits but CNN-LSTM BUNOW perform best among four models with accuracy 99.04%, 99.01%, 98.92% and 98.44% for 85% – 15%, 80% – 20%, 75% – 25% and 70% – 30% train-test splits respectively. Though GloVe word embedding considers the global context of words but does not succeed to achieve the best accuracy in this context. The accuracy of CNN-LSTM BUNOW is compared with other existing Machine Learning models and achieves better accuracy in most cases. This work can be extended by incorporating more complex pre-processing techniques like N-grams, spelling correction, etc. In the future, the performance of complex deep neural structures with word embedding techniques can also be evaluated for spam message classification.

Chapter 4

Annotation Detection for Cyber-bullying Messages

4.1 Introduction

Cyber-bullying is the use of the digital electronic devices to send or post illegal text or images over internet with intention to hurt or embarrass another person. Nowadays, people around the world sharing data and transferring knowledge by using different online forums, blogs, social networking sites. Online communities and social networks have become more common platform for communication, some users use these communities in illegal and unethical ways, which lead teens and youth people to get bullied over the internet. Typical cyberbully behaviour include insulting, humiliating, offensive or threatening messages or calls, identity theft, exclusion, the publication of confidential information, manipulation of photographs, the recording of physical assaults that are subsequently disseminated etc. [131]. People who face bullying in their childhood or at the teenage are at higher risk of suffering from anxiety, depression and low esteem than those who are not bullied [132]. Cyber-bullying victims report a higher level of suicidal trends and attempts, depression and anxiety, poor academic performance, poor work performance, and poorer physical and mental health. A survey of 174 students of 8th standard aged between 11-15 years in Delhi showed that, 8% indulged in cyber-bullying and 17% reported being victimized by such acts [133]. In 2018, Pew Research study [134] found that a majority of adolescents (59%) experienced some form of cyber-bullying. A more broad study in 2020 shows that this isn't unique to teens [135], with around two-thirds of adults under 30 having experienced online harassment. Therefore immediately there is a need to consider and tackle cyber-bullying from various perspectives including automatic detection and prevention.

Cyber-bullying prevention measures include human interference, deleting offensive terms, blacklisting or scoring of the author's cyber performance and educational awareness. Most online channels that are widely used have safe centers, such as YouTube Safety Center and Twitter Safety and Protection, which offer user assistance and track communications.

Hence to tackle the increasing rate of cyber-bullying, automated detection and prevention methods are in demand. To prevent cyber-bullying educational awareness, offensive term deletion, and black listing using the user's cyber performance score may be incorporated. Most of the online platforms incorporate safety measures that assist and track communications.

In our study, annotation of cyber-bullying is detected based on the content of the text using three deep neural network models. BUNOW (Binary Unique Number of Word) and GloVe word embeddings are combined with three deep neural models to enhance the performance by considering the context of words. In this proposed method, three annotations are considered, namely racism, sexism, and normal message. The performances of these three word-embedded deep neural network models are compared by combining two pre-processing methods – traditional pre-processing and advanced pre-processing. In the proposed advanced pre-processing URL, numbers, emojis, and abbreviated words are not ignored but are considered in decision making. The experimental results show that the proposed advanced pre-processing method obtains better accuracy with respect to all three word-embedded deep learning models with traditional pre-processing.

4.2 Past Works

Many researchers used supervised machine learning models for cyber-bullying detection for both text-based and multi-model. They used logistic regression [41, 42], Random forest [41–44], Support vector machine [41, 42, 44], Naïve Bayes [41, 43], and Decision Tree [44] for that purpose. The limitation of machine learning approaches is that generally, it does not consider the context of words. The decision is taken by simply counting the occurrence of words and assigning weights to words. Deep neural networks are also used by some researchers that overcome the limitations of traditional machine learning approaches.

Pronunciation-based cyber-bullying message detection technique [45] using Convolutional Neural Network (CNN) has been proposed. In this method, traditional pre-processing has been used for the removal of irrelevant words to increase the performance of the proposed technique. Approaches to detect cyber-bullying messages using GloVe (Global Vector) word-embedded convolution neural network were proposed [46, 47]. These techniques removed noisy data and duplicate tweets but no encouraging results were found. Cyber-bullying message detection using deep learning models has also been proposed [48–51]. The models have been tested by aggregating messages from different social media platforms. Different deep learning models like CNN, Long Short-Term Memory (LSTM), etc. are used by combining GloVe and SSWE (Sentiment Specific Word Embedding) word embedding techniques for achieving better performance. Many researchers also used Recurrent Neural Network (RNN), and Gated Recurrent Unit (GRU) for detecting cyber-bullying [52, 53]. To increase the performance of the proposed method, text cleaning, tokenization, lemmatization, stemming, and removal of stop words are used as in traditional pre-processing. Locality Sensitive Hashing with Similarity-Based Word Embedding (LSHWE) [54] has been used for the detection of cyber-bullying messages and the applicability of this method is judged by comparing the performance with GloVe, SSWE, Word2Vec word embedding techniques. The experimental results show that LSHWE gives better performance.

It can be concluded from the above studies that several researchers applied different deep neural networks for the detection of cyber-bullying messages. Some of these studies include word embedding for performance enhancement. Most of the work includes traditional pre-processing which includes lower casing the text, tokenization, lemmatization, and removal of stop words. Tokens consist of symbols, numbers, emojis, and URLs that are simply ignored and are not used in decision making. But these may have a significant influence. Also, annotation detection of cyber-bullying messages is very rare. These encourage us for this work.

4.3 Materials and Methods

The methodology for cyber-bullying annotation detection is illustrated in Figure 4.1. Initially, the messages are pre-processed by using two techniques – the traditional method and the advanced method. The advanced pre-processing method includes all the steps of the traditional method along with consideration of URLs, emojis, and abbreviations within messages. The pre-processed messages are used to train three deep neural models, namely Convolution Neural Network (CNN), Long Short-Term Memory (LSTM), and Bidirectional Long Short-Term Memory (BLSTM). To enhance the accuracy of the annotation detection, two word-embedding techniques BUNOW and GloVe are used. Finally, the performances of the three word-embedded deep neural models are evaluated. The broad steps of the proposed method are

- Text Pre-processing.
- Word Embedding.
- Deep neural network models for annotation Detection.

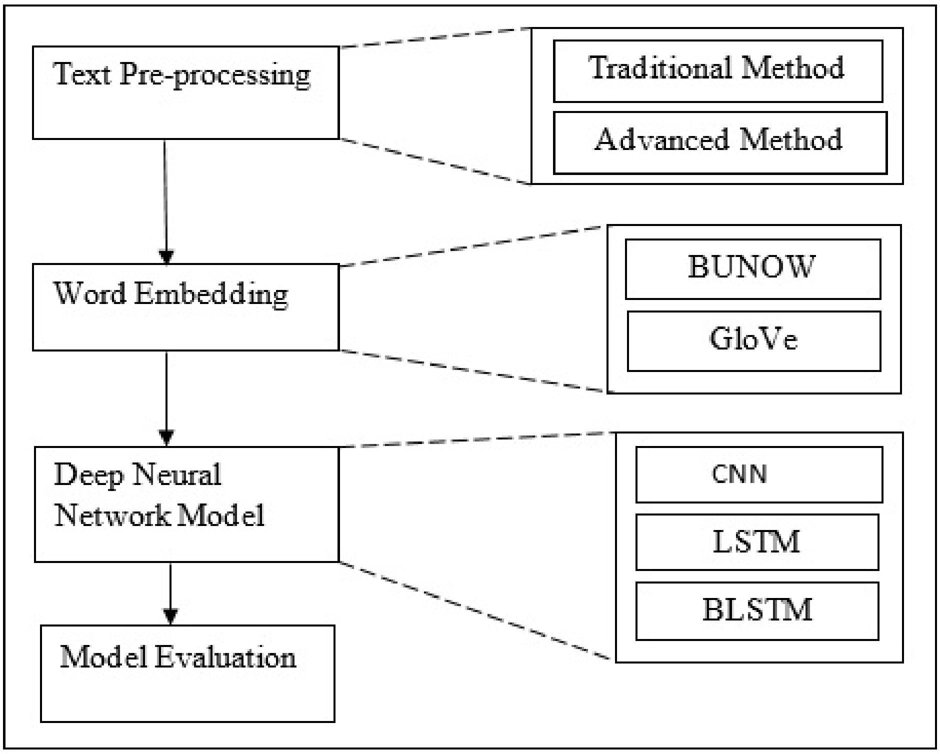


Fig. 4.1 Model Description for cyber-bullying annotation detection.

- Model Evaluation.

4.3.1 Text Pre-processing

Text pre-processing has been used extensively in text classification. Many researchers suggested that good text pre-processing can improve the result of text classification [127, 128]. Most of the researchers performed traditional text pre-processing [45, 52, 53] for text classification. In this work, we proposed an advanced pre-processing technique and compared the performance with the traditional method.

Traditional Pre-processing

Traditional pre-processing includes steps like lowercasing, tokenization, elimination of non-alphabetic tokens, lemmatization, and removal of stop words. Most of the researchers employed this traditional pre-processing step before using classification schemes.

Lowercasing: In this step, each alphabet present within text messages are converted into lowercase to make them in symmetric form.

Tokenization: After lowercasing, the characters in the text messages are grouped to form tokens. Token consists of n words is known as n-grams tokenization. In this experiment, 1-gram tokenization is used where each token consists of only one word. For example, if the input text is "never give up" then after tokenization, there will be three tokens 'never', 'give', 'up'. Tokens may include words, numbers, punctuation, URLs, emojis, symbols, etc, which are processed further for text classification.

Removal of non-alphabetic tokens: After tokenization, all non-alphabetic tokens i.e, the tokens not made with only alphabets a, ..., z are removed. For example, if the tokens are 'this', 'is', '#', 'value', 'is', '35', then after application of this step yields the tokens as 'this', 'is', 'value', 'is'. In traditional pre-processing non-alphabetic tokens are discarded and are not taking part in decision-making for text classification.

Lemmatization: In this step, tokens are reduced to their base form by removing ending variation using morphological analysis of words, which is also known as the lemma. For example, if tokens are 'troubles', 'cars', 'growing' lemmatization will produce 'trouble', 'car', 'grow' as output. Lemmatization is used to find the root words of the tokens.

Stop words removal: The most frequently and commonly used words in a language are termed stop words. They are less significant in decision making and hence can be removed to gain more focus on important words present in the text. Stop words like 'a', 'an', 'the', 'be', 'for', 'do', 'its', 'yours' etc. should be removed before classification to increase the performance of the classification scheme.

Advanced Pre-processing

The limitation of traditional pre-processing is the removal of non-alphabetic tokens which may include URLs, emojis, and abbreviations. But URLs may point to offensive content. The same may happen also for emojis and abbreviations. Thus proposed advanced pre-processing technique includes all the steps of traditional pre-processing along with consideration of non-alphabetic tokens. The steps for non-alphabetic tokens are:

URL replacement: Since URL content may have some offensive images and videos, URLs present within the messages are considered an important feature in deciding the annotation of cyberbullying. In this work, supervised learning is tried where the data set contains messages with three labels – 'racism', 'sexism', and 'normal'. Initially, all the URLs present within the messages in the data

Table 4.1 URL occurrence in different annotation.

URL	None	Label Racism	Label Sexism
http://â€¦	21	7	4
http://on...:	1	0	0
http://t.â€¦	20	3	8
http://t.câ€¦	22	6	5
http://t.co/04XVHx5vVl	0	1	0
http://t.co/06ApCFGDF0	1	0	0
http://t.co/07UbzYHWjq	1	0	0
http://t.co/08ZwRcTo88	0	1	0
http://t.co/0aIkkR7Csh	1	0	0
http://t.co/0bO7JAp1OP	0	0	1

set are detected using regular expressions and replaced by appropriate text. For example, if a message with the label 'racism' contains an URL, the URL is replaced by the text 'URLRACISM'. Similarly, an URL present in the message with the label 'sexism' and 'normal' is replaced by 'URLSEXISM' and 'URLNORMAL' respectively. A database is also created to store the number of occurrences of a particular URL under different labels. A snapshot of the database is shown in Table 4.1. This database will help to predict the annotation of a new message without the label. A new message contains a URL; the URL will be searched within the database. If the URL is found within the database, the URL is replaced by appropriate text by the voting method, i.e. the URL will be replaced by the label in which it occurs maximum. The database is also updated by incrementing occurrence by one under the label in which the URL occurs maximum. If the URL is not present in the database, the URL is removed from the message and the annotation is detected based on other tokens in the message. This new URL is added to the database by assigning value one in the appropriate label depending on the annotation. Initially, the database is created only once, after that it automatically grows if unknown URLs are found.

Number and emojis replacement: Numbers present in the messages are searched by using regular expressions and are replaced by the text 'number'. Similarly, emojis within the message are detected using the Unicode of emojis and are replaced by the text 'emoji'. For example, if the text is '402 is a smiley face \U0001f602' then the application of this step produces the output 'number is a smiley face emoji'.

Replacement of abbreviation word: Generally chat message consists of a huge number of abbreviated words. But from these abbreviated words, it is difficult to predict the annotation of the messages. To enhance the performance, abbreviated words are replaced with their full form. For this

Table 4.2 Sample of abbreviated words.

Abbreviated Words	Full form of abbreviated words
“4ao”	“For adult only”
“asap”	“as soon as possible”
“b2b”	“business to business”
“asl”	“age sex location”
“cv”	“curriculum vitae”
“faq”	“frequently asked questions”
“fb”	“facebook”
“ilu”	“I love you”
“wtf”	“what the fuck”
“p.a”	“per annum”

purpose, a dictionary of 229 abbreviated words is created. A snapshot of the dictionary is shown in Table 4.2. For example, if the input message is ‘This picture is 4ao, ilu imu!’, the expanded message ‘this picture is for adults only, i love you i miss you!’.

Removal of HTML Tag: HTML tags are not an important feature for annotation detection of cyber-bullying. Hence all HTML tags are removed from the message. For example, if the input text is ‘<s>your generation is so advanced</s>’ then after the removal of HTML tags output will be ‘your generation is so advanced’.

4.3.2 Proposed Deep Neural Models

For annotation detection of cyber-bullying messages, three deep neural models with two word-embedding techniques (BUNOW and GloVe) have been used here. BUNOW and GloVe word embedding are described in section 3.4.3. In this experiment, Convolution Neural Network (CNN), Long Short-Term Memory (LSTM), and Bidirectional Long Short-Term Memory (BLSTM) models are used.

Convolution Neural Network

The proposed CNN model is depicted in Figure 4.2. In this model, the word embedding technique is considered as the input layer. The hidden layer consists of convolution layers, max

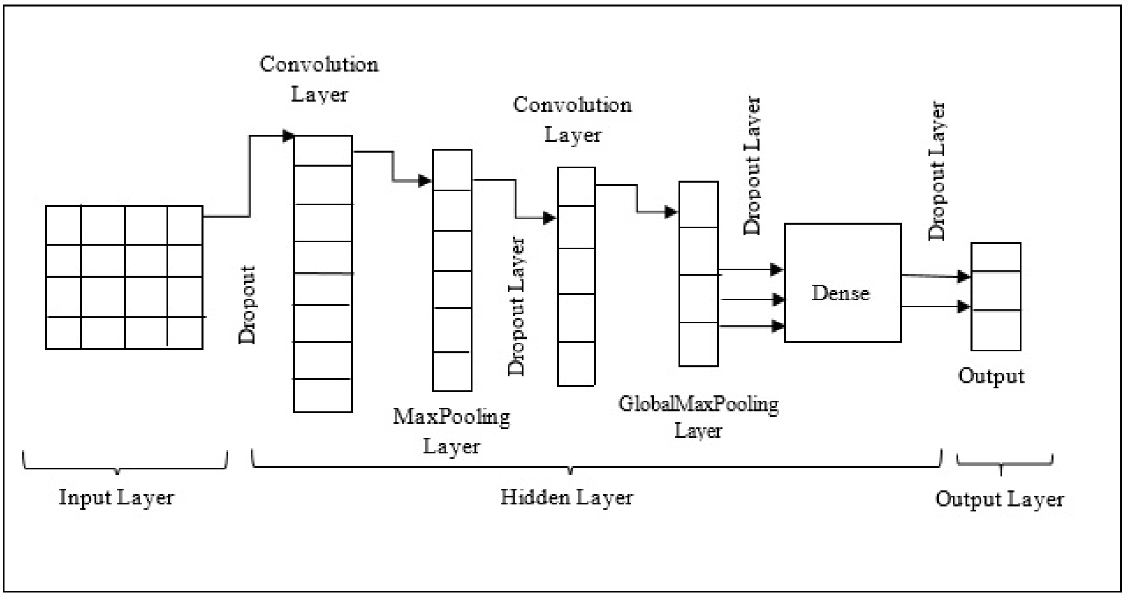


Fig. 4.2 Proposed CNN Architecture.

pooling layers, dropout layers, and fully connected layers. As shown in Figure 4.2, the hidden layer contains two 1-dimensional (1-D) convolution layers. The number of neurons in the first 1-D convolution layer is equal to the number of distinct words in the training corpus. Since URLs, emojis, and abbreviated words are replaced by appropriate strings in advanced pre-processing; the dimensions of the input layer may differ in the case of traditional and advanced pre-processing. The size of the filter and kernels are 32 and 3 respectively. The swish activation function has been applied to achieve non-monotonic behaviour. To decrease the dimension of feature vectors without losing the essential features, a 1-D max pooling layer with pool size 3 has been used. The output of the max pooling layer has been fed into another combination of the 1-D convolution layer and global max pooling layer. Finally, a fully connected dense layer of 128 units has been employed. To prevent over-fitting, multiple dropout layers have been used with a drop rate of 30 percent. The output layer consists of three units with a sigmoid function because three annotations are present in this experiment. The weights of the CNN network are randomly initialized.

LSTM architecture

The proposed LSTM model is shown in Figure 4.3. This model is similar to the proposed CNN model. Only the hidden layer has been modified. The hidden layer consists of the LSTM layer and a fully connected dense layer. The LSTM layer is designed with 256 units and a recurrent dropout

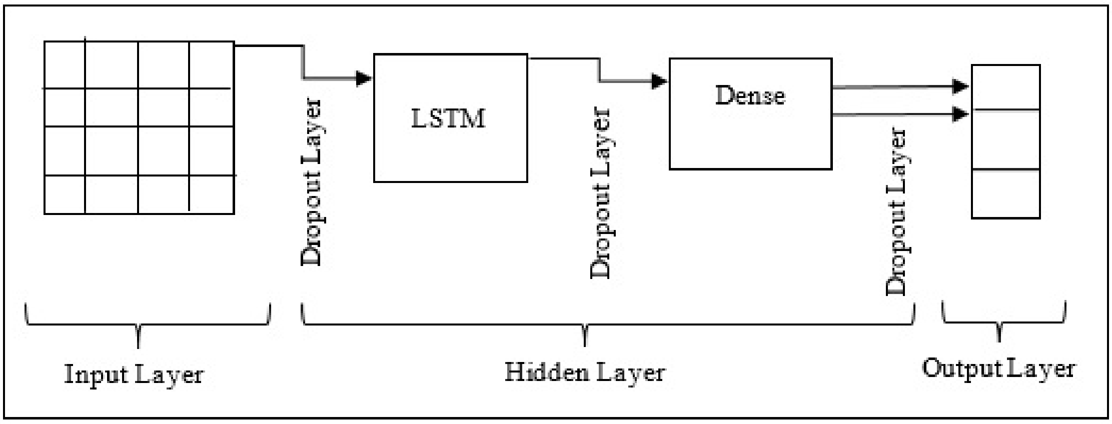


Fig. 4.3 Proposed LSTM architecture.

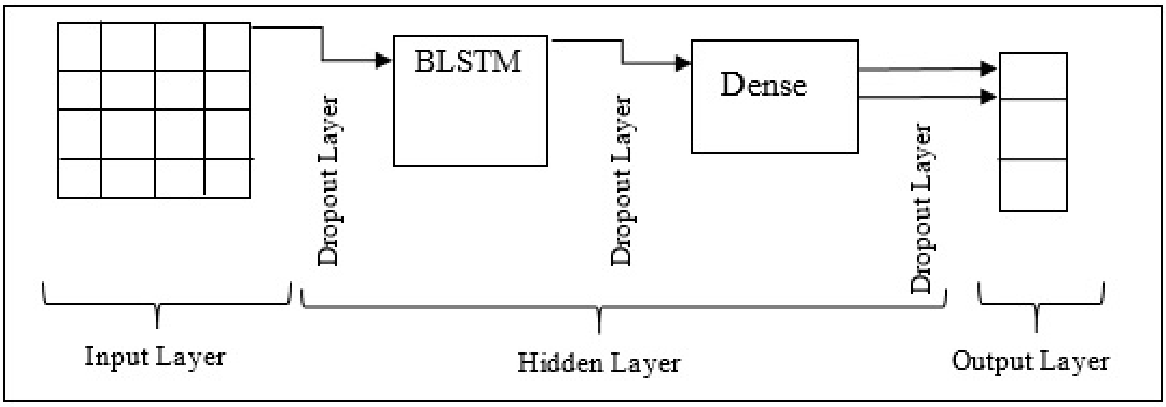


Fig. 4.4 Proposed BLSTM architecture.

rate of 0.3. A fully linked dense layer is present at the end of the model with 128 units and the swish activation function has been used.

BLSTM architecture

Figure 4.4 shows the proposed BLSTM architecture and it is similar to LSTM architecture. In place of LSTM layer, BLSTM layer has been used. The parameter's value remains the same as with LSTM model.

4.3.3 Working Principle

Initially, the messages in the data set are pre-processed by both traditional and advanced pre-processing techniques as discussed earlier. From these pre-processed messages, two batches are created – one for training purposes and another for testing. The distinct words in the training set are identified. Each word is represented using a vector following the BUNOW and GloVe word embedding techniques. Next, the number of words in the longest message (LW_{max}) within the training set is calculated. Each message in the training set is represented by a fixed size vector of length LW_{max} by concatenating the vector representation of the words that constitute the message. If the message length is less than LW_{max}, the remaining components in the fixed length vector are set to zero. These vectors are used to train three proposed deep learning models.

For detecting the annotation of a cyber-bullying message, the message is represented as fix length vector of size LW_{max} by a similar approach described above. If the message length is more than LW_{max}, the first LW_{max} words are considered. If a new word that is not in the vocabulary of the training data set is present within the message, the word is eliminated before representing the message using a vector. The vector representation is fed into the trained model to identify the annotation of the message.

4.3.4 Performance Evaluations

Performances of the proposed models are evaluated using several test matrices derived from the confusion matrix. These are described as follows.

- ***True Positive (TP):*** Among the testing data set, the number of data instances for which the model correctly predicted the positive class as positive.
- ***True Negative (TN):*** Among the testing data set, the number of data instances for which the model correctly predicted the negative class as negative.
- ***False Positive (FP):*** Among the testing data set, the number of data instances for which the model wrongly predicted the negative class as positive.
- ***False Negative (FN):*** Among the testing data set, the number of data instances for which the model wrongly predicted the positive class as negative.
- ***Confusion matrix:*** The performance of deep learning models can be visualized by a tabular format known as the confusion matrix. As this study is based on three annotations namely

Table 4.3 Cell structure of different annotations.

		Predicted class		
		Normal	Racism	Sexism
Actual Class	Normal	Cell1	Cell2	Cell3
	Racism	Cell4	Cell5	Cell6
	Sexism	Cell7	Cell8	Cell9

Table 4.4 Formulae for calculation of TP, TN, FP, and FN for three different classes.

Class	**Evaluation Parameter**	**Calculation Procedure**
Normal	TP_1	Cell1
	TN_1	Cell5+Cell6+Cell8+Cell9
	FP_1	Cell4+Cell7
	FN_1	Cell2+Cell3
Racism	TP_2	Cell5
	TN_2	Cell1+Cell3+Cell7+Cell9
	FP_2	Cell2+Cell8
	FN_2	Cell4+Cell6
Sexism	TP_3	Cell9
	TN_3	Cell1+Cell2+Cell4+Cell5
	FP_3	Cell3+Cell6
	FN_3	Cell7+Cell8

racism, sexism, and normal messages, the structure of the confusion matrix is shown in Table 4.3. The evaluation parameters for the three classes can be calculated in Table 4.4.

- ***Accuracy (A):*** It can be measured by the ratio between the total number of correct predictions and the total number of testing data instances. It is defined by Eq. 4.1.

$$Accuracy = \frac{\sum_{i=1}^{3}(TP_i + TN_i)}{\sum_{i=1}^{3}(TP_i + FP_i + TN_i + FN_i)} \tag{4.1}$$

- ***Precision:*** It can be measured by the ratio between the number of correctly predicted positive classes and the total number of instances that are predicted as positive. It is defined by Eq. 4.2.

$$Pricision = \frac{\sum_{i=1}^{3} TP_i}{\sum_{i=1}^{3}(TP_i + FP_i)} \tag{4.2}$$

- ***Recall or Sensitivity or True Positive Rate (TPR):*** It measures the ratio between the actual fractions of all positive data instances, which were correctly predicted as positive by the predicted model. It is defined by Eq. 4.3

$$TPR = \frac{\sum_{i=1}^{3} TP_i}{\sum_{i=1}^{3}(TP_i + FN_i)} \tag{4.3}$$

- ***f1 Score:*** It is the statistical measure used to find the harmonic mean of Precision and Recall. It is defined by Eq. 4.4.

$$f1 - Score = \frac{(2 * Precision * Recall)}{(Precision + Recall)} \tag{4.4}$$

- ***False Positive Rate (FPR):*** It measures what fraction of the negative class became incorrectly classified by the predicted model. It is defined by Eq. 4.5.

$$FPR = \frac{\sum_{i=1}^{3} FP_i}{\sum_{i=1}^{3}(TN_i + FP_i)} \tag{4.5}$$

- ***Area under the Curve (AUC) of Receiver Characteristic Operator (ROC):***The ROC is a two-dimensional probability curve. In this curve, the X–axis represents the False Positive Rate (FPR), and the Y–axis is used to plot the True Positive Rate (TPR) at various threshold values. To empower visualizing and establishing classifier performance, ROC graphs are used. Bradley et. al. [136] shows the method how to calculate the Area under the ROC Curve (AUC). Since the X and Y axes plot the ratio, the range of these axes varies between 0 and 1.0. Thus AUC value must be less or equal to 1 since it is a portion of a unit square. Random guessing (i.e. probability of correct prediction is 0.5) generates a diagonal line from (0, 0) to (1, 1) as shown in Figure 4.5 using the dotted line. It has an area of 0.5. All the models should have an AUC of more than 0.5. Areas under ROC curves (AUC) are used to compare the usefulness of tests. Greater area implies more useful tests.

4.4 Results and Discussions

The experiment has been carried out using the Twitter dataset [137]. It consists of a total of 16848 data instances and three annotations namely sexism, racism, and normal. Sexism, racism, and

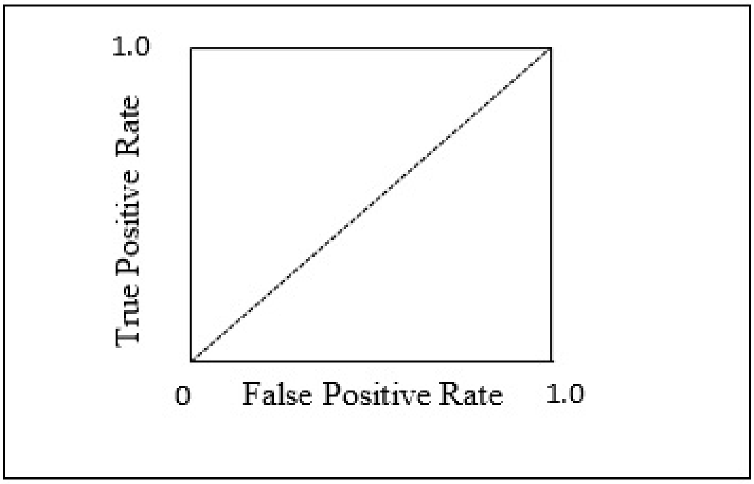


Fig. 4.5 ROC Curve.

normal consist of 3377, 1970, and 11501 data instances respectively. Figure 4.6 shows the dataset with percentages of occurrence.

Each class is assigned a numeric value for computation purposes. Three classes, namely, normal, racism and sexism are represented by numeric values 0, 1, and 2 respectively. The data set has been split for training and testing purposes with 80% and 20% of the data set respectively. The testing set consists of 3370 data instances of which 2326 data instances are of normal class, 386 are of racism class and 658 data instances are within sexism class.

To perform a comparative study between cyber-bullying annotations detection techniques using word-embedded deep neural networks several experiments have been performed. Three deep neural models are embedded with two word-embedding techniques to produce six models namely BUNOW embedded CNN, BUNOW embedded LSTM, BUNOW embedded BLSTM, GloVe embedded CNN, GloVe embedded LSTM, and GloVe embedded BLSTM. Each of these six models is studied with traditional and advanced pre-processing to compare their performances.

Table 4.5 shows the confusion matrix of each model. Since the goal is to predict the annotation of cyber-bullying with three annotations namely normal, racism and sexism, therefore it is a multi-class classification problem. The performance evaluation metrics like TP, TN, FP, and FN are calculated using equations in Table 4.3 and Table 4.4. After applying traditional pre-processing over the dataset and performing BUNOW word embedding we noticed from Table 4.5 out of 2326 normal testing data instances, the proposed CNN model predicts 2012 data instances correctly, the proposed LSTM model predicts 2036 data instances correctly, and the proposed BLSTM model predict 2008 data instances correctly. Whereas when we applied advanced pre-processing and the same word embedding over the same testing dataset our proposed model predicts 2044,2115, and 2117 data instances correctly

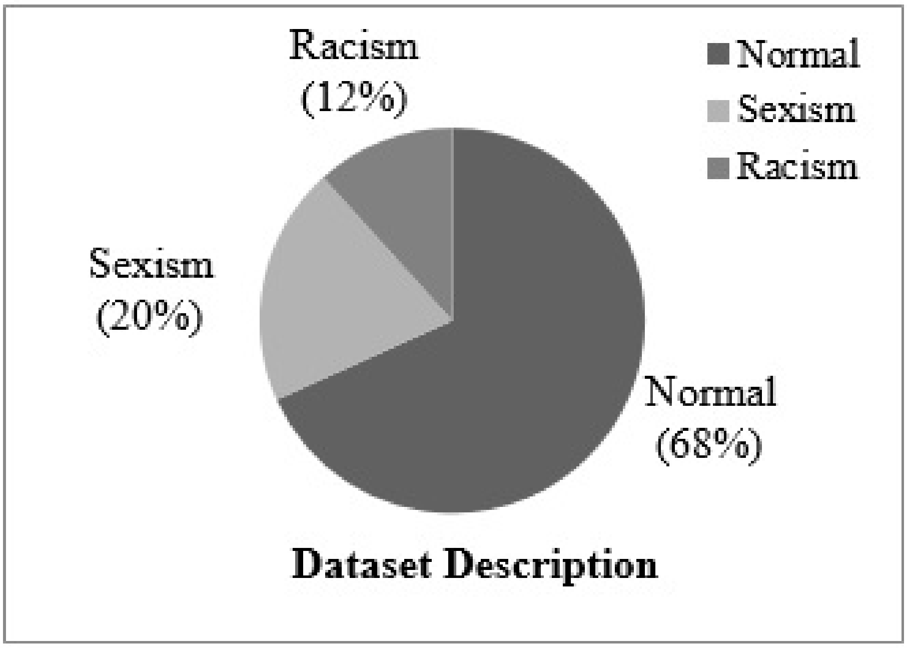


Fig. 4.6 Data set description.

for CNN, LSTM, and BLSTM models respectively. Out of 386 testing racism data instances our proposed CNN, LSTM, and BLSTM models predict 246,241, and 269 respectively whereas advanced pre-processing predicted 251,270, and 255 data instances correctly for the BUNOW word embedding model. Out of 658 sexism testing data instances, traditional pre-processing predicted 364,425, and 33 data instances correctly whereas advanced pre-processing 473,391, and 412 respectively for BUNOW word embedding proposed CNN, LSTM, and BLSTM models. We also observed that for the GloVe word embedding model in most cases our advanced pre-processing predict more data items correctly than traditional pre-processing. So we concluded that advanced pre-processing can outperforms traditional pre-processing because in the case of advanced pre-processing we have considered more features. If we compare GloVe and BUNOW word embedding, it can be observed that GloVe word-embedded deep neural models are capable to predict different annotations more efficiently rather than BUNOW word embedding.

After computing the confusion matrix, we have calculated the precision, recall, and f1 score of each model by using equations 4.2, 4.3, and 4.4 respectively. The values are presented in Table 4.7 and Table 4.7 for the six models with traditional and advanced pre-processing respectively. In the case of traditional pre-processing, we have achieved an f1-score of 0.89 only for GloVe embedded LSTM proposed model for normal class while in advanced pre-processing we have achieved the same f1-score for all three proposed models for the GloVe word embedding model. In the case of the BUNOW word embedding model, we achieved the highest f1-score of 0.86 for normal classes in LSTM and

Table 4.5 Confusion Matrices for all models.

	Predicted											
Actual		0	1	2		0	1	2		0	1	2
	0	2012	85	229	0	2036	101	189	0	2008	126	192
	1	116	246	24	1	129	241	16	1	112	269	5
	2	285	9	364	2	222	11	425	2	210	154	33
	BUNOW embedded CNN with Traditional Pre-processing				**BUNOW embedded LSTM with Traditional Pre-processing**				**BUNOW embedded BLSTM with Traditional Pre-processing**			
		0	1	2		0	1	2		0	1	2
	0	1992	150	184	0	2167	69	90	0	2063	109	154
	1	75	307	4	1	138	246	2	1	94	290	2
	2	173	11	474	2	242	6	410	2	186	4	468
	GloVe embedded CNN with Traditional Pre-processing				**GloVe embedded LSTM with Traditional Pre-processing**				**GloVe embedded BLSTM with Traditional Pre-processing**			
		0	1	2		0	1	2		0	1	2
	0	2044	140	142	0	2115	112	99	0	2117	91	118
	1	128	251	7	1	115	270	1	1	128	255	3
	2	164	21	473	2	263	4	391	2	237	9	412
	BUNOW embedded CNN with Advanced Pre-processing				**BUNOW embedded LSTM with Advanced Pre-processing**				**BUNOW embedded BLSTM with Advanced Pre-processing**			
		0	1	2		0	1	2		0	1	2
	0	2045	135	146	0	2103	105	118	0	2111	91	124
	1	73	311	2	1	111	272	3	1	104	279	3
	2	174	11	473	2	182	4	472	2	183	3	472
	GloVe embedded CNN with Advanced Pre-processing				**GloVe embedded LSTM with Advanced Pre-processing**				**GloVe embedded BLSTM with Advanced Pre-processing**			

BLSTM model with traditional pre-processing whereas in the case of advanced pre-processing we achieved an f1-score of 0.88 in all three models. For other classes, it can be observed that in most of the cases the performance of advanced Pre-processing is better than Traditional pre-processing.

Table 4.6 Performance measures for Traditional Pre-Processing.

		precision	recall	f1-score		precision	recall	f1-score
Traditional Pre-processing	Normal	.83	.87	.85	Normal	.89	.86	.87
	Racism	.72	.64	.68	Racism	.66	.80	.72
	Sexism	.59	.55	.57	Sexism	.72	.72	.72
	BUNOW embedded CNN				**GloVe embedded CNN**			
		precision	**recall**	**f1-score**		**precision**	**recall**	**f1-score**
	Normal	.85	.88	.86	Normal	.85	.93	.89
	Racism	.68	.62	.65	Racism	.77	.64	.70
	Sexism	.67	.65	.66	Sexism	.82	.62	.71
	BUNOW embedded LSTM				**GloVe embedded LSTM**			
		precision	**recall**	**f1-score**		**precision**	**recall**	**f1-score**
	Normal	.86	.86	.86	Normal	.88	.89	.88
	Racism	.66	.70	.68	Racism	.72	.75	.74
	Sexism	.69	.66	.67	Sexism	.75	.71	.73
	BUNOW embedded BLSTM				**GloVe embedded BLSTM**			

The accuracy of these models is calculated and shown in Figure 4.7. It can be observed that the accuracies of all three deep neural models with two word-embeddings are better when the proposed advanced pre-processing technique is applied. The GloVe embedded BLSTM model achieves the best accuracy of 84.93%.

Figure 4.8 and Figure 4.9 depicted the ROC-AUC curves of each model. The AUC value of normal messages in BUNOW embedded CNN is 0.83 indicating that if a curve of FPR v/s TPR is plotted then the area under the curve for normal messages is 0.83 v/s rest of the classes. Similarly, for racism messages, the AUC value is 0.91 v/s rest of the classes and for sexism messages, the AUC value is 0.81 v/s rest of the classes. The model with AUC values close to 1 is a better prediction model.

Table 4.7 Performance measures for Advanced Pre-Processing.

		precision	recall	f1-score		precision	recall	f1-score
Advanced Pre-processing	Normal	.88	.88	.88	Normal	.89	.88	.89
	Racism	.61	.65	.63	Racism	.68	.81	.74
	Sexism	.76	.72	.74	Sexism	.76	.72	.74
	BUNOW embedded CNN				**GloVe embedded CNN**			
		precision	**recall**	**f1-score**		**precision**	**recall**	**f1-score**
	Normal	.85	.91	.88	Normal	.88	.90	.89
	Racism	.70	.70	.70	Racism	.71	.70	.71
	Sexism	.80	.59	.68	Sexism	.80	.72	.75
	BUNOW embedded LSTM				**GloVe embedded LSTM**			
		precision	**recall**	**f1-score**		**precision**	**recall**	**f1-score**
	Normal	.85	.91	.88	Normal	.88	.91	.89
	Racism	.72	.66	.69	Racism	.75	.72	.74
	Sexism	.77	.63	.69	Sexism	.79	.72	.75
	BUNOW embedded BLSTM				**GloVe embedded BLSTM**			

4.5 Summary

In this chapter, we have experimented with three neural network models CNN, LSTM, and BLSTM with two different word embedding techniques BUNOW and GloVe for cyber-bullying annotation detection. Two pre-processing techniques i.e. traditional and advanced pre-processing are applied before the application of these models. Traditional pre-processing includes lowercasing the text, tokenization, removal of non-alphabetic tokens, lemmatization, and stop word removal. Proposed advanced pre-processing includes all traditional pre-processing steps along with replacing URLs, numbers, emojis, and abbreviated words with appropriate strings, and removal of HTML tags. In URLs replacement we have used a separate database, we store every URLs and their occurrences of different classes which leads to detect annotations of new messages more correctly. Moreover we replace emojis as well as we have created a data set of 229 abbreviated word, by replacing them in actual messages we can found the actual meaning of the messages which help us to detect annotations more accurately. The Twitter data set has been used for evaluating the performance. The performances of these models are compared with traditional and advanced pre-processing. It is found that the proposed advanced pre-processing method achieves better accuracy in all six models. It can

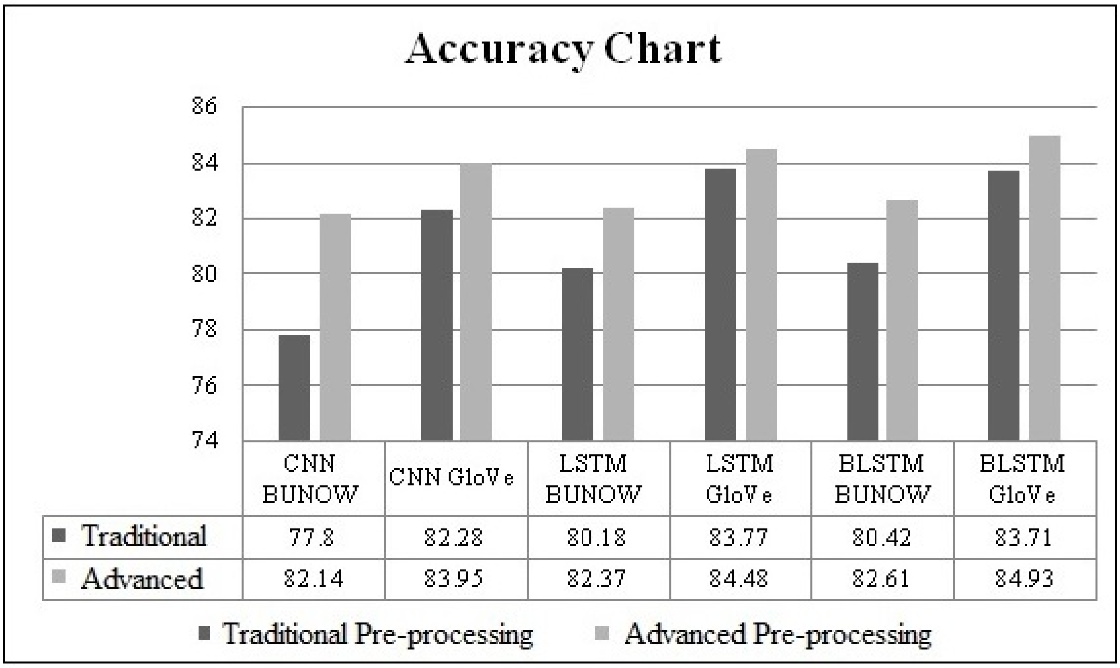


Fig. 4.7 Accuracy Chart for two different models.

be concluded that if the feature vectors are increased appropriately, the accuracy of the model may increase.

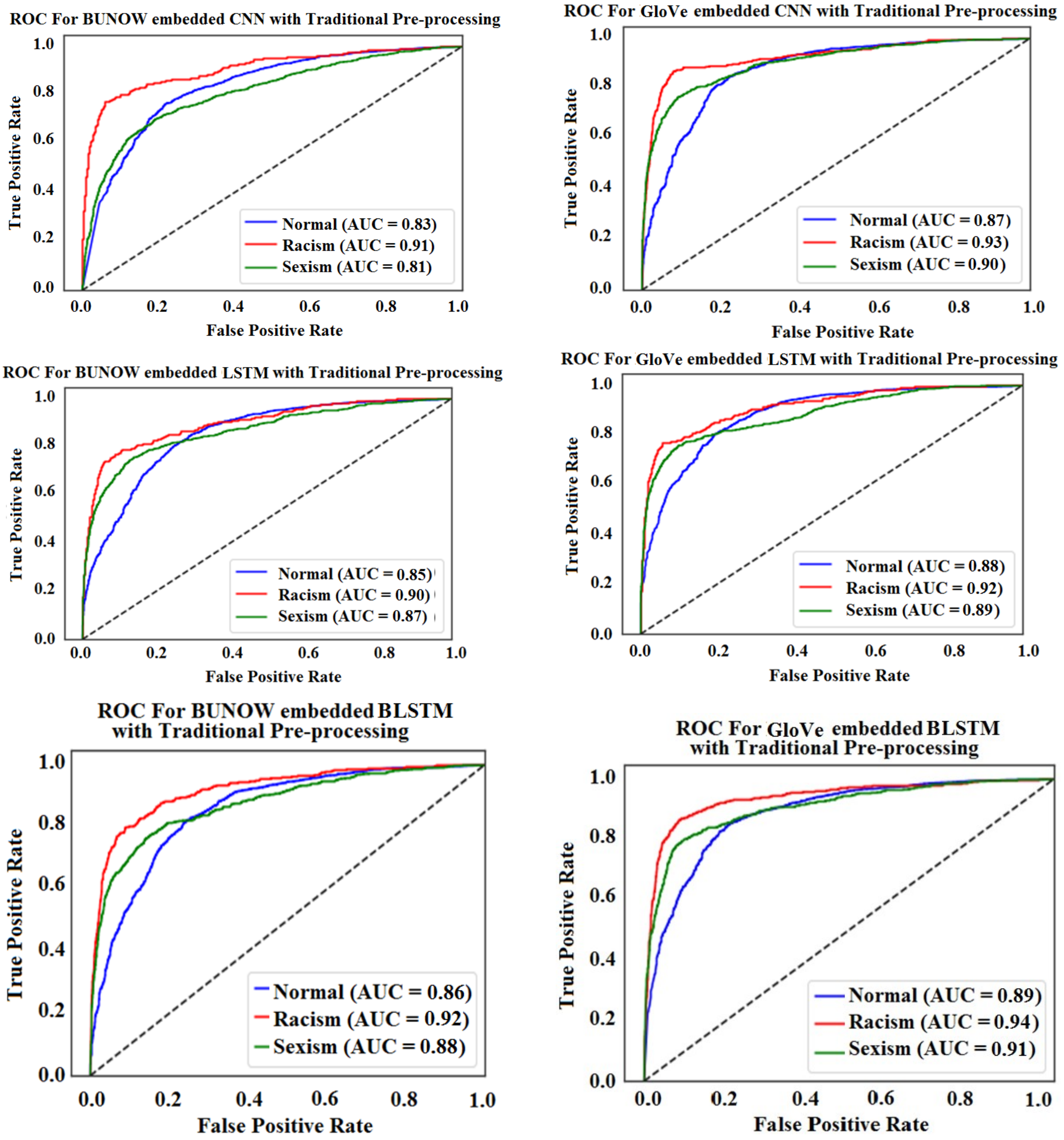


Fig. 4.8 ROC curve of different models using traditional Pre-processing.

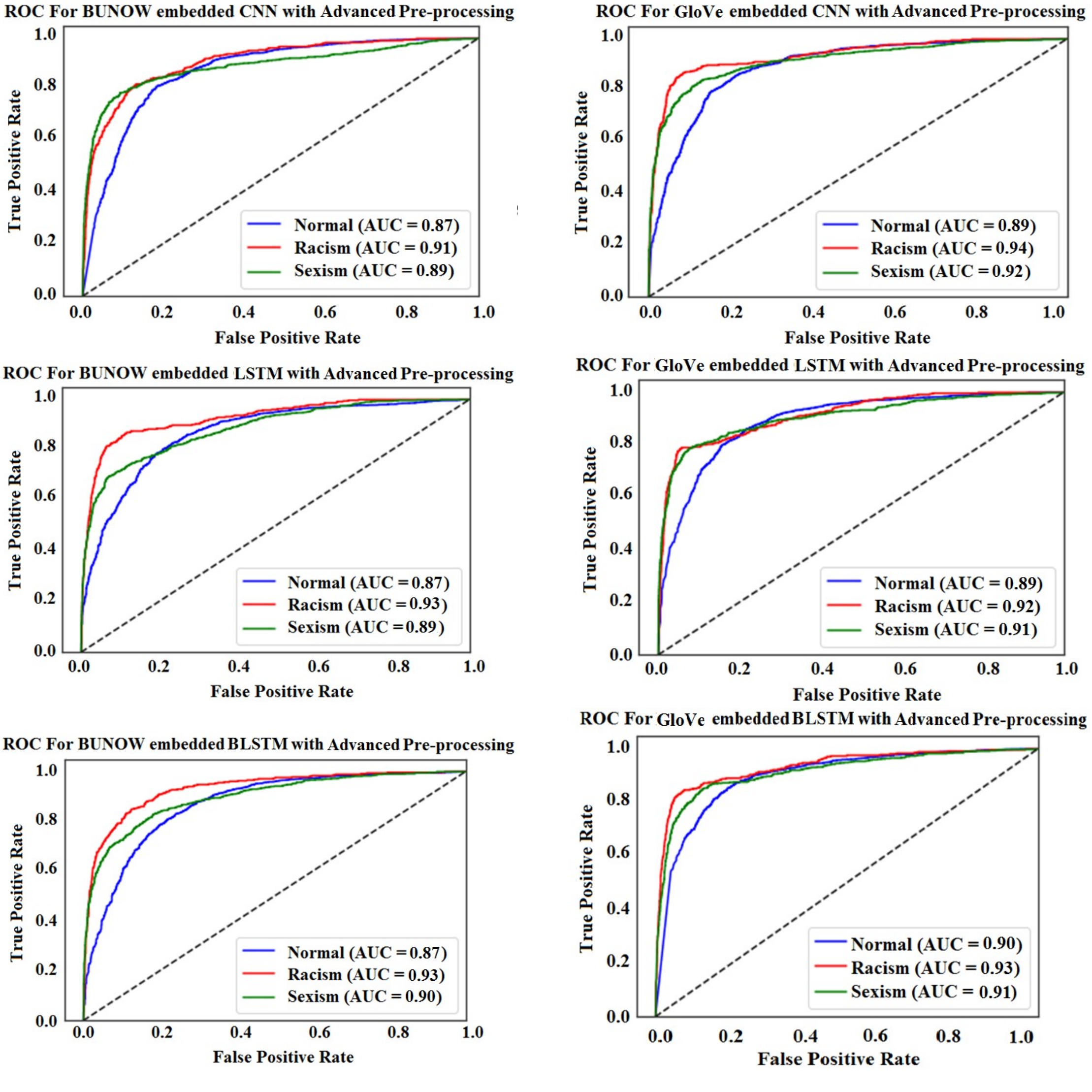


Fig. 4.9 ROC curve of different models using advanced pre-processing.

Chapter 5

Phishing Email Detection

5.1 Introduction

In recent times, Electronic Mail (email) is the universal medium for both formal and business communications. It is a crucial part of our day-to-day life due to inexpensive infrastructure, open access, quickness and reliability. Phishing emails are those emails which somehow successfully trap the user to get their personal information such as email password, debit or credit card details and many more. The attackers disguise themselves as some original organization like a bank, school, college, Microsoft, Facebook, etc., and use some social tricks to make the user believe that the email is sent by a trusted site or organization. Phishing emails not only pose a threat to the individual but also to the financial organization, especially e-commerce. Spam email detection is one of the major tasks in the cyber security because many internet users are getting affected financially. McAfee estimated that annual losses during cybercrime approximately 600 billion US dollar which is equal to 0.8% of global GDP [138]. Phishing emails are most commonly used for phishing attacks. Statistics report shows that 96% of overall phishing attacks arrived through email [139].

In literature, blacklist mechanisms and different classification algorithms based on machine learning and deep learning are used to detect phishing emails. The Blacklist mechanism is a filter-based approach. In this technique, IP addresses and email addresses of back-listed senders are stored in the database depending on their past activity. When a fresh email id is reached, the filter searches the database for this new email id. If it exists in the database, it is considered a phishing email. Otherwise, it is treated as a legitimate URL and ham email. The disadvantages of this mechanism are that the search process is time-consuming as well as it cannot detect new phishing URLs and email addresses.

To overcome the problem of the blacklist mechanism many researchers used machine learning approaches. Support Vector Machine (SVM) [55, 56], K-Nearest Neighbour (KNN) [57], Naïve Bayes (NB) [58], and Decision Tree (DT) [57, 58] have been tried in this context. Many researchers also used an ensemble model [59] which is a combination of multiple base models for detecting phishing emails. The main disadvantage of the traditional machine learning approach is feature identification, which will be used for classification [140]. Optimal feature selection can increase the accuracy of the model. But optimal feature selection is a very time-consuming task.

To resolve the problem of traditional machine learning problem, a deep learning approach has been used. In this approach, the optimal feature can be extracted automatically. The optimal feature will be extracted during the learning phase using the deep neural network and also these architectures are more robust compared to the traditional machine learning approach.

In this research two deep learning models have been proposed for the detection of phishing emails. Both models use pre-trained word embedding for context prediction. The first model uses GloVe word embedding whereas the second model uses the Bidirectional Encoder Representations from Transformers (BERT) model to find the contexts of words within an email. In the first model, context information produced by GloVe is used to train a convolution neural network to detect whether an email is phishing or not. In the second model, the output of BERT is fine-tuned using a simple fully connected neural network for phishing email detection. The performance of these two models is evaluated and compared by merging five publicly available data sets.

5.2 Past Works

There exist several techniques in literature for the detection of a phishing email. Some of the recent studies based on deep learning techniques are reviewed in this section.

Deep Learning Technique was proposed by Abdul Nabi et al. [141] for Spam mail detection. UCI machine learning and the kaggle websites dataset have been used for this purpose. They used the Bidirectional Encoder Representations from Transformers (BERT) word embedding model over their input dataset. They have also used the Bidirectional Long Short Term Memory (BLSTM) deep learning model. They achieved 98.67% accuracy and 98.66% f1-score.

Srinivasan et al. [62] proposed distributed word embedding method with Deep Learning for spam email detection. To measure the performance of the model they have collected their dataset from various well-known datasets like Lingspam, PU, Enron, Apache Spam Assassin train, etc. In the word embedding layer, they have used word2vec, FastText, and a neural bag of words (NBOW). They have applied various models like RNN with Keras embedding, LSTM with Keras embedding, CNN with Keras embedding, etc. Finally, they achieved 99.4% accuracy over the Lingsapm dataset, 95.7% over the PU dataset, and 95.9% over Enron and Apache Spam Assassin.

Sumathi et al. [63] used random forest and deep neural network classifiers for spam email detection. They have collected datasets from UCI Machine Learning Repository. After preprocessing they collected important features by using Random Forest (RF). Finally, they applied Deep Neural Network (DNN) and achieved 88.59% accuracy.

Advanced deep convolution neural network algorithms proposed by Soni [64] for Spam e-mail detection. THEMIS model is used for learning purposes which is an improved intermittent convolutional neural system (RCNN) to classify phishing emails. He considered both contents of the email and the header of the email at the character level and the word level. He achieved 99.84% accuracy for THEMES which is higher than both Long Short Term Memory (LSTM) and Convolutional Neural Network (CNN) in regarding his experiment.

Using a distinct neural network Castillo et al. [65] detected email threats. They have collected their dataset from employees of public companies and government departments. In the word embedding layer, they have used the word2vec model. They used various deep learning models like back propagation (BP), Long Short Term Memory (LSTM), and Convolutional Neural Network (CNN). They achieved 95.68% accuracy using back propagation.

Fang et al. [66] used improved Recurrent Convolution Neural Networks (RCNN) model and their proposed model consist attention mechanism. Their proposed THEMIS model is used to build an email model for the detection of phishing emails. In the embedding layer, they have used the word2vec model. They achieved 99.85% accuracy for this model

Manaswini et al. [67] used the THEMIS model proposed by Fang et al. [66] and to improve accuracy they used Recurrent Convolutional Neural Networks (RCNN). Their proposed model gained an accuracy of 99.87%.

Vinaya Kumar et al. [68] has proposed a new model named DeepSpamPhishEmailNet (DSPEN) over a deep Learning framework. They have collected email and URL datasets from publicly available datasets as well as privately. In the case of a private dataset, they have collected samples and manually assigned labels. Keras embedding was used in the word embedding layer. They achieved 65.3% accuracy for email classification and 98.4% accuracy for URL classification in the CNN-LSTM model.

Hiransha M et al. [69] proposed CEN-Deepspam for phishing email detection. Their dataset consists of email texts with headers and only email texts. Keras Word Embedding has been used for their proposed model. They used Convolutional Neural Network (CNN) for classification. They achieved 96.8% accuracy without a header and 94.2% accuracy with a header.

Chetty et al. [70] classified phishing emails using a deep learning model. They have collected three different datasets namely, UCI email spam data, UCI SMS spam data, and UCI YouTube spam data. They performed different preprocessing for cleaning the text. They achieved 92.8% with word embedding neural network.

From the above, it can be concluded that deep learning models are extensively used for phishing email detection. New models are proposed to improve the accuracy of phishing email detection. But most of the researchers evaluate the accuracy of the proposed model using different data sets separately. But these data sets are not robust. These data sets include 1000 to 5000 emails. In our research, we merged five of these data sets to form a single data set that contains approximately 23000 emails. This merged data set is used to evaluate the performance of two proposed models.

5.3 Materials and Methods

The steps of the proposed work for phishing email detection are illustrated in Fig. 5.1. First, the email data sets are preprocessed to perform lower casing of the words in the emails, tokenization, lemmatization, removal of non-alphabetic tokens, and stop words. Next two different models are used separately to detect whether an email is phishing or not. Finally, the performances of these two models are evaluated and compared.

5.3.1 Pre-processing of email content

Initially, the content of every email in the entire database is converted into lowercase. In the next step, lowercase email is tokenized. In this work, we have used 1-gram tokenization which means every token consists of only one word. After tokenization, all non-alphabetic tokens are removed which consist of numbers, special characters, etc. After that lemmatization is performed which means tokens are transformed into their base form. Finally, we removed stop words like ‘a’, ‘an’, ’the’ etc. from emails.

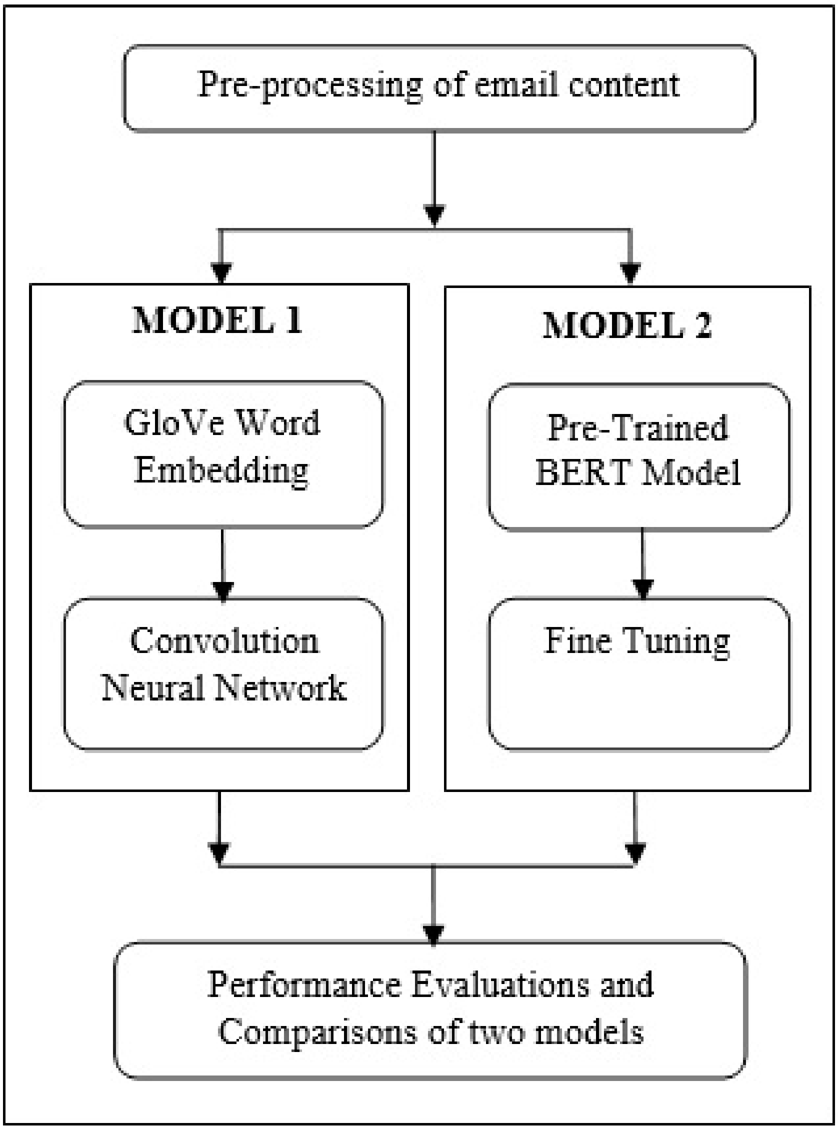

Pre-processing of email content
MODEL 1
GloVe Word Embedding
Convolution Neural Network
MODEL 2
Pre-Trained BERT Model
Fine Tuning
Performance Evaluations and Comparisons of two models

Fig. 5.1 Proposed work for phishing email detection.

After pre-processing, the pre-processed emails are passed into two different models to detect phishing emails. Model 1 uses CNN with GloVe word embedding whereas Model 2 uses a pre-trained BERT model with fine-tuning.

5.3.2 Model 1

In Model 1, initially, GloVe word embedding is used to find the context information from the pre-processed emails. After that, this context information is used to train a convolution neural network to classify an email as phishing or not.

GloVe word embedding:

In the word embedding technique, every word is represented by a numeric value by considering the context of a word in a document. The words with similar meanings are represented by a similar numeric value. In this work, we have used GloVe word embedding model. The full form of GloVe is Global Vector for word representation, proposed by Stanford University [118]. GloVe word embedding considers the global context of the word rather than considering only local meaning. To determine the context, a co-occurrence matrix is prepared. This matrix represents which pair of words has more likelihood of occurrence than other pairs.

Convolution Neural Network:

The proposed CNN architecture consists of an input layer and an output layer with hidden layers between them. The architecture is shown in Fig. 5.2. The hidden layer of the proposed CNN model consists of two 1-dimensional convolution layers with filter size 32, kernel size 3, and swish activation function. The Max Pooling layer consists of pool size 3. After the second convolution layer, the Global Max Pooling layer has been used, because it downsamples the input representation by taking the maximum values over time. A fully connected dense layer of 128 units with a swish activation function is used. A dropout layer is used to prevent overfitting with a rate of 0.3. The output layer consists sigmoid function for binary classification to detect whether an email is phishing or not. The weights of the CNN network were randomly initialized.

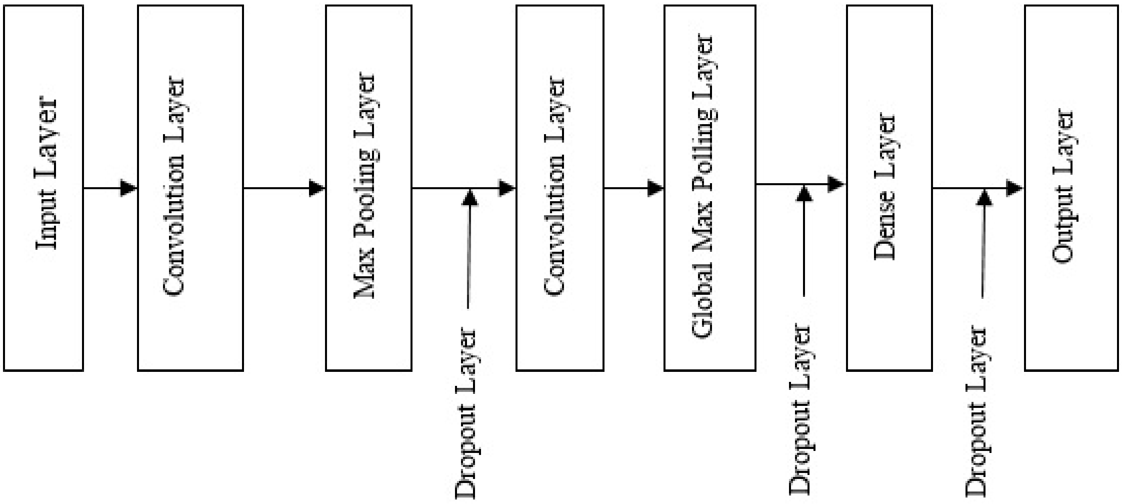


Fig. 5.2 Proposed CNN Model.

5.3.3 Model 2

In Model 2, after pre-processing, emails are passed to the pre-trained Bidirectional Encoder Representations from Transformers (BERT) model. After that, the output of the model is fine-tuned using a fully connected neural network for phishing email detection.

Pre-trained BERT Model:

Within the BERT model, initially pre-processed emails are tokenized in words, and tokens are represented by a numeric value. After that, the sentences of the emails are represented by numeric values of the tokens and applied in the transformer. Our proposed model consists of twelve transformer blocks; a hidden size of seven hundred sixty-eight dimensions and twelve attention heads. The BERT model produces a vector representation of each input sequence and a vector representation of the input token context within the sentences. Within these vector representations of tokens, the classification token (CLS) holds the combined context information of all the input tokens. This classification token vector is used for the classification of email using a neural network.

Fine tuning:

To detect whether an email is phishing email or not we use a fully connected neural network. The network consists of an input layer of 768 nodes, two hidden layers with 256 and 64 nodes

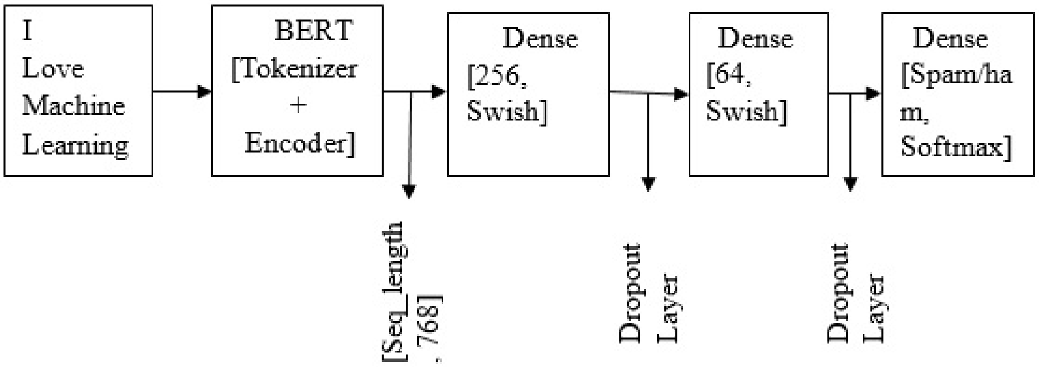


Fig. 5.3 Proposed BERT model with fine-tuning.

respectively and an output layer with single node. In the hidden layer swish activation function is used. The network is trained by the classification token vector of the BERT model. The Classification token vector consists of context information of the words within the email. The output node uses a sigmoid activation function for binary classification to detect whether an email is phishing or not. To prevent overfitting, dropout layers are used with a rate of 0.2. Figure 5.3 shows the proposed BERT model. For compiling the model we have used Adam optimizer due to its computational and space efficiency with a learning rate of 0.001 along with binary cross entropy loss. The training validation set in fixed-length data is used to train the classification model with 10 epochs and 64 batch sizes.

5.3.4 Performance Evaluations

To estimate the performance of these two models for phishing email detection, several test matrices are used.

- ***Confusion Matrix:***It is a tabular format for visualizing the performance of the proposed model; it can be expressed as in Table 5.1.

  Where,

  – TN (True Negative): It measures the number of legitimate emails classified as legitimate.

  – FP (False Positive): It measures the number of legitimate emails classified as Phishing.

  – FN (False Negative):It measures the number of Phishing emails classified as Legitimate.

  – TP (True positive): It measures the number of Phishing emails classified as Phishing.

Table 5.1 Classification of Confusion Matrix.

		Predicted	
		Legitimate	**Phishing**
Actual	**Legitimate**	TN	FP
	Phishing	FN	TP

- ***Accuracy:*** It measures the overall rate of correct prediction. It is defined by Eq. 5.1

$$Accuracy = \frac{(TP+TN)}{(TP+FP+TN+FN)} \tag{5.1}$$

- ***Precision:*** It measures the rate of instances correctly detected as phishing emails with respect to all instances detected as phishing. It is defined by Eq. 5.2.

$$Precision = \frac{TP}{(TP+FP)} \tag{5.2}$$

- ***Recall or True Positive Rate (TPR):*** It measures the proportion of phishing emails that are identified correctly with respect to the total number of phishing emails, defined by Eq. 5.3.

$$Recall = \frac{TP}{(TP+FN)} \tag{5.3}$$

- ***f1 Score:*** It is the harmonic mean of Precision and Recall. It is defined by 5.4.

$$f1Score = \frac{(2*Precision*Recall)}{(Precision+Recall)} \tag{5.4}$$

The performances of these two models are compared with respect to the accuracy, precision, recall, and f1-score. The ROC curves are also drawn for these two models and ROC scores are compared.

5.4 Results and Discussions

To perform the experiment, five publicly available data sets are merged together to form a single data set. These five data sets with the number of spam and ham emails within these data sets are listed in Table 5.2. It can be observed from Table 5.2, each data set contains 1000 to 5000 emails which is very small to evaluate the performance of a machine learning model. Thus to increase the size of the data set, these five data sets are merged together. The merged data set consists of 22965 emails. Out of 22965 data instances, 15502 emails are legitimate emails and 7463 emails are phishing emails. To maintain uniformity and ease of calculation, we label the phishing email as 1 and the legitimate email as 0 in the merged data set. The merged data set is splited into 80% and 20% for training and validation respectively. The validation set consists of 4593 emails, of which 3111 are legitimate and 1482 are phishing emails.

Table 5.2 Dataset collection from different websites.

Datasets	**Spam Email**	**Ham Email**
lingSpam [142]	423	2168
enronSpamSubset [142]	4760	4927
completeSpamAssassin [142]	1378	3915
Jose Nazario's phishing dataset [143]	902	-
Enron email dataset [144]	-	4492
Total:	7463	15502

The confusion matrix over validation data sets for two models is calculated and shown in Table 5.3. In the case of Model 1, out of 3111 legitimate data instances, 3051 are correctly classified and 1441 phishing data instances are correctly predicted out of 1482 phishing data instances. On the other hand in Model 2, 3013 legitimate data instances and 1409 phishing emails are predicted correctly.

Table 5.3 Confusion matrix of two models.

		Predicted			
		Legitimate	**Phishing**	**Legitimate**	**Phishing**
Actual	**Legitimate**	3051	60	3013	98
	Phishing	41	1441	73	1409
		Model 1		**Model 2**	

After finding the confusion matrix in both models, the precision, recall, and f1 score of each class are measured using Eq. 5.2, 5.3, and 5.4 respectively. The values are shown in Table 5.4. It can be observed that in Model 1, f1-score for legitimate and phishing emails is 0.98 and 0.97 respectively whereas in the BERT model f1-score is 0.97 and 0.94 respectively.

Table 5.4 Performance of two models.

Model	Class	Precision	Recall	f1-Score
Model 1	**Legitimate**	0.99	0.98	0.98
	Phishing	0.96	0.97	0.97
Model 2	**Legitimate**	0.98	0.97	0.97
	Phishing	0.93	0.95	0.94

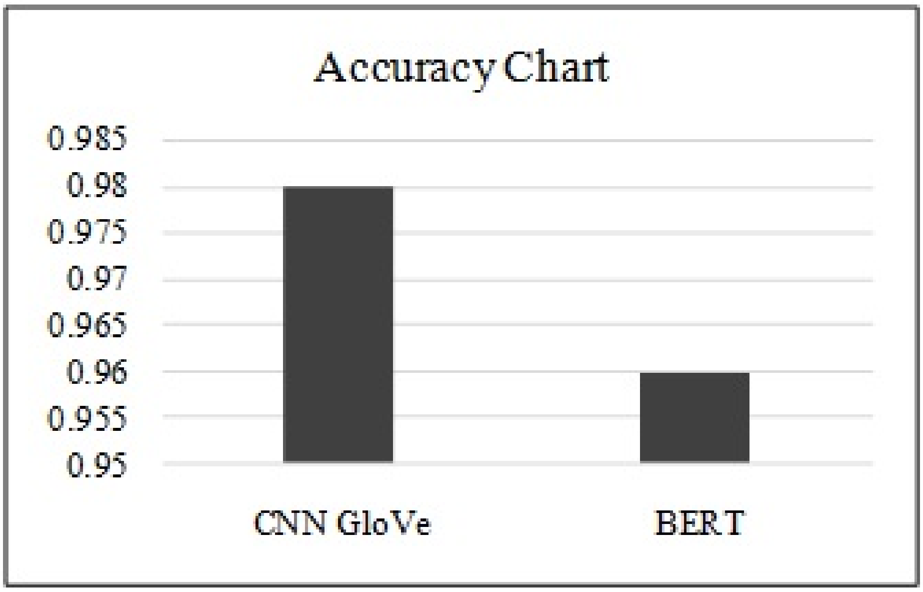


Fig. 5.4 Overall Accuracy of two models.

The overall accuracy of the two models is shown in Fig. 5.4. Model 1 achieves 98% accuracy whereas Model 2 achieves 96%. The ROC curves for both the models are shown in Fig. 5.5 and Fig. 5.6. Model 1 and Model 2 achieve approximately 1 and 0.96 ROC-score respectively.

Though the BERT model is superior to Glove, the performance of Model 1 outperforms the performance of Model 2. In this experiment, the same text pre-processing technique has been applied to the email data set before the application of both models. After pre-processing, the total number of tokenized words is 1912929. These tokens are applied to the input of both models. In the case of Model 1, which consists of CNN with GloVe word embedding, we have considered output dimension 300 which is randomly chosen, and the maximum length of emails is 19000 words. But, since the BERT model is predefined, an output dimension of 768 is considered and the maximum length of an email is 512 words in Model 2. In Model 2, If the length of the email is more than 512 words, the

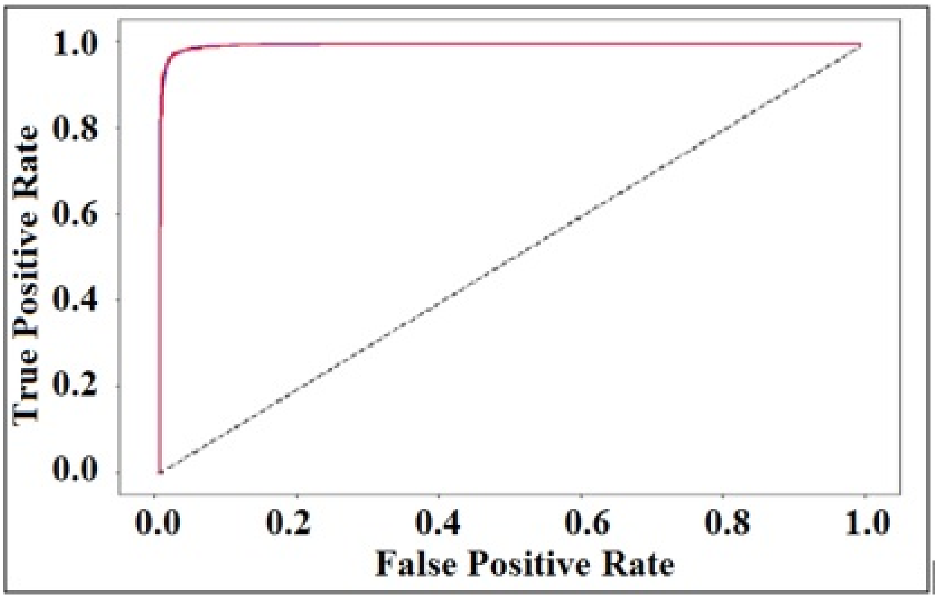


Fig. 5.5 ROC Curve for Model 1.

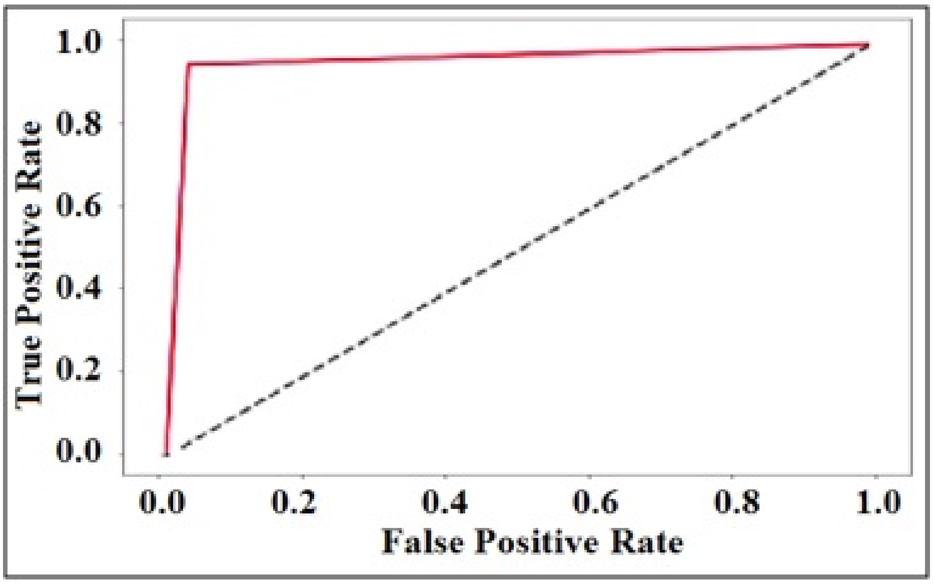


Fig. 5.6 ROC Curve for Model 2.

first 512 words are considered to classify the email as phishing or not. Thus, it can be concluded that Model 1 trained more features than Model 2 and hence Model 1 achieves better accuracy than Model 2.

5.5 Summary

In this chapter, we have used two different models for the classification of phishing emails. Due to the lack of a large data set, five data sets have been merged together to evaluate the performance of the proposed models. The emails in the data set are pre-processed before application to these two models. Then pre-processed emails are classified using two models. Model 1, which consists of the CNN model with GloVe word embedding, achieves 98% accuracy. But Model 2, which uses the BERT model, achieves a 96% accuracy score. The limitation of the BERT model is that it takes a maximum sequence length of 512, which means the first 512 words of an email are considered to classify phishing emails. But emails may contain more than 512 words. In this case, several number of the BERT model is required, which increase the training and classification time. In the future, we will enhance the model to overcome the limitation of BERT by splitting large email text into smaller subtexts and will apply the BERT model for each of these subtexts. Finally, the outputs of subtexts will be combined for the classification of phishing and legitimate email. But in this case, time requirement should also be measured to justify the application of this model in a real-time context.

Chapter 6

Phishing Website Detection

6.1 Introduction

Phishing is a cyber-crime by which the attackers steal credential information via email, telephone, and text messages. The attackers trick the users using psychological manipulation to make security mistakes or to give away sensitive information. In recent times, phishing is the fastest-growing cyber attack. In a recent survey in April 2021, it is witnessed that internet users around the world are almost 4.72 billion. Currently, the annual growth rate of internet users is 7.6% [145]. As the number of internet users is increasing, phishing is also increasing rapidly. According to the Anti-Phishing Working Group (APWG) report, 1st, 2nd, 3rd, and 4th Quarter of 2020 [146] number of unique phishing websites detected from January 2020 to December 2020 are 54926, 49560, 60286, 48951, 52007, 46036, 171040, 201591, 199133, 225304, 212878 and 199120 respectively. Figure 6.1 represents the total number of phishing websites in different months of the year 2020 according to the APWG report. It is observed that from July 2020 onward the number of phishing websites has increased rapidly. Therefore an effective technique for detecting phishing websites is urgently required.

Phishing website detection can be classified into IF-THEN-ELSE form that is either it is "phishing websites" or it is "legitimate websites". As attacker relies on the unawareness of using network tools of internet users, it is difficult to solve phishing attacks permanently. It is a better idea to train or literate end users by means of sending messages to make them aware of phishing. Many approaches are proposed by different researchers for this purpose ([147, 148]).

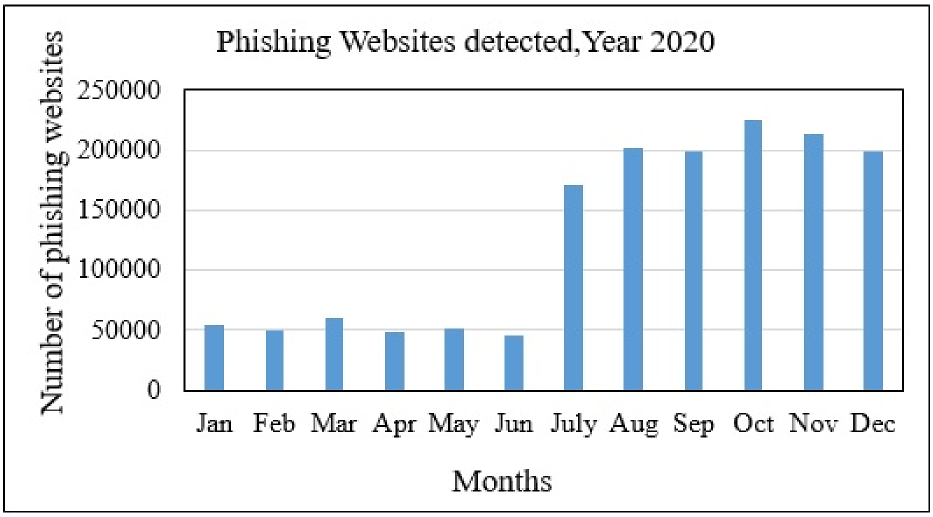


Fig. 6.1 Phishing websites detected in the year, 2020.

Different approaches are present for detecting phishing websites automatically, such as the black list and white list approach, heuristic-based approach, content-based approach, visual similarity-based approach, and machine learning-based approach. In the black list and white list approach, malicious URLs or IP addresses are previously stored in a database. Google safe browsing and Phish Tank are used to store blacklist websites. The main drawback of this approach is the lack of ability to detect newly generated suspicious URLs. In the heuristic approach [71], by scanning the web page of known attacks, a signature database is created. When a user submits an URL, this approach matches it with the signature database and warns the user if any malicious behaviour is found. The drawback of this approach is that, it is time-consuming and attackers easily bypass it through obfuscation, which is widely used by malware writers. It is a technique by which understanding programs is made difficult [149]. In the content-based approach [72] detailed analysis of page content is required. Features are extracted from that content and a classifier is built to detect phishing. The drawback of this approach is that it relies on third-party services such as search engines, DNS servers, etc. In the visual similarity approach [73] phishing websites look similar to legitimate websites by embedding objects like images, scripts, etc. In this approach stored snapshots of different legitimate websites are compared with newly generated websites. The drawback of this approach is time and space consumption. In the Machine learning approach, first, the important features are selected from data sets, and then different machine learning models like Random Forest (RF), Naïve Bayes, Support Vector Machine (SVM), etc. are applied for classifying phishing or legitimate websites. The machine learning approach can detect phishing website which is newly generated. Improper selections of features are not capable of accurately detecting phishing websites.

A new approach known as the Ensemble learning model is also used by some researchers for the detection of phishing websites. In machine learning models a single model like Logistic regression

(LR), Decision Tree (DT), Artificial Neural Network (ANN), and SVM are used for classification purposes. But Ensemble methods are machine learning techniques where predictive models are generated by combining multiple base models. In most of the research, it has been found that the performance of the ensemble predictive methods gives better accuracy compared to individual machine learning algorithms [79–81]. "The law of diminishing returns in ensemble construction" states that there is an ideal number of component classifiers for an ensemble that gives better performance; more or less of that ideal number would deteriorate the accuracy [150, 151].

Ensemble learning can be classified into three categories, namely Bagging, Boosting and Stacking. Bagging is one of the simple and strong ensemble methods. In this method training data set is divided into several samples by randomly selecting data from the training data set with replacement. After that, each sample is used to train each base model. For the classification problems, the outputs of all models are aggregated and by voting a single output is created. For the regression problem, the outputs of all models are averaged to get the final result. The model is stable because it minimizes the variance and bias in data distribution [152]. Boosting is an ensemble method that is able to convert a set of weak classifiers into strong classifiers [153]. The predictors are trained sequentially. The first predictor model learns from whole training data sets and the next predictor model learns the performance of previous learners and so on. In each iteration, weight should be increased for each instance which is wrongly classified by previous learners. For the classification problem, the final prediction is considered by combining all predictor's output along with a weighted majority vote. For regression problems, the final prediction is considered by the weighted sum of all predictors' output.

The Bagging and Boosting ensemble method is based on voting but in Stacking, lower-level base learners are combined to produce high-level learners. Wolpert observed that the performance of Stacking methods achieved high accuracy than a single-trained model [154]. In the case of the Stacking model, initially, individual models are called the base model or level-0 model. Each base model is trained by the whole training data set. After that, the Level-1 model or meta-model is used to merge multiple base models to find the best combination of prediction to achieve the best performance. It can be used for both supervised learning models (Regression [152], and classification [155]) and unsupervised learning models [156].

In our experiment, an ensemble model has been applied to detect the phishing website. In this method, the important features have been selected from a given URL by applying the Random Forest Regressor model. After extracting features, seven machine learning models namely Naïve Bayes (NB), k-Nearest Neighbour (k-NN), Support Vector Machine (SVM), Random forest (RF), Bagging, Logistic Regression (LR) and Artificial Neural Network (ANN) are used as base models to evaluate the performance of these individual base models. Finally, using the optimal stacking model with a greedy approach, we have aggregated individual models one by one to get better accuracy.

6.2 Past Works

Since the number of new phishing websites is increasing rapidly, many researchers developed different methods for detecting phishing attacks. In this section, some recent machine learning-based approaches for phishing website detection are presented. These machine learning-based approaches perform the task in two steps. Initially, optimal features are extracted from data sets and then the extracted features are used to train a different supervised base model to measure the performance of the model. In the case of ensemble models, the results of base models are combined to increase the accuracy of the proposed model. These machine learning-based approaches mainly differ in the methods that have been used for feature selection, the number of features considered and the base models used for classification. For ensemble approaches the methods used to combine the results of base models may also differ. But most of the researchers rely on the voting method for combining the results of the base model.

Ubing et al. [83] used Random Forest Regressor (RFG) technique for feature selection and considered nine features based on the feature importance. They applied ensemble learning by combining multiple base models namely, Gaussian Naive Bayes, Support Vector Machine, K-Nearest Neighbour, Logistic Regression, Multilayer perceptron NN, Gradient boosting and Random Forest classifiers to improve the accuracy for phishing website detection. They have collected data set from UCI Machine Learning Repository which is publicly available. After applying ensemble learning approach accuracy obtained was 95.4%.

A novel twofold ensemble model was used by Nagaraj et al. [82] for detection of phishing websites. They have collected there data sets from UCI Machine learning Repository. Then used filter and wrapper method for features selection and collected six significant features from data set. They have applied three ensemble learning models - Random Forest Neural Network model (RF_NN), Bagging Neural Network ensemble model (Bagging_NN) and Boosting Neural Network ensemble model (Boosting_NN) on the retrieved important features. The experimental results show that the RF_NN model perform best with 93.41% accuracy score and 0.000026 mean squared error.

Subasi et al. [74] used Random forest (RF) and Rotation Forest (RoF) ensemble approach to learning over publicly available phishing websites data set collected from the UCI machine learning repository. For testing the model performance they used WEKA machine learning tool. They also used k-Nearest neighbour (k-NN), Support Vector Machine (SVM), Artificial Neural Networks (ANN), Naïve Based (NB) machine learning approaches. According to them, the accuracy of the RF is not more than 97.26% and corresponding F1 score is 0.974.

A diverse machine learning approach was proposed by Zamir et al. [157] for detecting phishing websites. They have collected data set from the Kaggle website which consists of 32 attributes. Their feature selection methods include Information Gain (IG), Gain Ratio (GR), and Relief-F. They considered 28 features and applied them over two stacking ensemble models. The first stacking model consists of Random Forest (RF), Feed forward neural network, and bagging. The second staking model consists of k-Nearest Neighbour (k-NN), Random Forest (RF) and bagging. The first stacking model achieved better accuracy (97.4%) than the second model.

Kamal et al. [158] proposed the use of machine learning for phishing website detection. The features are extracted from the URL only and the Naïve Bayes algorithm is used for the classification of phishing websites. Using the ensemble methodology, an accuracy of 97.08% was achieved using Stacking, Bagging, and Boosting along with the Naive Bayes, Decision Tree and Random Forest algorithm.

An optimized stacking ensemble model for phishing websites detection was proposed by Al-Sarem et al. [159]. They have collected their data sets, one from UCI and the remaining two from Mendeley website. The first data set consists of 30 features, and the remaining two consist of 48 and 111 features. Their proposed model includes three phases, namely the training, ranking, and testing phase. In the training phase, they applied different ensemble methods which included Random Forests, AdaBoost, XGBoost, Bagging, Gradient Boost, and LightGBM. In the ranking phase, they considered the best three ensemble methods GA–Gradient Boost, GA–XGBoost, and GA–Bagging. Finally, in the testing phase, they collected new websites and tested whether it is phishing or legitimate. They achieved 97.16% accuracy in their experiment using GA–Gradient Boost ensemble model.

Ensemble based logistic model trees for phishing website detection were proposed by Adeyemo et al. [160]. They have collected two different data sets from the UCI machine learning repository. They proposed Ensemble based Logistic Model Trees (LMT) which is a combination of logistic regression and tree induction methods into a single model tree. They proposed an Adaptive Boost Logistic Model Tree (ABLMT) which used weighted averages to amplify the performance of LMT algorithm by iteratively selecting features for building models and Bagging Logistic Model Tree (BGLMT) which is the integration of the LMT in a Bootstrap Aggregating (Bagging) ensemble method. They achieved 97.18% accuracy for both Adaptive Boost Logistic Model Tree (ABLMT) and Bagging Logistic Model Tree (BGLMT) ensemble methods.

Akanbi et al. [161] proposed a machine learning approach for phishing website detection. They have collected their data set from Phish Tank websites. They used a gain ratio algorithm for featured selection and collected 17 important features. Their proposed approach is divided into three phases. In first phase, pruning decision trees have been proposed for classifying phishing websites. In second phase, they created four ensemble learning methods. First ensemble model consists of k-nearest

Neighbour (KNN), Decision Tree C4.5 and Linear Regression (LR). Second ensemble model consists of k-nearest Neighbour (KNN), Decision Tree C4.5 and Support Vector Machine (SVM). The third ensemble model consists of k-nearest Neighbour (KNN), Linear Regression (LR) and Support Vector Machine (SVM). Fourth ensemble model consists of Decision Tree C4.5, Linear Regression (LR) and Support Vector Machine (SVM). In the final phase, they evaluated the performance of each proposed method. Finally, they conclude that pruning decision trees is comparatively potent in website phishing detection with an accuracy score of 99.71%.

Phishing website detection based on an effective machine learning approach was proposed by Lokesh et al. [162]. They have collected data sets from the Miller Smiles archive and Phish Tank archive which are extracted using data mining algorithms. They discussed important features by using the Random Forest classifier method. They applied different machine learning classifiers like Random Forest classifier, Decision tree classifier, K nearest neighbours, Linear SVC classifier, and one class SVM classifier. Finally, a Random forest ensemble classifier was applied to this data set with the values of n_estimators as 500, max_depth as 15, and max_leaf_nodes = 10,000 and obtained the highest accuracy rate of 96.87%.

Forest by Penalizing Attributes (PA) algorithm and its enhanced variations for phishing websites detection were proposed by Alsariera et al. [163]. They collected data sets from the UCI machine learning repository. They proposed three ensemble approaches namely Forest PA, Bagged-Forest PA, and AdaB-Forest PA. Finally, they achieved the highest accuracy rate of 97.4% after applying AdaB-Forest PA.

From these above mentioned researches, it can be concluded that ensemble model is very important for phishing website detection. These ensemble techniques mainly differ in the feature selection procedure and the base models that are used in these techniques. Depending on the strong feature selection methodology, accuracy can be enhanced. As well as accuracy also depends on the ensemble technique itself. In this paper, we have developed a novel approach that uses an ensemble stacking model for phishing website detection.

6.3 Materials and methods

The methodology for the detection of phishing websites is illustrated in Figure 6.2. The broad steps of this method are

- Selection of important features from website URL

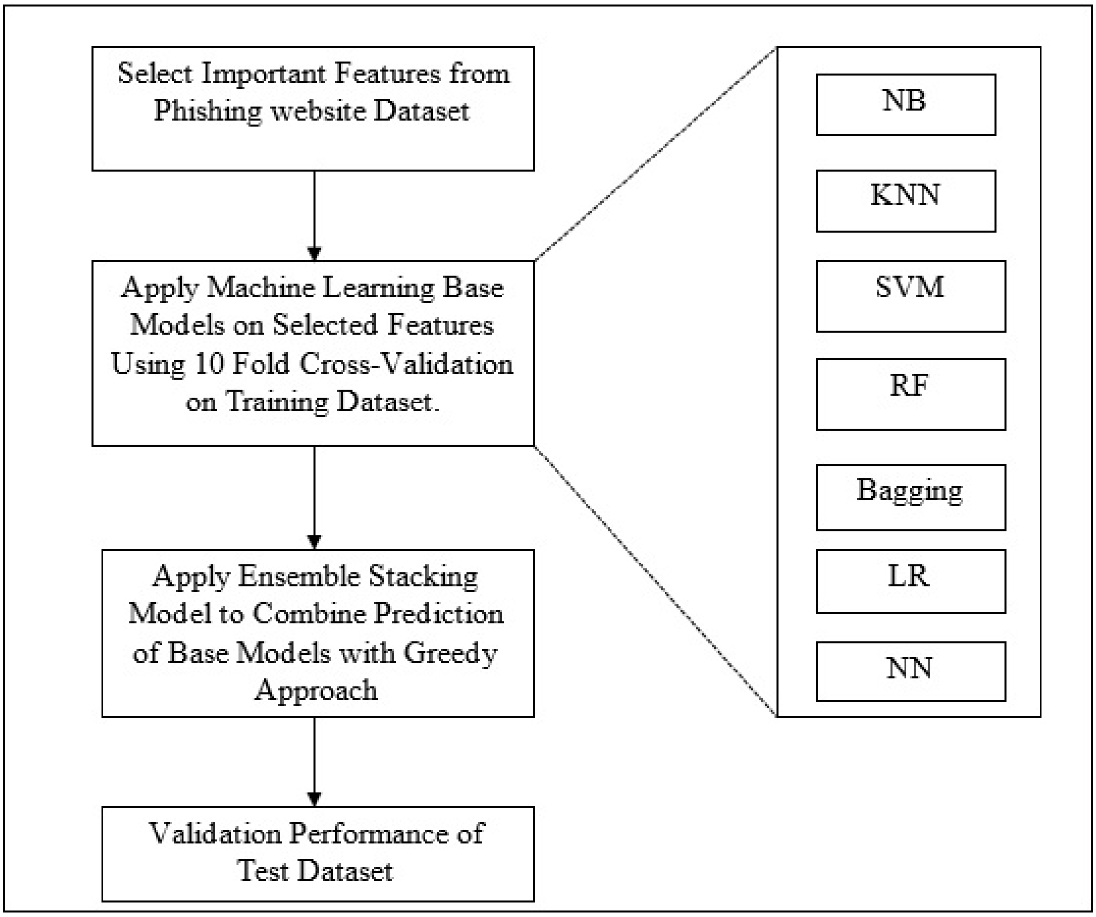

Select Important Features from Phishing website Dataset
Apply Machine Learning Base Models on Selected Features Using 10 Fold Cross-Validation on Training Dataset.
Apply Ensemble Stacking Model to Combine Prediction of Base Models with Greedy Approach
Validation Performance of Test Dataset
NB
KNN
SVM
RF
Bagging
LR
NN

Fig. 6.2 Proposed model for the detection of phishing websites.

- Application of machine learning-based models to classify the website as phishing or legitimate
- Integrating the base models using Ensemble learning.

6.3.1 Selection of important features of the website

In this work, we have used UCI Machine Learning Repository [164] data set which includes 11,055 websites with 30 web-based parameters, and also phishing and non-phishing classes are present. Phishing and non-Phishing classes include 6157 and 4898 instances respectively. This data set consists of thirty features with Address bar, Abnormal, HTML and JavaScript, and Domain features.

A set of rules have been framed to categorize each website into three classes, namely phishing, suspicious, and legitimate depending on these 30 features are divided into 4 parts -

1. Address bar features, which consist of 12 features shown in Table 6.1
2. Abnormal features consist of 6 features shown in Table 6.2
3. HTML and Java Script features consist of 5 features shown in Table 6.3
4. Domain features consist of 7 features shown in table 6.4.

The inappropriate category is marked cross with (“X”) in the table. The classes are assigned unique labels indicating 2 for phishing, 0 for suspicious and 1 for legitimate website.

Feature selection is one of the important parts of the machine learning model. In this experiment, important features are extracted using the Random Forest regression model [165] which is as follows:

First, any random feature is selected from 30 features present in the data set. For this feature, a node is created which consists of five fields namely feature name, criterion, number of samples in the training dataset, mean, and Mean Squared Error (MSE). The labels are assigned for the samples that belong to this node depending on the feature of that node. Criterion is set as the average of assigned class labels of the samples belonging to this node. After the creation of the node, it is checked how many samples in this node have assigned class labels less than or equal to the value assigned in the criteria. Thus the node may have two children depending on whether the condition is satisfied or not. If the condition is satisfied, a new node is created with any other arbitrary feature that is not already considered. The number of samples field is set as the number of samples satisfying the condition. The same step is also performed if the condition is not satisfied. In each step mean and MSE is calculated by Eq. 6.1 and Eq. 6.2.

Table 6.1 Address bar features.

	Features	**Categories**		
		Phishing	**Suspicious**	**Legitimate**
Address bar features	IP Address	If the IP address is present in the Domain portion.	X	Not present.
	Length of the URL	The length of the URL is greater than 75	Between 54 and 75	Less than 54
	Shortening Service	URL is TINY	X	Not TINY
	@ symbol	URL consists @ symbol	X	Not present
	Redirect symbol	The last “//” position is greater than 7	X	Less than or equal to 7
	Prefix or Suffix	Domain portion consists (-) Symbol	X	Not consists (-) Symbol
	Sub Domain	The total of (.) in the Domain portion is greater than 2	Exactly 2	Exactly 1
	HTTPS with a trusted certificate	If it does not satisfy the rules of suspicious and legitimate	The user using https but is not trusted.	Age of trusted certificate greater than equal to 1.
	Length of Domain Registration	Domains Expiry is less than equal to 1 year	X	More than 1 year
	Favicon	Graphics image loaded from outside Domain	X	Inside Domain
	Nonstandard Port	Port number is of favourite Status	X	Not favourite Status
	HTTP	HTTP tokenpresent in the Domain part	X	Not present

$$Mean(\mu) = \frac{1}{n}\sum_{i=1}^{n} y_i \tag{6.1}$$

Table 6.2 Abnormal Features.

	Features	Categories		
		Phishing	**Suspicious**	**Legitimate**
Abnormal Features	External Objects present in URL	Request URL consists of more than 61% of external objects.	Between 22% and 61%	Less than 22%
	Anchor present in URL	URL consists Anchor more than 67%	Among 31% and 67%	Less than 31%
	Meta, link and script tag	More than 81%	Between 17% and 81%	Less than 17%
	Server from handler	Consists of "about blank" or empty	Mentions to a different Domain	If it does not satisfy the rules of suspicious and phishing.
	Email submitting	Submit user data using "mailto()".	X	Not present
	Abnormal URL	URL does not mention the hostname.	X	Mention hostname

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \mu)^2 \tag{6.2}$$

where n is the total number of samples, and y_i is the label of each sample. Figure 6.3 shows a part of a decision tree as an example. This process continues until all the samples in a node have the same class label.

Several trees are created using the same technique assigning different features in the root node and assigning different arbitrary features at each level also. After the creation of several decision trees randomly i.e. Random Forest, the Gini impurity of each node is calculated by Eq. 6.3

$$GV_i = 1 - (P_i^2 + Q_i^2 + R_i^2) \tag{6.3}$$

Table 6.3 HTML and Java Script Features.

	Features	**Categories**		
		Phishing	**Suspicious**	**Legitimate**
HTML and Java Script Features	Forwarding	Number of forwarding websites more than 4	Among 2 and 4	Less than equal to 1
	Customization of the status bar	True using on Mouse Over event	X	False
	Disabling Right Click	True	X	False
	Popup Window consists text field	True	X	False
	Redirection using IFrame	True	X	False

where GV_i is the Gini impurity value of node i. P_i indicates the ratio between the number of samples with label 0 among the samples of node i and the number of samples in node i. Similarly, Q_i and R_i indicate the ratio for labels 1 and 2 respectively. After that, the importance of each node is calculated using the Gini impurity value by Eq. 6.4.

$$imp\_node_i = WNS_i * GV_i - WNS_{left(i)} * GV_{left(i)} - WNS_{right(i)} * GV_{right(i)} \tag{6.4}$$

where imp_node_i is the importance of node i. The Weighted Number of Samples of node i, i.e., WNS_i indicates the ratio between the number of samples in node i and the number of samples in the dataset. $WNS_{left(i)}$ and $WNS_{right(i)}$ represent the Weighted Number of Samples of the left and right child of node i.

The feature importance is calculated from the importance of the root node of decision trees in the Random Forest. Since the root node of different trees in the Random Forest may have the same feature, the feature importance of feature i (imp_fi) is calculated by averaging the importance of root nodes in the Radom Forest with feature name i. Then the computed feature importance is normalized between 0 and 1 by using Eq. 6.5.

Table 6.4 Domain Features.

	Features	Categories		
		Phishing	Suspicious	Legitimate
Domain Features	Domain age	Less than 6 months	X	Greater than 6 months
	Search DNS Record	Not Found	X	Found
	Alexa database Rank of a website	X	Greater than or equal to 100000	Less than 100000
	Page Rank	Less than 0.2	X	Greater than or equal 0.2
	Website indexed by Google	False	X	True
	Number of Links Directing to Page	Equal to 0	Between 1 and 2	More than 2
	Feature-based Statistical Report	IP or domain name present in Phish Tank	X	Not present

$$norm\_imp\_f_i = \frac{imp\_f_i}{\sum_{j \in allfeatures} imp\_f_i} \tag{6.5}$$

After finding the importance of individual features, important features are selected from 30 features present in the dataset. The feature with the highest importance score is the most important. The features with an importance score of more than 1% of the highest score are selected as important features.

To illustrate the feature calculation, consider the decision tree shown in Figure 6.3. The root node A consists of all the samples in the dataset i.e. 11055. The feature of the root node is considered "URL_of_Anchor" arbitrarily. Among these samples, 7700 samples are assigned label 0, 73 are assigned label 1 and 3282 samples are assigned label 2 depending on the feature "URL_of_Anchor". Thus the total of 7773 samples of node A satisfy the criteria and a new node is created for these samples as node B. Similarly, node C is created for the samples of node A which are not satisfying the criteria. The Gini impurity value of node A, will be calculated as follows:

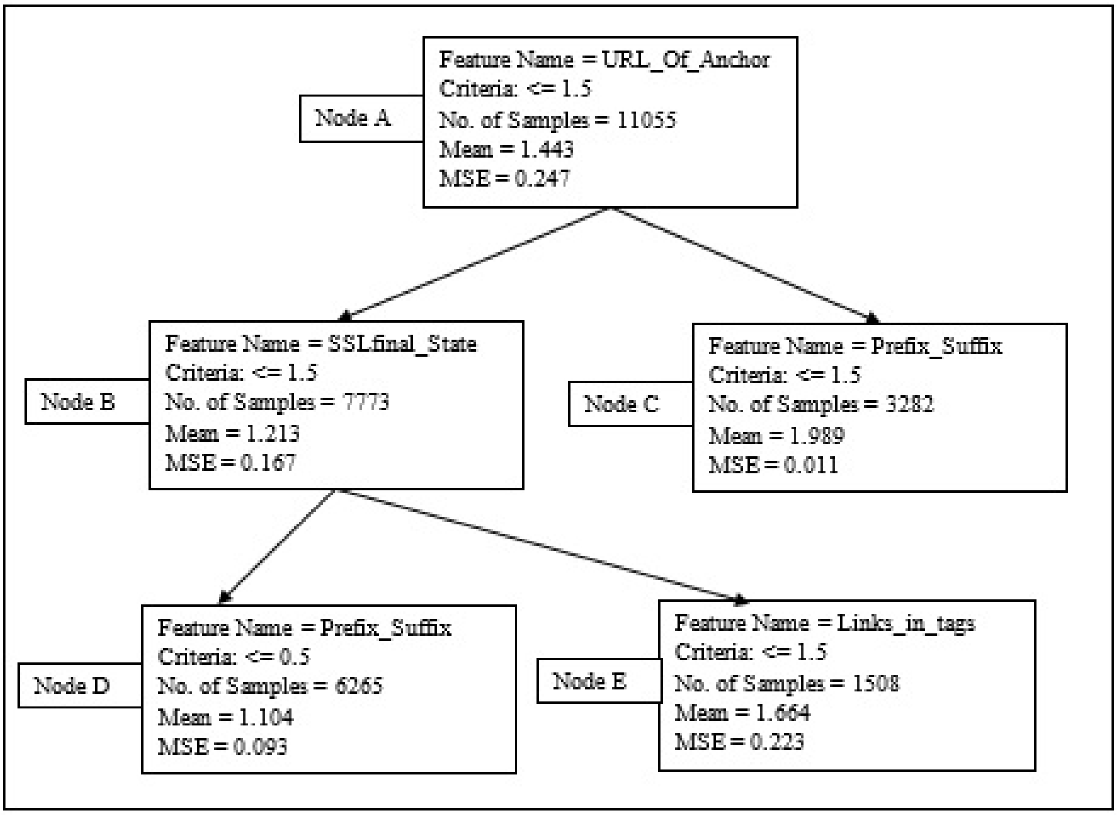


Fig. 6.3 Part of a decision tree in Random forest regression model.

P_A = No. of samples with label 0 in Node A/ No. of samples in Node A = 0.696

Q_A = No. of samples with label 1 in Node A/ No. of samples in Node A = 0.007

P_A = No. of samples with label 2 in Node A/ No. of samples in Node A = 0.297

Thus Gini Value of node A = 1 - (P_A^2 + Q_A^2 + R_A^2) = 1 - 0.573 = 0.427

Similarly assume that the Gini value of nodes B, C, D and E are 0.231, 0.352, 0.135 and 0.127 respectively. After that, the importance of each node is calculated using Equation (6.4). The evaluation of node importance for node B is shown as an example.

WNS_B = Number of samples in node B / Number of samples in the dataset = 0.703

$WNS_{left(B)}$ = Number of samples in node D / Number of samples in the dataset = 0.567

$WNS_{right(B)}$ = Number of samples in node E / Number of samples in the dataset = 0.136

Thus imp_node_B=$WNS_B * GV_B - WNS_{left(B)} * GV_{left(B)} - WNS_{right(B)} * GV_{right(B)}$

=0.703*0.231 – 0.567*0.135 – 0.136* 0.127 = 0.068

The importance of each node in the decision tree is calculated in a similar way. The importance of the root node is preserved for future use. The importance of each feature is calculated from the importance of the root nodes of all the decision trees in the Random Forest. Importance of feature i is calculated by averaging the importance of root nodes with feature_name i.

6.3.2 Application of machine learning-based models to classify the website as phishing or legitimate

Seven different supervised machine learning models are used as a base model over the phishing data sets for determining the phishing websites. These seven base models include Naïve Bayes (NB), K-Nearest Neighbour (K-NN), Support Vector Machine (SVM), Random Forest(RF), Bagging, Logistic Regression(LR) and Neural Network (NN). The K-fold cross-validation technique has been used to generalize the performance of individual models. In this experiment, 10-fold cross-validation is used, in which original data sets are divided into 10 equal size sub-samples. Each sub-sample is then used once for validation or testing purposes and the remaining 9 sub-samples for training purposes. After ten iterations, the final result is calculated by finding the mean of the results of all iterations. In this experiment, we have used true positive, false positive, true negative, false negative, f1 score, accuracy, precision, and recall to evaluate the performance of each base model.

6.3.3 Integrating the base models using Ensemble learning

After applying individual machine learning algorithms over phishing websites data sets, an ensemble learning model has been used to improve the performance. Most of the researchers rely on the voting method to combine the result of base models. But in this experiment ensemble stacking model with a greedy approach has been employed to integrate the result of the base models. The Greedy approach is an algorithmic paradigm that always chooses the best at that moment. To integrate the base models, the base model with the highest accuracy is considered first. After that, the base model with the next highest accuracy is integrated if the accuracy of the integrated model increases. This process continues until we find a base model whose inclusion decreases the accuracy. The algorithm for the integrating base model is given below.

Input: Results of base models.
Output: Integrated model that provides optimal accuracy.

```
1. Let n be the total number of base models.
2. Let [$ML_i$, $ACC_i$] be the different base model and their corresponding Accuracy, where i=1,2,..,n.
3. Sort accuracy ($ACC_i$) in non-increasing order.
4. Let STACK=$ML_1$ ( Initialize stack with the first ML whose accuracy is maximum)
5. Let OACC=$ACC_1$ (Initialize optimal accuracy[OACC] with first ML accuracy)
6. For i =2 to n
    (a) STACK = STACK + $ML_i$
    (b) Applying Stacking Classifier over STACK.
    (c) Applying 10-fold cross-validation.
    (d) Fit the model with our training data sets.
    (e) Predict the output of our testing data sets.
```

(f) Calculate TOALACC by summing each fold validation.

(g) Calculate MACC (mean accuracy) = TOTALACC/10.

(h) If MACC > OACC then Set OACC = MACC. Otherwise, exit from the loop.

End For

6.4 Results and Discussions

6.4.1 Selection of important features of the website

The UCI Machine Learning Repository has been used in this project to evaluate the performance of the proposed model. The data set contains 30 features. The feature importance score of these 30 features has been calculated using the Random Forest Regressor model and is listed in Table 6.5. The highest score represents the most important feature. Among these 30 features, 16 features are selected as important features since their score is more than 1% of the highest score.

Table 6.5 Thirty features with their feature importance scores.

SL. No.	Features	Score	SL. No.	Features	Score
1	URL_of_Anchor	0.54246	16	URL_Length	0.00657
2	SSLfinal_State	0.1947	17	Redirect	0.00394
3	Links_in_tags	0.03534	18	having_At_Symbol	0.00386
4	web_traffic	0.0329	19	Submitting_to_email	0.00366
5	having_Sub_Domain	0.02642	20	double_slash_redirecting	0.00344
6	Prefix_Suffix	0.02059	21	Statistical_report	0.00334
7	Links_pointing_to_page	0.017	22	popUpWidnow	0.00331
8	Request_URL	0.01284	23	HTTPS_token	0.00328
9	age_of_domain	0.01254	24	on_mouseover	0.00309
10	having_IP_Address	0.01038	25	Shortining_Service	0.00298
11	SFH	0.0103	26	Favicon	0.00247
12	Domain_registeration_length	0.00999	27	Abnormal_URL	0.00214
13	DNSRecord	0.00994	28	Iframe	0.00177
14	Google_Index	0.00945	29	RightClick	0.00149
15	Page_Rank	0.00873	30	port	0.00107

6.4.2 Application of machine learning-based models to classify the website as phishing or legitimate

The seven base models namely Naïve Bayes (NB), K-Nearest Neighbour (K-NN), Support Vector Machine (SVM), Random forest (RF), Bagging, Logistic Regression (LR) and Neural Network (NN) have been used to classify the website as phishing or legitimate based on the sixteen selected features. In each model, performance is measured by using evaluation parameters that include accuracy, precision, recall, and f1-score. The performance measures are listed in Table 6.6 for each of these seven base models. From Table 6.6 it can be observed that SVM, RF, bagging, and NN give an accuracy of over 96%, among these RF gives the maximum accuracy which is 96.59% with an f1-score of 96.95.

Table 6.6 Performance measures of seven base models.

Algorithm	**Accuracy**	**Precision**	**Recall**	**f1-score**
NB	92.57	92.11	94.78	93.42
KNN	95.10	94.92	96.37	95.64
SVM	96.11	95.99	97.05	96.52
RF	96.59	96.31	97.61	96.95
Bagging	96.30	96.14	97.26	96.70
LR	93.31	93.24	94.86	94.04
NN	96.50	96.24	97.52	96.87

6.4.3 Integrating the base models using Ensemble learning

After finding the performance measures on the seven base models, the optimal stacking model with a greedy approach is applied by combing higher accuracy models one by one. Table 6.7 displays the results of the stacking model. From Table 6.6, it can be observed that Random Forest (RF) achieved the highest accuracy 96.59% with an f1-score of 96.95. Thus this model is chosen initially. The second highest accuracy score (96.50%) was achieved by Neural Network (NN) with an f1-score of 96.87 among seven base models. We integrate these two classifiers in the proposed ensemble stacking model and observed that the accuracy score is 96.68% (First row in Table 6.7), which is more than the accuracy score of the individual classifier. After that, the next higher accuracy model is included in the ensemble stacking model till the accuracy increases. It can be observed from Table 6.7, combining the outcomes of RF, NN, Begging, SVM and KNN gives the highest accuracy (96.73%) with an f1 score of 97.07%. Since after including the next higher accuracy model (i.e. LR), the accuracy of the ensemble learning model decreases, the algorithm stops without considering the rest of the base models. Since our dataset is highly dimensional therefore after applying Logistic Regression, the performance of our proposed model is degraded. Furthermore, a theoretical framework [150, 151] suggested that there is an ideal number of component classifiers for an ensemble model, such that having more or less than this number of classifiers would deteriorate the accuracy. It is called "the law of diminishing returns in ensemble construction".

Table 6.7 Results of ensemble stacking model with a greedy approach.

SL. No.	Stacking	Accuracy	Precision	Recall	f1-score
1	RF + NN	96.68	96.54	97.51	97.02
2	RF + NN + Begging	96.70	96.56	97.60	97.04
3	RF + NN + Begging + SVM	96.71	96.56	97.54	97.04
4	RF + NN + Begging + SVM + KNN	**96.73**	96.58	97.58	**97.07**
5	RF + NN + Begging + SVM + KNN + LR	96.72	96.54	97.59	96.06

To measure the performance of the proposed ensemble model with a greedy approach, the accuracy is compared with other existing ensemble models. The accuracy comparisons are shown in Table 6.8. It can be observed from the table that the proposed algorithm achieves better accuracy than the methods proposed by other researchers.

6.5 Summery

The proposed method uses ensemble stacking models to detect whether a website is phishing or legitimate. The base models used by the proposed method are Naïve Bayes (NB), K-Nearest Neighbour (K-NN), Support Vector Machine (SVM), Random forest (RF), Bagging, Logistic Regression (LR) and Neural Network (NN). Random Forest Regressor model has been used to select 16 important features from the data set which includes 30 features. The accuracy of the proposed method is compared with existing ensemble learning models for phishing website detection. The main contribution of this paper is to apply a greedy approach to improve the accuracy of the ensemble stacking model. The proposed method achieved 96.73% accuracy with an f1 score of 97.07%. The study has been performed over the UCI Machine Learning Repository of phishing website data sets which is publicly available and easy to analyse. But in real-life scenarios, sensitive features of a phishing attack are continuously changing, so it is very necessary to assemble more features to achieve optimal feature selection.

Table 6.8 Performance comparisons.

Year of Publication	Authors	Model	Accuracy (%)
2016	Hadi et al. [166]	Associative classification algorithm	92.4
2017	Machado et al. [167]	c4.5 Decision Tree	89.4
2018	Nagaraj et al. [82]	Novel twofold ensemble model	93.4
2018	Sonmez et al. [168]	Extreme Learning Machine	95.34
2019	Ubing et al. [83]	Ensemble model	95.4
2019	Babagoli et al. [169]	Nonlinear regression technique and support vector machine (SVM)	92.8
2019	Zhu et al. [170]	Neural Network	96.44
2019	Chandra et al. [84]	Ensemble Model	92.72
2020	Alotaibi et al. [171]	Combination of the AdaBoost classifier and the LightGBM	95.29
2020	Folorunso et al. [85]	Stacking Model	95.97
2020	Folorunso et al. [85]	Stacking Model	96.04
2020	Alsariera et al. [163]	ForestPA-PWDM	96.26
2021	Altyeb et al. [172]	Weighted Soft Voting	95.0
2021	Dharani et al. [173]	Random Forest	91.1
2021	Dharani et al. [173]	XGBoost	93.8
	Proposed method	**Optimal Stacking model with a greedy approach**	**96.73**%

Chapter 7

Analysis of Cyber Crime Trends and its Prediction: A Case Study in India

7.1 Introduction

Nowadays the Internet has become a part of life and living. It became the backbone of the social and economic world. National Crime Records Bureau (NCRB) of India defines that cyber criminals perform cyber crimes for many reasons like to earn money, to become famous, to just have fun, to sexually exploit someone, to blackmail someone, to develop of own business, for illegal contents selling or purchasing, to take a revenge of someone, or to do a prank with someone. Economic growth and quality of life are affected by to occurrence of crime [5]. India ranks 3rd in terms of the highest number of internet users in the world after the USA and China and the number has grown 6-fold between 2012-2017 with a compound annual growth rate of 44% (Joint Chiefs of Staff 2018). Crime means anything that violates the law. In this context, cyber crime can be defined by any criminal activity done by using a computer or with the help of a computer (actively or passively) or any electronic devices, more specifically with the help of the internet. Different types of cyber crimes exist in literature like Computer integrity in cyber crime, Computer assisted cyber crime, Computer content cyber crimes, Violent or potentially violent cyber crime, Cyber terrorism, Assault Threats, etc.

The Government of India enacted its Information Technology Act 2000 with the objective to provide legal recognition for transactions carried out by means of electronic data interchange and other means of electronic communication, commonly known as E-commerce [174]. Table 7.1 describes different sections of the IT Act 2000 (The Information Technology Act, 2000).

Table 7.1 Different Section in IT acts 2000.

Offences	Section
Tampering with computer source documents	65
Hacking with computer system	66
Obscene publication or transmission in electronic form	67
Failure of compliance or orders of certifying authority	68
Failure to assist in decrypting the information intercepted by government agency	69
Unauthorized access or attempt to access to protected computer system	70
Obtaining license or digital signature certificate by misrepresentation or suppression of fact	71
Breach of confidentiality or privacy	72
Publishing false digital signature certificate	73
Fraud digital signature certificate	74

The scientific study for understanding the behaviour of crime, the nature of the crime, and deriving some anti-crime strategies by identifying the characteristics of crime is known as criminology. Crime analysis is a sub-branch of criminology. It studies the pattern of the crimes and tries to find indicators of the events. But the presence of a huge amount of data and due to increasing crime rate, it is impossible for security authority and their personnel to manually analyze these data and find the hidden secrets within these data ([101–103], [18]). The fact is that every incident holds a piece of valuable information that can be used for forecasting the occurrence of the incident in the future, i.e., the past is the true providence of the future [106]. Crime is also predictable since human nature is not reversible [107]. Human nature generally evokes periodically to perform the same event. With the advancement of big data and easy-to-implement algorithms for analysis of the data, the prediction of crime is a growing field of study. In this study, we analyze the cyber crime records for 28 States and 8 Union Territories of India, published by the National Crime Records Bureau during the year 2011 to 2018. The trends of cyber crimes in India as a nation and also the trends for individual states and union territories are observed in different sections under the IT Act 2000. These observations are used to predict the cyber crime events in India as well as individual states and territories for the years 2019 and 2020.

7.2 Past Works

Several researchers proposed different methods to analyze and evaluate the cyber crime offenses. Tsakalidis and Vergidis [86] presented the features of cyber crime incidents as a classification system for related offences and a schema that binds together the various elements. Their interrelations are measured for better understanding of corresponding actions taken and policies required. The process involved revision of reports conducted by authorities, academia and agencies related to information security. For better classification and correlation, they proposed a comprehensive list of cyber crime related offences which are ordered in a two-level classification system based on specific criteria.

Ganesan and Mayilvahanan [87] proposed a methodology that offered the discovery of unpredicted patterns. They analyzed the cyber crime data from the database, which is a collection of data fields from the Internet web pages. The data fields include cyber-bullying, stalking, scams, robbery, identity theft, defamation and harassment. They introduced this model in order to categorize cyber crime offenses such as whether they are violent or non-violent, and further, they can be categorized as various types of cyber crimes such as cyber terrorism, cyber stalking, pornography, cyber bullying, cyber fraud and cyber theft.

Five types of crime i.e. fraud detection, traffic violence, violent crime, web crime and sexual offense are studied by Prabakaran and Mitra [88]. They discussed how different machine learning models such as Hidden Markov Model (HMM), Naive Bayesian, Fuzzy c-means algorithm, Cumulative logistics model, K-means Clustering, Logistic regression etc can be used for fraud detection.

Kigerl [89] uses K-means clustering analysis on prior reporting attempting to rank nations into categories based on cyber crime output. Seven cyber crime variables are used to capture fraud, malware, spam, and digital piracy, as well as each nation's GDP and internet use per capita. Nations were assigned to one of four clusters, including low cyber crime countries with low GDP and internet connectivity, advance fee fraud specialist nations, with modestly low connectivity; non-serious cyber crime nations, high in piracy and email spam and that were the wealthiest with the most internet users, and phishing specialist countries, also with high internet connectivity but average wealth.

Soomro and Hussain [90] discussed recommendations and techniques for preventing cyber crime. They discussed different types of cyber crime such as Burglary via Social Networking, Social Engineering and Phishing, Identity Theft, Cyber-Stalking etc. and their prevention techniques correspondingly.

7.3 Materials and methods

To analyze the cyber crime pattern, crime records are collected from the National Crime Record Bureau (NCRB) [175] and Open Government Data (OGD) [176] of India recorded during the year 2011-2018 for 28 States and 8 Union Territories of India (Table 7.2). Data has been collected under the IT act 2000 and some Indian Penal Code (IPC) such as –

1. Offences by/ against Public Servant
2. False electronic evidence
3. Destruction of electronic evidence
4. Criminal Breach of Trust/Fraud
5. Counterfeiting
    (a) Property/mark
    (b) Tampering
    (c) Currency/Stamps

To analyze the crime pattern during these mentioned years, Linear Regression [114] and Polynomial Regression [114] models are used. The linear regression method is useful for finding the relationship between two continuous variables. One is the predictor or independent variable and the other is the response or dependent variable. Since real-life data is not generally linear, we also use Polynomial Regression to establish a non-linear relationship between the independent and dependent variables.

Linear regression is a used to model the relationship between two variables by a straight line of the form $Y = a + bX$. Y is called the dependent variable, X is the independent variable, b is the slope of the line and a is the y-intercept. a and b are measured using Eq. 7.1 and Eq. 7.2

Table 7.2 Number of cyber crimes recorded during the year 2011-2018 in India.

State and Union Territories	2011	2012	2013	2014	2015	2016	2017	2018
Andhra Pradesh	372	454	651	282	536	616	931	1207
Arunachal Pradesh	14	12	10	18	6	4	1	7
Assam	31	28	154	379	483	696	1120	2022
Bihar	38	30	139	114	242	309	433	374
Chhattisgarh	78	59	101	123	103	90	171	139
Goa	18	32	58	62	17	31	13	29
Gujarat	67	78	77	227	242	362	458	702
Hariyana	45	182	323	151	224	401	504	418
Himachal Pradesh	12	20	28	38	50	31	56	69
Jammu and Kashmir	14	35	46	37	34	28	63	73
Jharkhand	33	35	26	93	180	259	720	930
Karnataka	160	437	533	1020	1447	1101	3174	5839
Kerala	245	312	383	450	290	283	320	340
Madhya Pradesh	103	197	342	289	231	258	490	740
Maharashtra	393	561	907	1879	2195	2380	3604	3511
Manipur	0	0	1	13	6	11	74	29
Meghalaya	6	6	17	60	56	39	39	74
Mizoram	3	0	0	22	8	1	10	6
Nagaland	0	0	0	0	0	2	0	2
Odisha	12	27	104	124	386	317	824	843
Punjab	79	78	156	226	149	102	176	239
Rajasthan	146	154	297	697	949	941	1304	1104
Sikim	4	0	0	4	1	1	1	1
Tamil Nadu	45	41	90	172	142	144	228	295
Telangana	0	0	0	703	687	593	1209	1205
Tripura	0	14	14	5	13	8	7	20
Uttar Pradesh	114	249	682	1737	2208	2639	4971	6280
Uttarakhand	6	4	27	42	48	62	124	171
West Bengal	57	309	342	355	398	478	568	335
A and N Island	0	2	18	13	6	3	3	7
Chadigarh	10	33	11	55	77	26	32	30
D and N Haveli	6	0	0	3	0	1	1	0
Daman and Diu	1	0	1	1	1	0	0	0
Delhi UT	99	84	150	226	177	98	162	189
Lakshadweep	0	0	0	1	0	0	0	4
Puducherry	2	4	5	1	0	2	5	14

$$a = \frac{(\sum y)(\sum x^2) - (\sum x)(\sum xy)}{n(\sum x^2) - (\sum x)^2} \tag{7.1}$$

$$b = \frac{n(\sum xy) - (\sum x)(\sum y)}{n(\sum x^2) - (\sum x)^2} \tag{7.2}$$

Polynomial regression is a special case of linear regression where we fit a polynomial equation on the data with a curvilinear relationship between the target variable and the independent variables. The value of the target variable changes in a non-uniform manner with respect to the predictor(s) following Eq. 7.3.

$$Y = a + b_0 X + b_1 X^2 + b_2 X^3 + ... + b_n X^n \tag{7.3}$$

In this equation a is the y-intercept, and b_o, b_1, b_2, ..., b_n are the weights in the equation of the polynomial regression and n is the degree of the polynomial.

7.4 Results and Discussions

7.4.1 National Level Cyber crime prediction under IT act 2000

To predict the number of cyber crime events at a national level, year wise total number of cyber crime occurrences in the nation is calculated by adding the number of cyber crime events in different states and territories. The total number of recorded cyber crimes in India from 2011 to 2018 is listed in Table 7.3. The plot shown in Figure 7.1 shows the total number of recorded crimes in India from 2011 to 2018 assuming the x-axis represents the year and the y-axis represents the number of crimes. It can be observed that the highest number of crimes occurred in 2018. But to measure the growth of cyber crime events per year, Eq. 7.4 is used, where GP_Y indicates the Growth Percentage of Year Y. The growth rate is shown in Table 7.3. From this information, it can be concluded that the growth rate of cyber crime events is increasing every year and the highest growth has been observed in the year 2017 through the highest number of crimes occurring in 2018.

$$GP_Y = \frac{(Numberofcyber\ crimesinYearY - Numberofcyber\ crimesinYear(Y-1))}{(Numberofcyber\ crimeeventsinYear(Y-1))} X100 \tag{7.4}$$

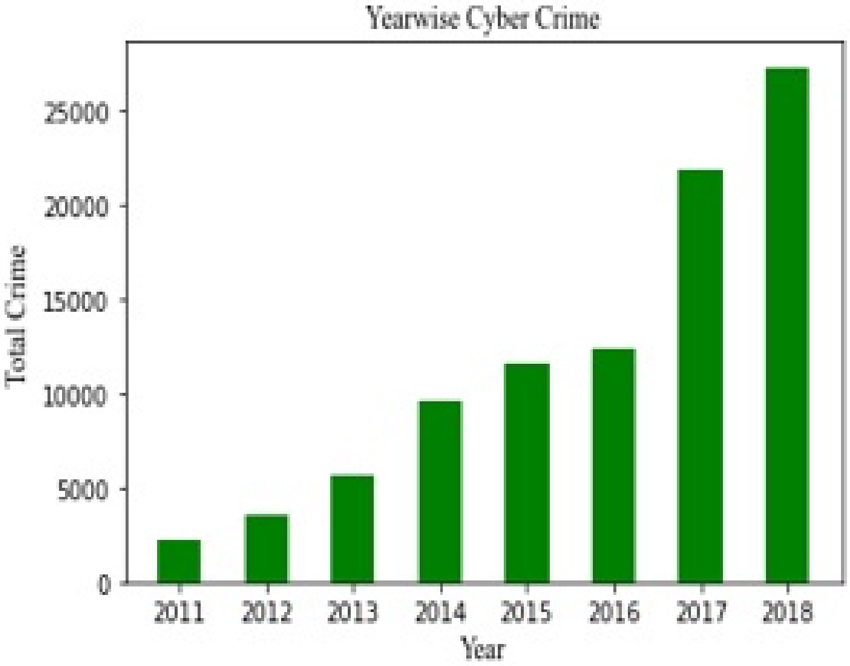


Fig. 7.1 Year-wise total cyber crime.

Table 7.3 Growth percentage of cyber crimes.

Year	2011	2012	2013	2014	2015	2016	2017	2018
Total Crime	2213	3477	5693	9622	11592	12317	21796	27248
Growth Rate	-	57.11%	63.73%	69.01%	20.47%	6.25%	76.96%	25.01%

The relationship between total cyber crime events (shown in Table 7.3) and year is established using linear regression assuming year as an independent variable (X) and Total cyber crime as a dependent variable (Y). Figure 7.2 shows the associations between the total number of cyber crime events in India with year. The relationship achieves a R^2 value of approximately 0.915. The relationship is also established using polynomial regression of degrees 2, 3, and 4 using Eq. 7.3. The associations using polynomial regression of degrees 2, 3, and 4 are shown in Figure 7.3 (a-c) respectively. It is observed that the polynomial regression of degree 2 achieves a R^2 value of approximately 0.971, whereas both polynomial regressions of degree 3 and 4 achieve a R^2 value of approximately 0.978.

The number of cyber crime in India is predicted for the year 2019 to 2021 using polynomial regression of degree 4 since it gives the best R^2 value. The predicted numbers of cyber crimes are listed in Table 7.4. The predicted value is compared with the actual cyber crime events that occured

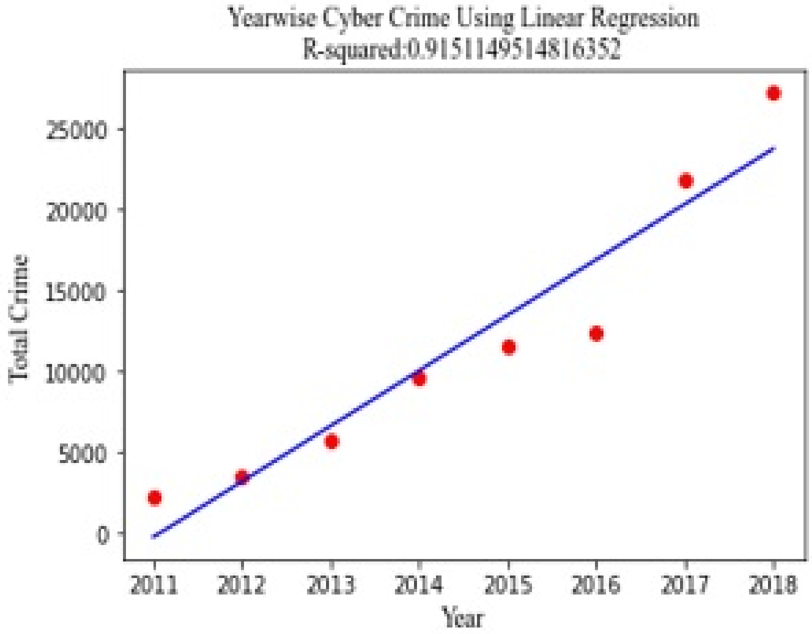


Fig. 7.2 The associations between total numbers of cyber crime with year using Linear Regression.

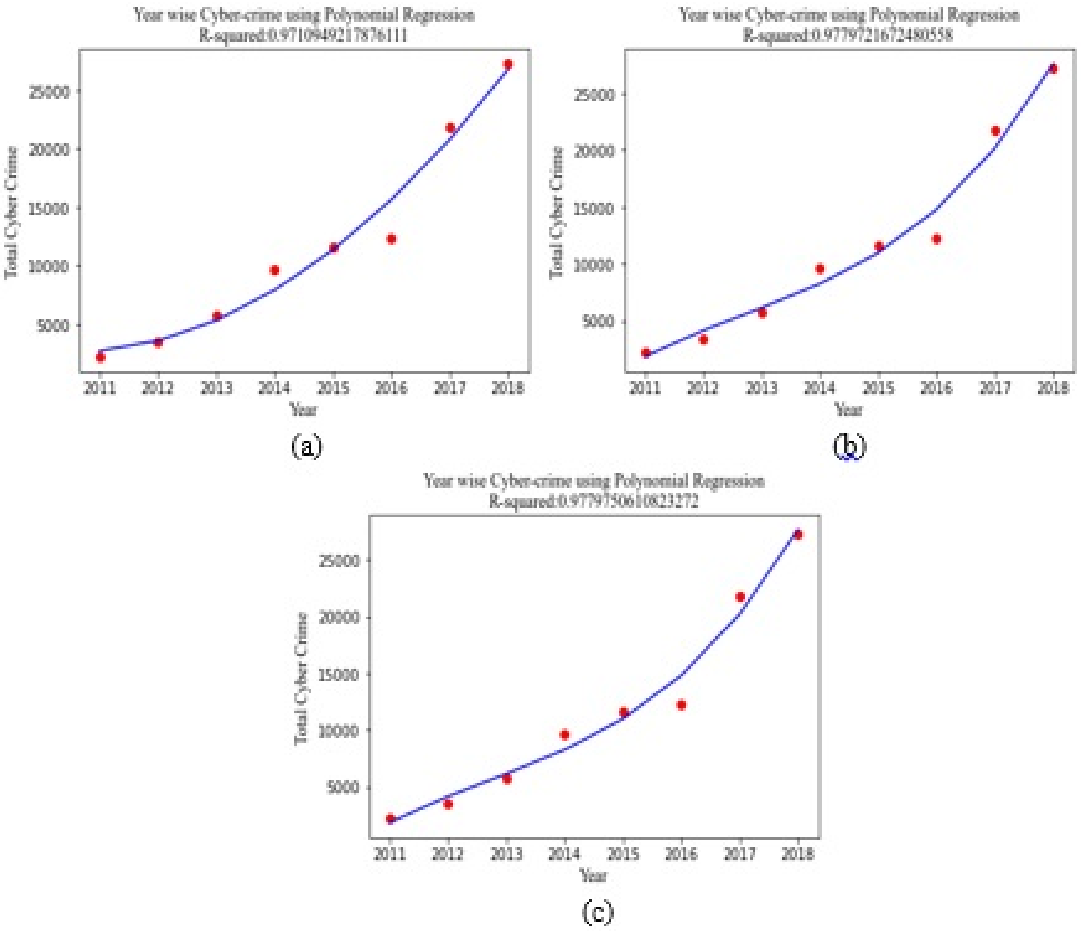
in 2019 for measuring the accuracy of the prediction, since the data are not currently available for 2020 and 2021 in the database. It is observed that the actual cyber crime event in 2019 is 44546 and the predicted value for the year is 37512. The predicted value has a relative error rate nearly of 15%.

Table 7.4 Predicted numbers of cyber crime events in India.

Year	2019	2020	2021
Predicted Crime	37512	50440	66841

7.4.2 State Level Cyber crime prediction under IT act 2000

In this section, we have concentrated on analyzing the cyber crime events in each of the 28 States and 8 Union Territories of India during the same period of time, i.e. from the year 2011 to 2018. For each state and union territory, the relationship between the year and number of cyber crime events is established using polynomial regression of degree 4 using Equation 7.3. In this model year is considered as an independent variable (X) and year-wise total crime is considered as the dependent variable (Y). The associations for each state and union territory are shown in Fig. 7.9 (i - xxxvi).



(a)

(b)

(c)

Fig. 7.3 The associations between total numbers of cybercrime with year using Polynomial Regression model with (a) Degree 2 (b) Degree 3 (c) Degree 4.

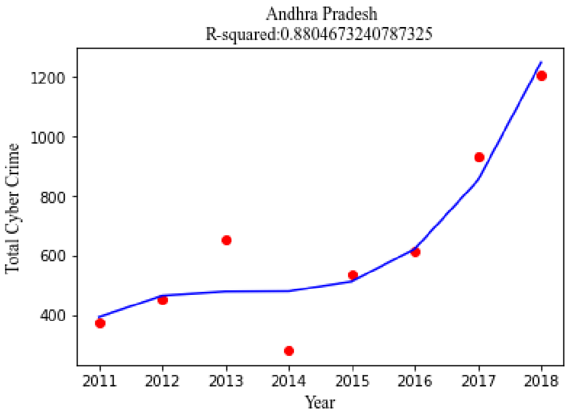
Among these 28 States and 8 Union Territories, high correlations are observed for 17 (with R^2 value > 0.8), a moderate correlation is observed for 10 (with R^2 value between .6 and .8) and very low correlations are noted for the rest 9 (R^2 value < .6). The numbers of cyber crime events for the year 2019, 2020 and 2021 are predicted for those 17 states that pose high correlations. The predictions are shown in Table 7.5. For some states, negative values are obtained as the number of cyber crime events that are ignored and represented as a blank entry ('-') in Table 7.5. For validation of the claimed prediction, the actual cyber crimes events for the year 2019 are compared with the predicted value and the relative error rate of the prediction is calculated (Table 7.6).

Table 7.5 Predicted value of cyber crime for 17 states during years 2019 to 2021 that poses high correlation (with R^2 value > 0.8).

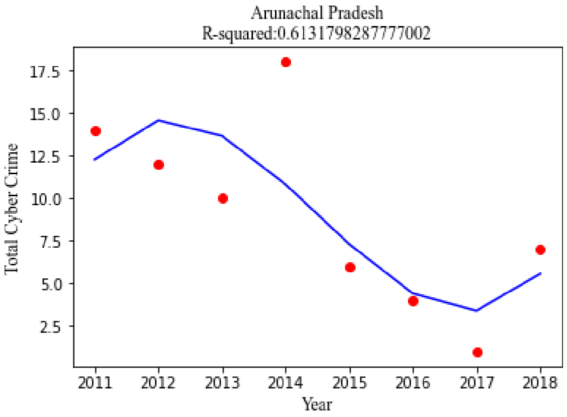
State	2019	2020	2021
Andhra Pradesh	1855	2716	3878
Assam	3066	4564	6530
Bihar	349	209	-
Gujarat	928	1233	1606
Himachal Pradesh	93	129	178
Jammu and Kashmir	130	212	330
Jharkhand	1438	2037	2780
Karnataka	9829	15656	23561
Madhya Pradesh	1297	2122	3286
Maharashtra	3697	3493	2916
Odisha	1117	1350	1581
Rajasthan	789	140	-
Tamil Nadu	395	539	953
Telangana	1287	1249	1078
Uttar Pradesh	8644	11464	14897
Uttarakhand	255	365	507
Puducherry	29	53	87

7.4.3 National Level Section Wise Cyber crime prediction under IT act 2000

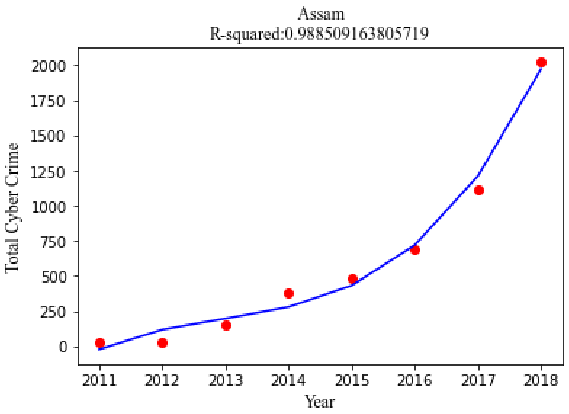
We have also studied the section-wise national-level crime pattern in India from 2011 to 2018. But in this study, only sections 65, 66, and 67 are considered since other sections have negligible occurrence (number of crimes less than 10). The number of cyber crimes under these sections is listed in Table 7.7. The correlations of the number of cyber crimes in each of these sections with year



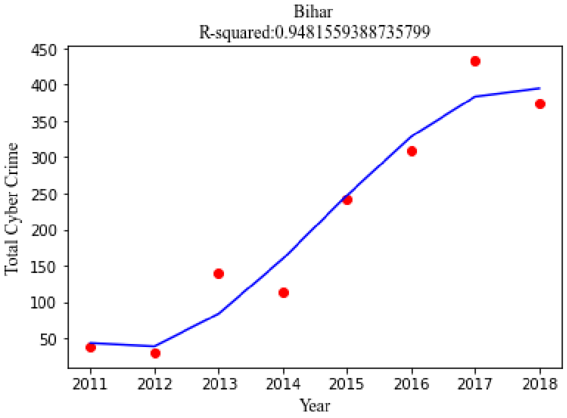
(i) Andhra Pradesh



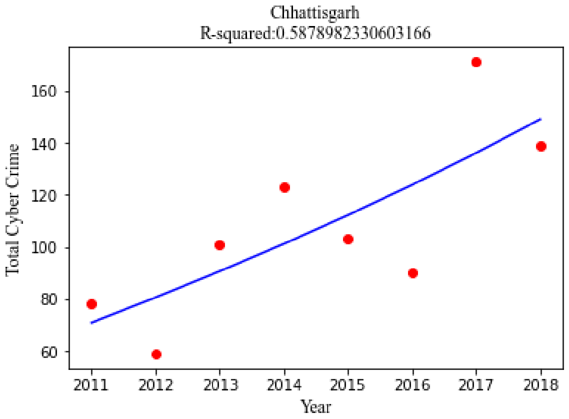
(ii)Arunachal Pradesh



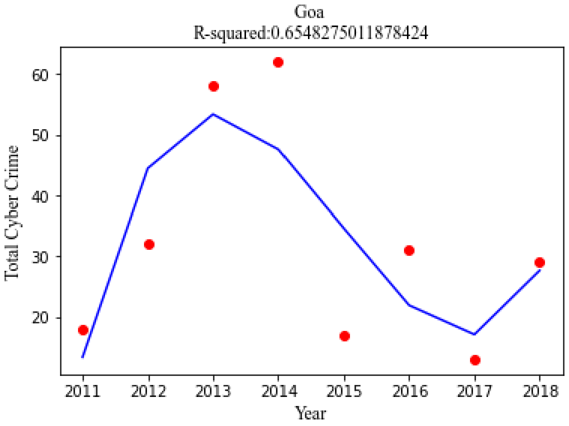
(iii) Assam



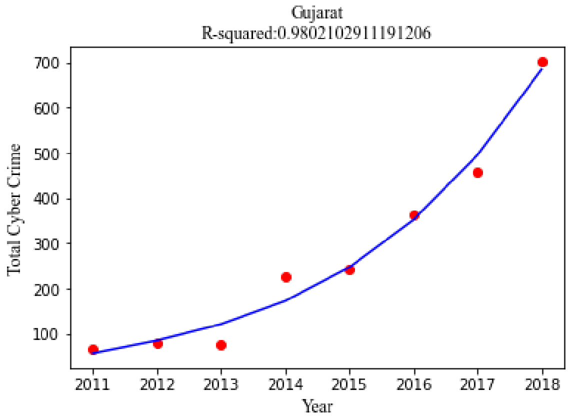
(iv) Bihar



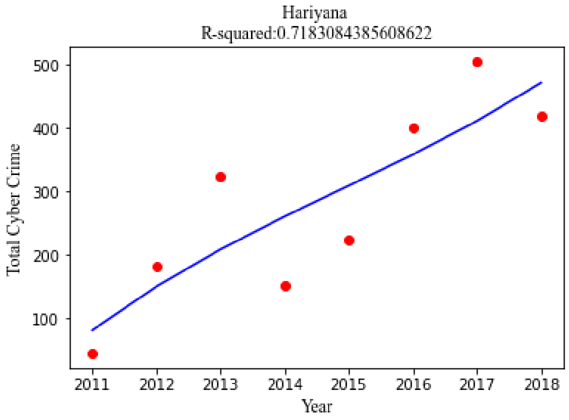
(v) Chhattisgarh



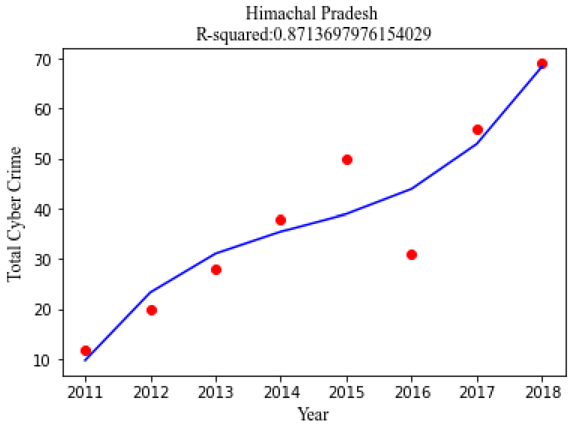
(vi) Goa



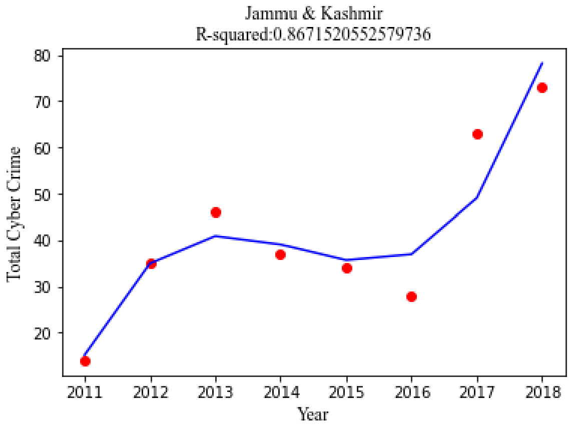
(vii) Gujarat



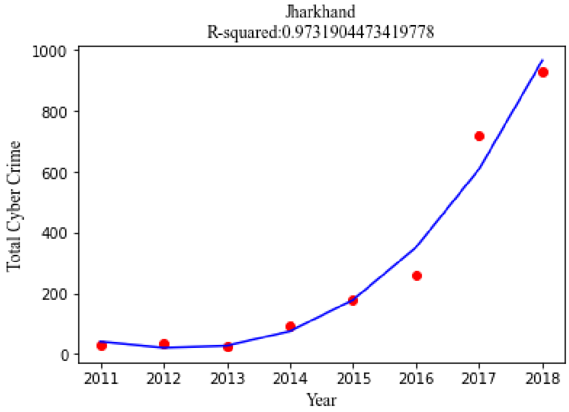
(viii) Hariyana



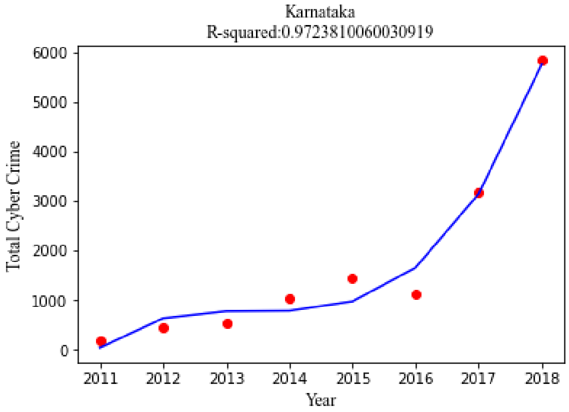
(ix) Himachal Pradesh



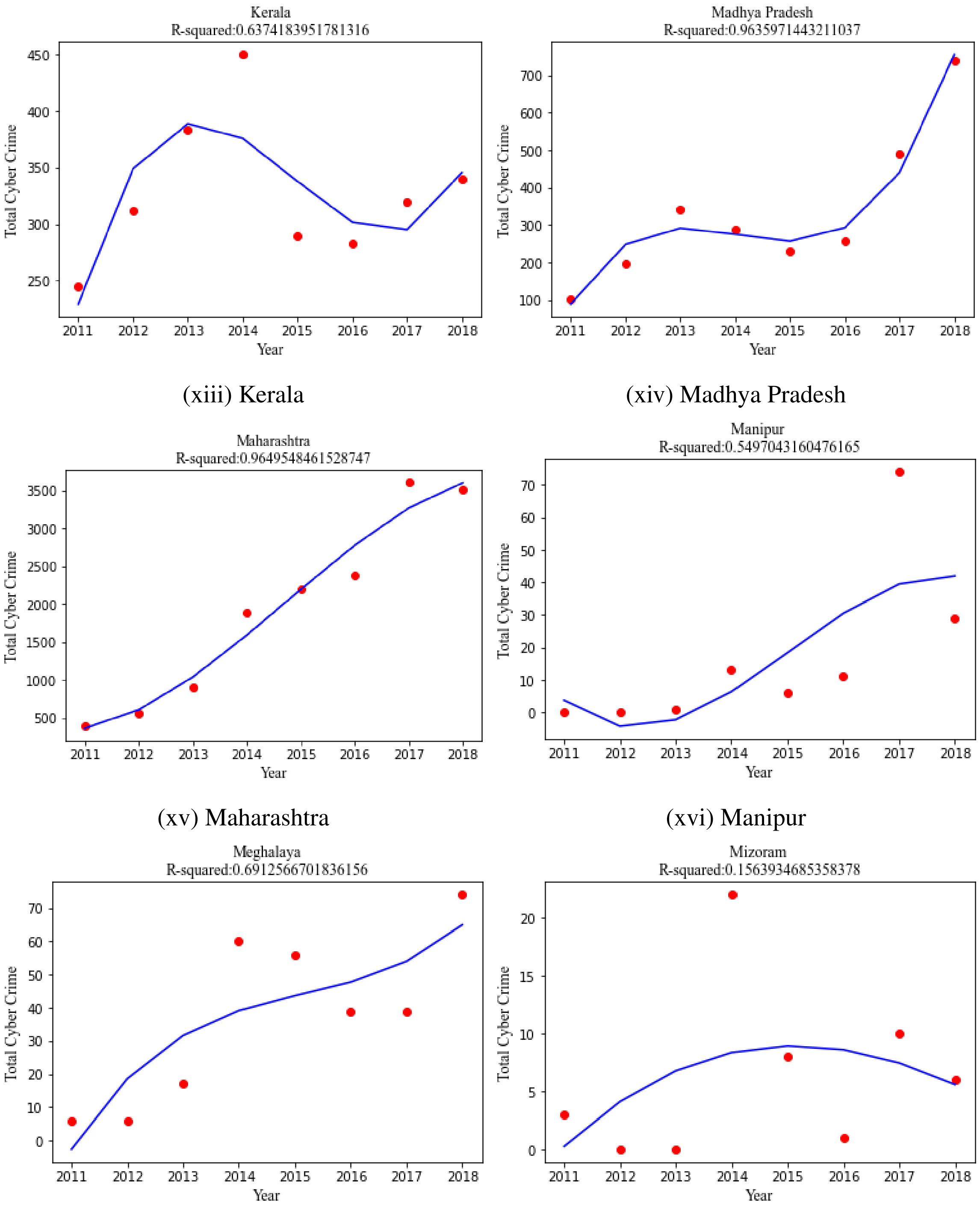
(x) Jammu and Kashmir



(xi) Jharkhand



(xii) Karnataka



(xiii) Kerala

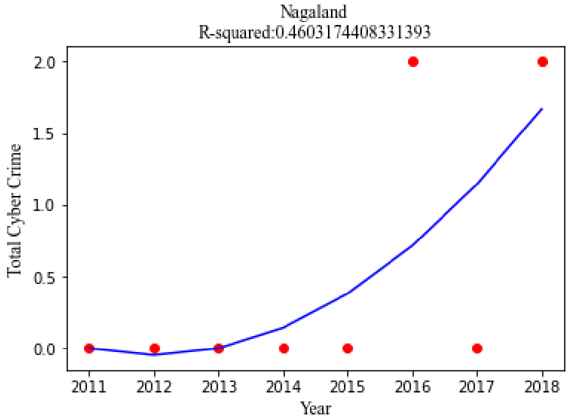
(xiv) Madhya Pradesh

(xv) Maharashtra

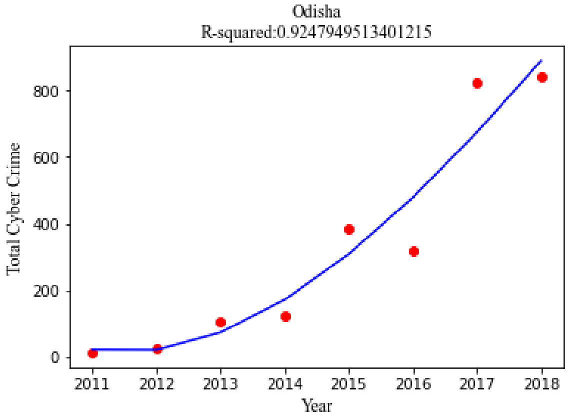
(xvi) Manipur

(xvii) Meghalaya

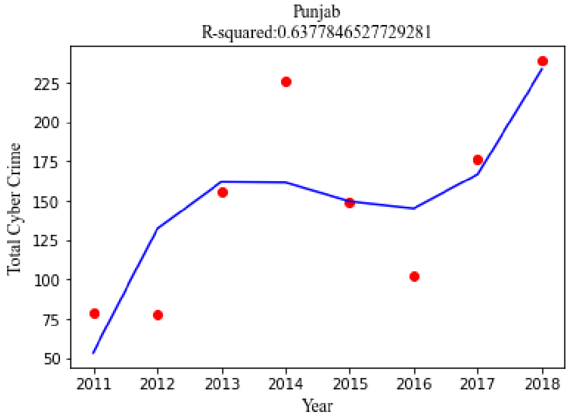
(xviii) Mizoram



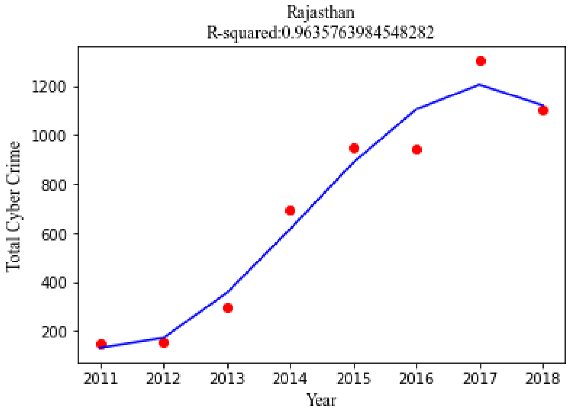
(xix) Nagaland



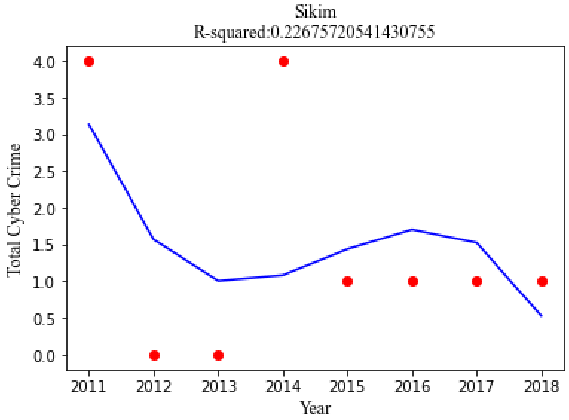
(xx) Odisha



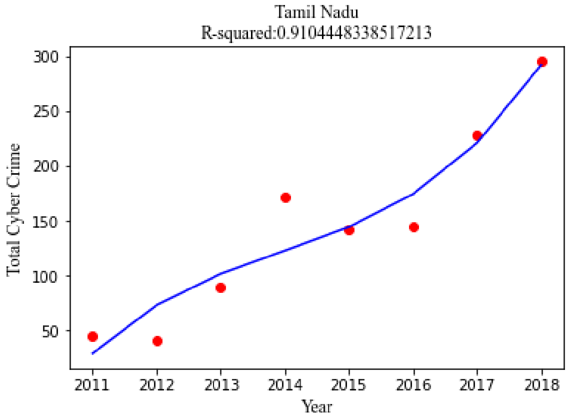
(xxi) Punjab



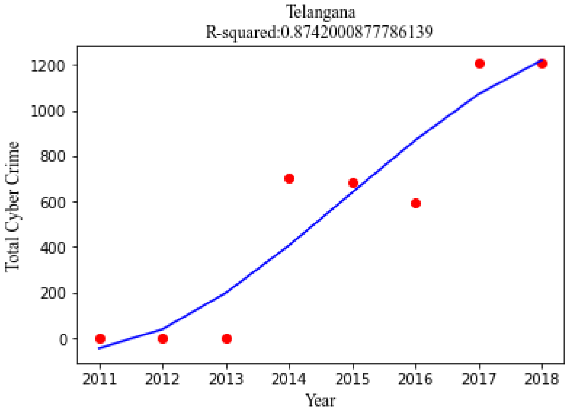
(xxii) Rajasthan



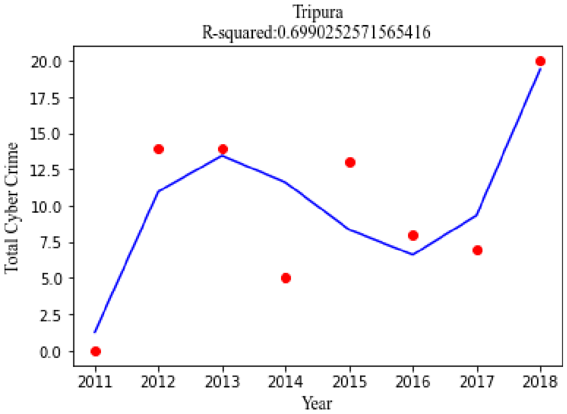
(xxiii) Sikim



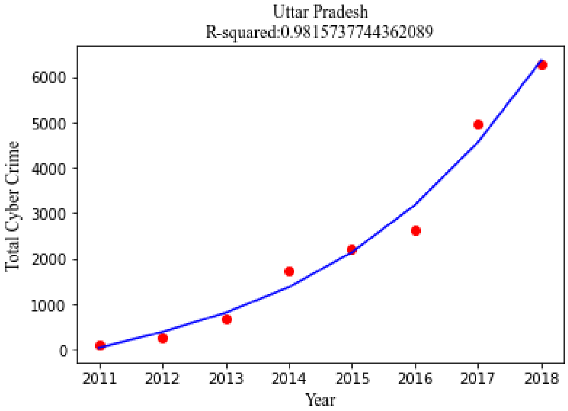
(xxiv) Tamil Nadu



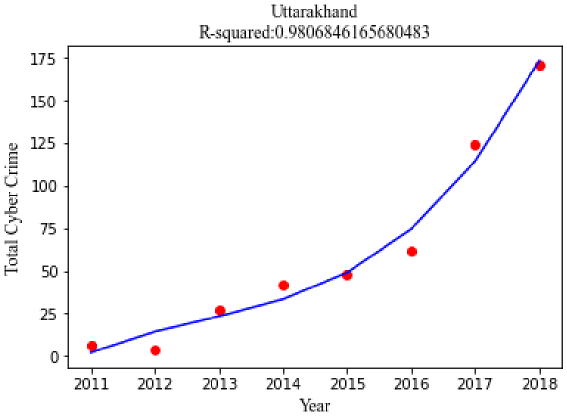
(xxv) Telengana



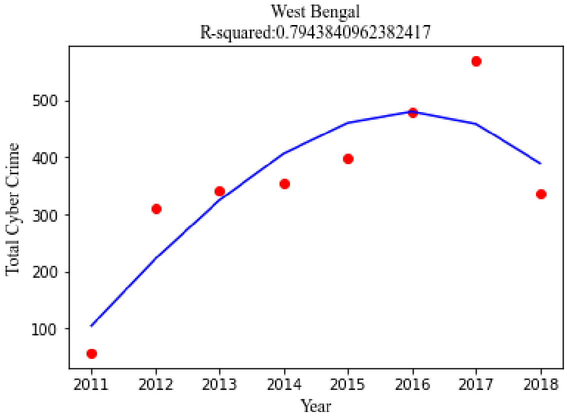
(xxvi) Tripura



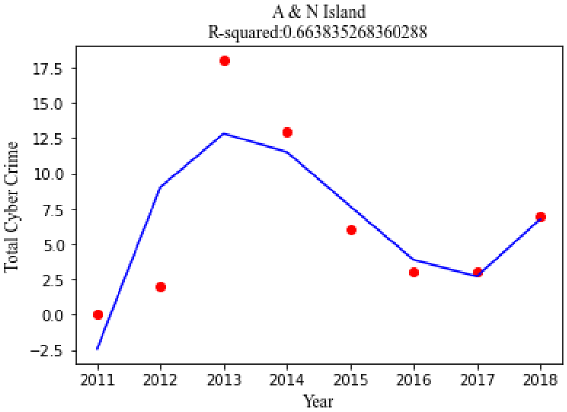
(xxvii) Uttar Pradesh



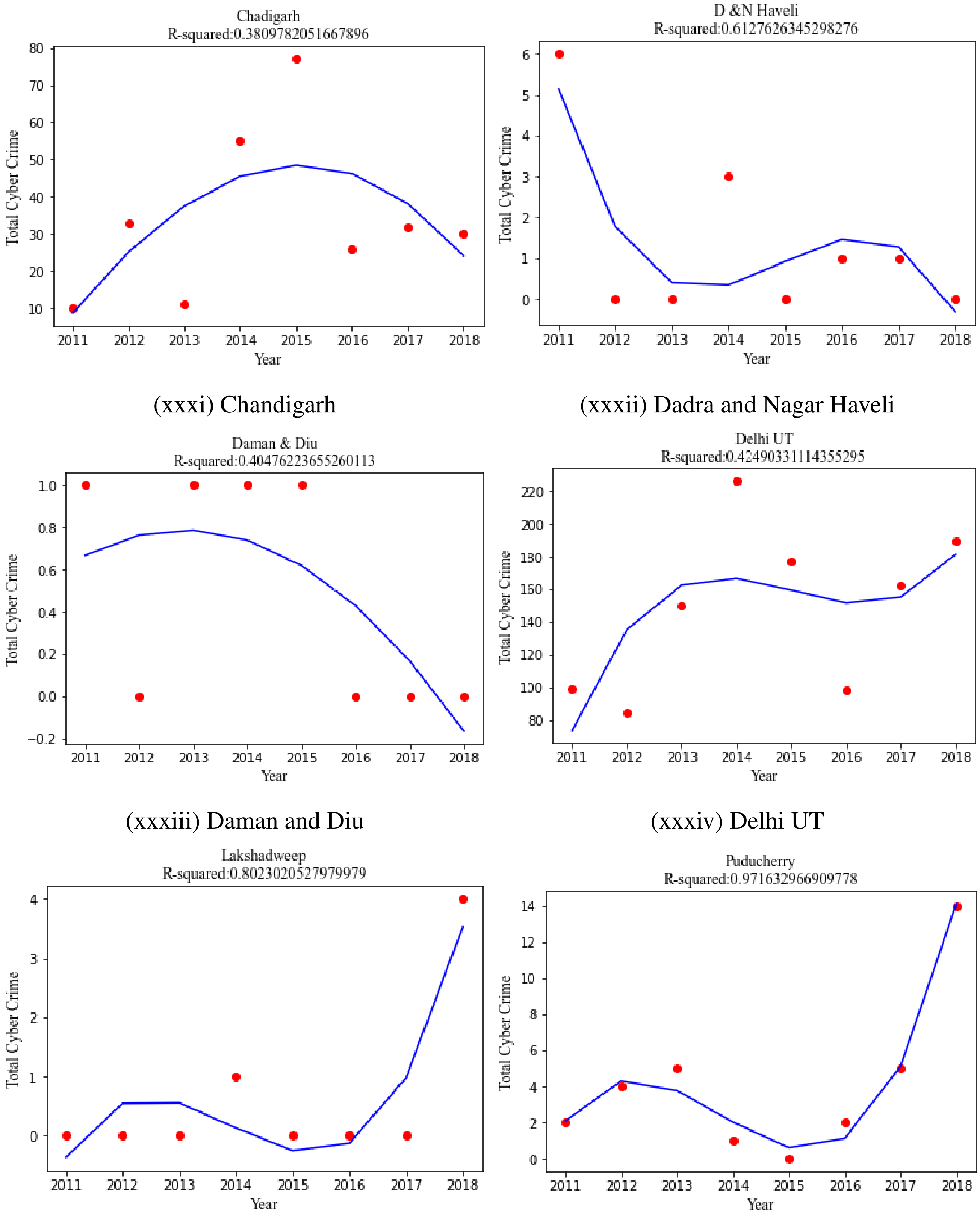
(xxviii) Uttarakhand



(xxix) West Bengal



(xxx) Andaman and Nikobar Island



(xxxi) Chandigarh

(xxxii) Dadra and Nagar Haveli

(xxxiii) Daman and Diu

(xxxiv) Delhi UT

(xxxv) Lakshadweep

(xxxvi) Puducherry

Fig. 7.9 The associations for each state and union territory in India using Polynomial

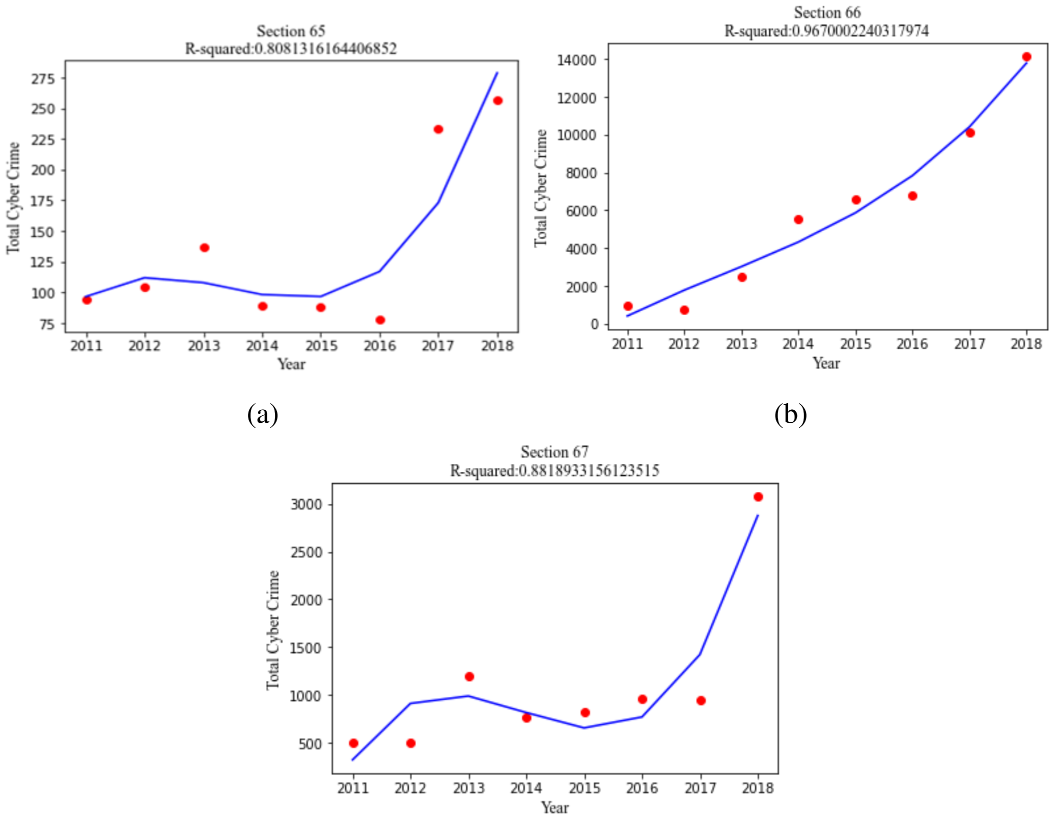
Table 7.6 Relative error rate in the prediction of the state-wise number of cyber crimes in 2019.

State	Predicted Crime (2019)	Actual Crime (2019)	Relative Error
Andhra Pradesh	1855	1886	1.64%
Assam	3066	2231	37.43%
Bihar	349	1050	66.76%
Gujarat	928	784	18.37%
Himachal Pradesh	93	76	22.37%
Jammu and Kashmir	130	73	78.08%
Jharkhand	1438	1095	31.32%
Karnataka	9829	12020	18.23%
Madhya Pradesh	1297	602	115.45%
Maharashtra	3697	4967	25.57%
Odisha	1117	1485	24.78%
Rajasthan	789	1762	55.22%
Tamil Nadu	395	385	2.60%
Telangana	1287	2691	52.17%
Uttar Pradesh	8644	11416	24.28%
Uttarakhand	255	100	155.00%
Puducherry	29	4	625.00%

are measured using polynomial regression of degree 4 using Eq. 7.3. These correlations achieve R^2 values of approximately 0.80, 0.97, and .88 for sections 65, 66, and 67 respectively. The associations are shown in Figure 7.10. Since high correlations (R^2>0.8) are observed in section 66 and section 67 of crimes, the associations are used to predict the section-wise number of cyber crimes (for sections 66 and 67) in India for years 2019, 2020, and 2021. The predicted values are shown in Table 7.8. The results of the prediction are compared with actual data for 2019 and the relative error rate is calculated (Table 7.9).

7.5 Summary

In this chapter, the growth of cyber crime events in India from 2011 to 2018 is analyzed for 28 States and 8 Union Territories. The data are analyzed at the national level, and state level as well as in different sections of crimes. It is observed that the number of occurrence of cyber crime events in India maintain a high correlation ($R^2 > 0.8$) with the year. The same investigation is also



(a) (b)

(c)

Fig. 7.10 The associations between total numbers of cyber crime with year using Polynomial Regression model with degree 4 for (a) Section 65 (b) Section 66 (c) Section 67.

Table 7.7 The number of cyber crimes in India under sections 65, 66, and 67 in the year 2011 to 2018.

Year	Section 65	Section 66	Section 67
2011	94	983	496
2012	104	749	497
2013	137	2516	1203
2014	89	5553	758
2015	88	6580	816
2016	78	6816	957
2015	233	10108	948
2018	257	14141	3076

Table 7.8 Predicted value of cyber crime during the year 2019 to 2021 under Section 66 and Section 67 that poses high correlation (with R^2 value > 0.8).

Section	2019	2020	2021
Section 66	18144	23663	30532
Section 67	5399	9252	14701

Table 7.9 Relative error rate in the prediction for Section 66 and Section 67 in 2019.

Section	Predicted Data in 2019	Actual Data in 2019	Relative Error rate
Section 66	18144	23612	23%
Section 67	5399	4332	25%

performed state-wise. But at the state level among 28 States and 8 Union Territories, only 17 show high correlations. To analyze section-wise cyber crime events it is observed that only three sections (sections 65, 66, and 67) have a dominant influence, since the number of occurrences of cyber crimes under other sections is less than 10. Among these three sections, only two sections (sections 66 and 67) hold a high correlation. This information is used to predict the number of cyber crime events for the year 2019 to 2021 at the national level as well as state and section levels for those which show high correlation. The results of predictions are compared with the actual occurrence of cyber crimes for 2019. The predicted values have a relative error rate of less than 25% in most of the cases.

Chapter 8

Conclusion and Future Scope

Crime analysis is a systematic approach for reviewing the collected data in order to identify the pattern, nature, behaviors, and trends of criminal activities. Machine learning has immense applications in the field of crime analysis. The results of systematic crime analysis can help police or different agencies to reduce the rate of crime.

Crime is a major problem in day to day life. Though no one has any right to do the crime, but crimes occur in society. It has a direct impact on the economic prosperity of a nation, privacy and security of the compatriots. To reduce crime rates, novel techniques should be developed to analyze patterns and trends of crimes. There are various forms of crimes present in our society. Our research focuses on both traditional crimes and cyber crimes. In the case of traditional crime analysis, crime against children in India is investigated and the different factors that influence the crime rate are measured. An automated approach is proposed to predict the geological location of a crime. Within cyber crimes, (i) spam SMS classification, (ii) annotation detection of cyber-bullying messages, (iii) phishing email classification, and (iv) phishing website classification are performed. Finally, an automated technique is proposed to predict the pattern and trends of cyber crimes in India. To achieve the said objectives, different statistical, machine learning, and deep learning models are proposed.

In chapter 2, a statistical approach has been applied to find the pattern of crime against children in India. Different factors which influence the rate of crime against children, are also determined. Finally, an automated methodology is proposed to predict the location of the crime using a machine learning algorithm by analyzing past crime records. To analyze crime against children in India, crime records have been collected from NCRB over the period 2001 to 2020. These records have been analyzed to find the crime rate of different states and union territories in India. The experimental results show that the union territory Delhi has the highest crime rate, followed by the A&N Islands, Chandigarh, Sikkim, and Madhya Pradesh. ADF stationary test has been applied and test results establish that first-order differences support the permission level of stationary. Using these crime records, growth rate of crimes against children in each state has been calculated. It is found that the state of Delhi achieved the highest growth rate of 4, followed by the A&N Islands, Chandigarh, Sikkim, and Madhya Pradesh. The overall growth rate of crime against children in India is 0.6622. We use the Cuddy –Della Valle Index to analyze the stability of growth rate. The Ordinary Least Square (OLS) technique is used for the multi co-linearity test between the factors unemployment rate, literacy rate, digitization rate, and urbanization rate. The test results indicate that the unemployment rate and literacy rate have a negative impact, whereas the digitization rate and urbanization rate directly influence the rate of crime against children. For predicting the location of crimes, linear regression and support vector machine models are used. The models are tested using publicly available crime records of Indore city. It is observed from the experimental results that, the Support vector regression works well and has more accuracy if the data points lie within the clustering region since it has a low RMS error value. But, the Linear Regression works well if the points are located near the boundaries.

Chapter 3, proposed two different deep neural networks, namely, CNN and a hybrid CNN-LSTM models along with two different word embedding techniques – BUNOW and GloVe for detecting spam SMS. The proposed models were applied over widely used Tiago's data set. The data set consists of text messages. Lower casing of the text, tokenization, lemmatization, and removal of stop words, symbols, numbers, and words with lengths less than 2 were applied in Text pre-processing steps. The experiments have been carried out for different train-test splits. Experimental results show that the CNN models give best accuracy with BUNOW and GloVe word embedding on 90% -10% train-test split. But CNN-LSTM BUNOW performed best among the four models with an accuracy of 99.04%, 99.01%, 98.92% and 98.44% for 85% – 15%, 80% – 20%, 75% – 25% and 70% – 30% train-test splits respectively. The performances of the proposed models are also compared with state of the art methods. It is observed that the proposed simple models succeed in performing better than the existing complex models. To increase the performance of the proposed model further, N-grams, spelling correction, replacing abbreviated words may be applied in the pre-processing steps in the future. The proposed model can also be extendable by using more generalized deep neural structures.

Chapter 4, proposed an automated technique for detecting the annotation of cyber bullying messages using CNN, LSTM, BLSTM deep neural networks with BUNOW and GloVe word embedding models. Two different text pre-processing techniques are applied to cyber bullying text messages. Traditional pre-processing includes lower casing the text, tokenization, lemmatization, removal of non-alphabetic tokens, and stop word. The proposed advanced text pre-processing includes all the steps present in traditional pre-processing as well as a few new steps, like replacing URLs, numbers and emojis; removal of HTML tags; replacing abbreviated words with appropriate strings; and considering the occurrence of an URL by storing it in separate database. The Twitter data set has been used to evaluate the performance of the proposed method. The proposed models with advanced pre-processing give better accuracy in all cases. Thus, it can be concluded that if the feature vectors are increased appropriately, the accuracy of the model may also increase. The proposed technique considers only three types of annotations - racism, sexism, and normal. It can further be extended to include other annotation types. In proposed works, the annotation detection is performed on text data. The proposed method can also be extended to predict the annotation by considering audio-video messages also.

A comparative study of content based phishing email detection using two different models is discussed in chapter 5. The first model consists of a convolution neural network (CNN) with GloVe word embedding. The second model uses the Bidirectional Encoder Representation from Transformer (BERT) model and fine-tuned it by using fully connected neural networks. Due to the lack of a proper data set, five publicly available data sets have been merged together to evaluate the performance of proposed models. The first model achieves 98% accuracy whereas the second model achieves 96% accuracy. Since the BERT model is predefined, it classifies an email as phishing or legitimate, depending on the first 512 words of the email.But if an email consists of more than 512 words, the proposed BERT model may fail to properly classify the email. Since our data set consists of many emails with more than 512 words, the second model achieves less accuracy than the first one. This limitation may be removed by extending the current study in the future. An email with more than 512 words may be divided into k parts, not more than 512 words in each. After that, classify each subtext to identify whether it is phishing or legitimate. If any one of the subtext is phishing, then the overall emails is considered as a phishing email, otherwise it is a legitimate email.

Chapter 6, proposed an ensemble stacking model for detecting phishing websites. Ensemble models are a combination of multiple base models. Our proposed ensemble model consists of seven base models, namely, Naïve Bayes (NB), K-Nearest Neighbour (K-NN), Support Vector Machine (SVM), Random forest (RF), Bagging, Logistic Regression (LR) and Neural Network (NN). The study was performed over the publicly available data set - UCI Machine Learning Repository of phishing websites. The data sets consists of 30 features. The optimal features selection is an important part of detecting phishing websites. For this context, Random Forest Regressor model has been used. Out of 30 features, 16 features are selected as important features because their score is more than 1% of the highest score. The proposed method achieved 96.73% accuracy with an f1 score of 97.07%. The main contribution of this paper is the application of a greedy approach to improve the accuracy of the ensemble stacking model. The accuracy of the proposed method is compared with existing ensemble learning models for phishing website detection. Nowadays, the types of phishing attacks are continuously changing. Therefore, the study can be easily enhanced by considering more sensitive features.

Chapter 7, analyzes the growth of cyber crimes in India and proposed a model for predicting cyber crimes in upcoming years. The study is performed by collecting the cyber crime data from

NCRB from the year 2011 to 2018 over 28 states and 8 union territories in India. The data is analyzed at national level, state level as well as different sections of cyber crimes. It is found that, the incident of cyber crimes data is highly co-related to the R^2 value with a correlation value of 0.8. In India, 17 states are found to be highly co-related to incidents of cyber crime.To study section-wise cyber crime events in India, it is observed that only three sections (sections 65, 66, and 67) have a dominant influence, since the number of occurrences of cyber crimes under other sections is less than 10. Among these three sections, only two sections (sections 66 and 67) hold a high correlation. The study is also performed for prediction of cyber crime events for the year 2019 to 2021 at the national level as well as state and section level for those that show high correlation. The results of prediction is also compared with the actual incidence of cyber crime for the year 2019. The predicted values have a relative error rate of less than 25% in most of the cases.

Analysis of crime in India is performed by analyzing crime data, but the major issue is the availability of the crime data set. Adequate data is not available for performing machine learning or deep learning models. Nowadays, National Crime Records Bureau (NCRB) provides recent crime data in the public domain. Due to the lack of sufficient data, the analysis of crime against children and different cyber crimes have been analyzed using different statistical tools. The work can be extended to analyze and predict the crimes using different machine learning or deep learning models more accurately, when sufficient data sets become available.

In this research, Machine learning and deep learning models are employed, which are powerful capabilities but they also come with several limitations. Implementing Machine learning and mainly deep learning can be complex and increases dataset can increased complexity. To train deep learning models powerful hardware is required such as GPUs or TPUs, which are huge cost. Overfitting is one of the major problem in deep learning and resolve it in our research by using different technique.

Finally, the analysis of crimes in India is a huge task. In this study, some specific domains of crimes are analyzed. The study can be extended further by considering other crime domains like crime against women, white collar crime, etc.

References

[1] C. Du Plessis. Ensemble based systems in decision making. *The Links between Crime Prevention and Sustainable Development*, 24:33–40, 1999.

[2] S Gottlieb, S Arenberg, and R Singh. *Crime Analysis: From First Report to Final Arrest.* OJP Publication, 1994.

[3] Sandipa Bhattacharjee, Ankit Chopra, and Ankit Mohanty. CRIME AGAINST CHILDREN IN INDIA: PREVENTIVE AND PROTECTIVE LAWS. *INTERNATIONAL JOURNAL OF LEGAL DEVELOPMENTS AND ALLIED ISSUES*, 6(5), 2020.

[4] V. Shaharban. CRIMES AGAINST CHILDREN-PREVALENCE AND PREVENTION STRATEGIES. In *Two Day International Conference on CHILD RIGHTS: EMERGING ISSUES AND CONCERNS*, 2019.

[5] A. Bogomolov, B. Lepri, J. Staiano, N. Oliver, F. Pianesi, and A. Pentland. Once upon a crime: towards crime prediction from demographics and mobile data. In *Proc. of the 16th Intl. Conf. on Multimodal Interaction*, pages 427–434, 2014.

[6] I. Rizwan, M. Masrah anf A. Mustapha, P. Panahy, and K. Nasim. An Experimental Study of Classification Algorithms for Crime Prediction. *Indian Journal of Science and Technology*, 6:4219–4225, 2013.

[7] V. Dhar. Data Science and Prediction. *Communications of the ACM*, 56(12):64–73, 2013.

[8] C. Cappa and I. Jijon. COVID-19 and violence against children: A review of early studies. *Elsevier Ltd*, 116(2), 2021.

[9] S. Ramaswamy and S. Seshadri. Children on the brink: Risks for child protection, sexual abuse, and related mental health problems in the COVID-19 pandemic. *Indian Journal of Psychiatry*, 62(Suppl 3), 2020.

[10] C. H. Zeanah and K. L. Humphreys. Child abuse and neglect. *Journal of the American Academy of Child and Adolescent Psychiatry*, 57(9):637–644, 2018.

[11] S. D. Hillis, J. A. Mercy, and J.R. Saul. The enduring impact of violence against children. Psychology. *Health and Medicine*, 22(4):393–405, 2017.

[12] S. Maity and S. Roy. Analysis of Growth and Identifications of the Determinants of Crime against Women: Insight from India. *Journal of International Women's Studies* , 22(1), 2021.

[13] A. Mavi. A study of the impact of macroeconomic factors on crime against children in India. *International research journal* , 1(3):332, 2021.

[14] S. Shaik and R.P. Rajkumar. Internet access and sexual offenses against children: an analysis of crime bureau statistics from India. *Open J Psychiatry Allied Sci*, 6:112–6, 2015.

[15] M. Gupta and P. Sachdeva. Economic, Demographic, Deterrent Variables And Crime Rate In India. *Munich Personal RePEc Archive(UTC)*, 80181, 2017.

[16] N. Bodhgire and A. Muley. Identification of Regression Model for Crime Rate and Literacy Rate in India. *Journal of the University of Shanghai for Science and Technology*, 23 (4):245, 2021.

[17] M. Dutta and Z. Husain. Determinants of crime rates: Crime Deterrence and Growth in post-liberalized India. *Munich Personal RePEc Archive(UTC)*, 14478, 2009.

[18] S. Shojaee, A. Mustapha, F. Sidi, and M.A. Jabar. A Study on Classification Learning Algorithms to Predict Crime Status. *Int. J. Digit. Content Technol. its Appl.*, 7(9):361–369, 2013.

[19] T. Wang, C. Rudin, D. Wagner, and R. Sevieri. Learning to Detect Patterns of Crime: Machine Learning and Knowledge Discovery in Databases. In *European Conference, ECML PKDD, Prague, Czech Republic*, 2013.

[20] R.Yadav and S.K. Savita. Analysis of Criminal Behavior through Clustering Approach. *International Journal of Computer Sciences and Engineering*, 6, 2018.

[21] X. Chen, Y. Cho, and S. Y. Jang. Crime prediction using Twitter sentiment and weather. In *Systems and Information Engineering Design Symposium (2015)*, 2015.

[22] T. A. Almeida, J. María, G.Hidalgo, and A. Yamakami. Contributions to the study of SMS spam filtering: new collection and results. In *DocEng '11: Proceedings of the 11th ACM symposium on Document engineering*, page 259–262, 2011.

[23] P. Sethi, V. Bhandari, and B. Kohli. SMS spam detection and comparison of various machine learning algorithms. In *2017 International Conference on Computing and Communication Technologies for Smart Nation (IC3TSN)*, 2017.

[24] P. Navaney, G. Dubey, and A. Rana. SMS Spam Filtering Using Supervised Machine Learning Algorithms. In *2018 8th International Conference on Cloud Computing, Data Science and Engineering (Confluence)*, 2018.

[25] A. Alzahrani and D.B.Rawat. Comparative Study of Machine Learning Algorithms for SMS Spam Detection. In *Proceeding of the IEEE South East Conference, Huntsville*, 2019.

[26] T.Xia and X. Chen. A Discrete Hidden Markov Model for SMS Spam Detection. *Applied Sciences*, 10(14), 2020.

[27] T.Xia and X. Chen. A weighted feature enhanced Hidden Markov Model for spam SMS filtering. *Neurocomputing*, 444:48–58, 2021.

[28] B. Diallo, J. Hu, T. Li, G. Khan, and A. S. Hussein. Concept-Enhanced Multi-view Clustering of Document Data. In *IEEE 14th International Conference on Intelligent Systems and Knowledge Engineering (ISKE) - Dalian*, page 1258–1264, 2019.

[29] B. Diallo, J. Hu, T. Li, G. Khan, and A. S. Hussein. Multi-view document clustering based on geometrical similarity measurement. *International Journal of Machine Learning and Cybernetics*, 13:663–675, 2021.

[30] R. Taheri and R. Javidan. Spam filtering in SMS using recurrent neural networks. In *Artificial Intelligence and Signal Processing Conference (AISP)*, 2017.

[31] G. Jain, M. Sharma, and B. Agarwal. Optimizing semantic LSTM for spam detection. *International Journal of Information Technology*, 11 (2):239–250, 2019.

[32] M. Popovac, M. Karanovic, S. Sladojevic, M. Arsenovic, and A. Anderla. Convolutional Neural Network based SMS Spam Detection. In *26th Telecommunication Forum (TELFOR)*, pages 1–4, 2018.

[33] S. Annareddy and S. Tammina. A Comparative Study of Deep Learning Methods for Spam Detection. In *Third International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)*, 2019.

[34] P. K. Roy, J. P. Singh, and S. Banerjee. Deep Learning to Filter SMS Spam. *Future generation computer systems*, 102:524–533, 2019.

[35] A. Chandra and S. K. Khatri. Spam SMS Filtering using Recurrent Neural Network and Long Short-Term Memory. In *4th International Conference on Information Systems and Computer Networks (ISCON)*, 2019.

[36] S. Kotni, C. Potala, and L. Sahoo. Spam Detection Using Deep Learning Models. *International Journal of Advanced Research in Engineering and Technology*, 13 (5):55–64, 2022.

[37] O. Abayomi-Alli, S. Misra, and A. Abayomi-Alli. A deep learning method for automatic SMS spam classification: Performance of learning al-gorithms on an indigenous dataset. *Concurrency and Computation: Practice and Experience*, 34(17), 2022.

[38] B. Diallo, J. Hu, A. G. Khan T. Li, X. Liang, and Y. Zhao. Deep embedding clustering based on contractive autoencoder. *Neurocomputing*, 143:96–107, 2021.

[39] A. Ghourabi, A. M. Mahmood, and M. A. Qusay. A Hybrid CNN-LSTM Model for SMS Spam Detection in Arabic and English Messages. *Future Internet*, 12(9):156, 2020.

[40] M. A. Shaaban, F. H. Yasser, and K. G. Shawkat. Deep convolutional forest: a dynamic deep ensemble approach for spam detection in text. *Complex and Intelligent Systems*, 8(6):4897–4909., 2022.

[41] L. Cheng, J. Li, Y.N. Silva, D. L. Hall, and H. Liu. PIBully: Personalized Cyberbullying Detection with Peer Influence. In *Proceeding of Twenty-Eighth International Jt. Conference Artificial Intelligence*, pages 5829–5835, 2019.

[42] A. Muneer and S. M. Fati. A Comparative Analysis of Machine Learning Techniques for Cyberbullying Detection on Twitter. *Future Internet*, 12 (11):187, 2020.

[43] D. Chatzakou, I. Leontiadis, J. Blackburn, E.D. Cristofaro, G. Stringhini, and A. Vakali nd N. Kourtellis. Detecting cyberbullying and cyberaggression in social media. *ACM Trans. Web (TWEB)*, 13(3):1–51, 2019.

[44] J. Zhang, T. Otomo, L. Li, and S. Nakajima. Cyberbullying Detection on Twitter using Multiple Textual Features. In *IEEE 10th Int. Conf. Aware. Sci. Technol. (iCAST), Japan*, pages 1–6, 2007.

[45] X. Zhang, J. Tong, N. Vishwamitra, E. Whittaker, J. Mazer, R. Kowalski, H. Hu, F. Luo, J. Macbeth, and E. Dillon. Cyberbullying Detection with a Pronunciation Based Convolutional Neural Network. In *IEEE International Conference on Machine Learning and Applications*, pages 740–745, 2016.

[46] M. A. Al-Ajlan and M. Ykhlef. Optimized Twitter Cyberbullying Detection based on Deep Learning. In *21st Saudi Computer Society National Computer Conference (NCC).*, 2018.

[47] V. Banerjee, J. Telavane, P. Gaikwad, and P. Vartak. Detection of Cyberbullying Using Deep Neural Network. In *5th International Conference on Advanced Computing Communication Systems (ICACCS)*, 2019.

[48] S. Agrawal and A. Awekar. Deep learning for detecting cyberbullying across multiple social media platforms. In *European Conference on Information Retrieval, Springer*, page 141–153, 2018.

[49] N. Lu, G. Wu, Z. Zhang, Y. Zheng, Y. Ren, and R. ChooK. Cyberbullying detection in social media text based on character-level convolutional neural network with shortcuts. *Concurrency and Computation: Practice and Experience*, 2020.

[50] M. Dadvar and K. Eckert. Cyberbullying Detection in Social Networks Using Deep Learning Based Models. In *Springer Nature Switzerland AG 2020*, page 245–255, 2020.

[51] D. V. Bruwaene, Q. Huanget, and D. Inkpen. A multi-platform dataset for detecting cyberbullying in social media. *Language Resources and Evaluation*, 54 (4):851–874, 2020.

[52] C. Iwendi, G. Srivastava, S. Khan, and P. K. Maddikunta. Cyberbullying detection solutions based on deep learning architectures. *Springer-Verlag GmbH Germany, part of Springer Nature*, 2020.

[53] S. Ghosh, A. Chaki, and A. Kudeshia. Cyberbully Detection Using 1D-CNN and LSTM. In *Proceedings of International Conference on Communication, Circuits, and Systems, Springer Singapore*, pages 295–301, 2021.

[54] Z. Zhao, M. Gao, F. Luo, Y. Zhang, and Q. Xiong. LSHWE: Improving Similarity-Based Word Embedding with Locality Sensitive Hashing for Cyberbullying Detection. In *International Joint Conference on Neural Networks (IJCNN).*, 2020.

[55] S. O. Olatunji. Extreme Learning machines and Support Vector Machines models for email spam detection. In *2017 IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE)*, 2017.

[56] M. Singh, R. Pamula, , and S. k. shekhar. Email Spam Classification by Support Vector Machine. In *2018 International Conference on Computing, Power and Communication Technologies (GUCON)*, 2019.

[57] V. D. Sharma, S. Yadav, S. Yadav, K. Singh, and Suraj P. Sharma. An effective approach to protect social media account from spam mail – A machine learning approach. In *Materials Today: Proceedings.doi:10.1016/j.matpr.2020.12.377*, 2021.

[58] R. Nayak, S. AmiraliJiwani, and B. Rajitha. Spam email detection using machine learning algorithm. In *Materials Today: Proceedings.doi:10.1016/j.matpr.2020.12.377*, 2021.

[59] U. Bhardwaj and P. Sharma. Email Spam Detection using Ensemble Methods. *Int. J. Recent Technol. Eng. (IJRTE)*, 8 (3):4148–4153, 2019.

[60] Panagiotis Bountakas and Christos Xenakis. HELPHED: Hybrid Ensemble Learning PHishing Email Detection. *Journal of Network and Computer Applications*, 210, 2023.

[61] Jemal Abawajy and Andrei Kelarev. A multi-tier ensemble construction of classifiers for phishing email detection and filtering. In *nternational Symposium on Cyberspace Safety and Security*, pages 48–56, 2012.

[62] S. Srinivasan, V. Ravi, M. Alazab, S. Ketha, A.Z. Ala'M, and S.K. Padannayil. Spam emails detection based on distributed word embedding with deep learning. *Machine Intelligence and Big Data Analytics for Cybersecurity Applications, Springer*, page 161– 189, 2021.

[63] S. Sumathi and G. K. Pugalendhi. Cognition based spam mail text analysis using combined approach of deep neural network classifier and random forest. *Journal of Ambient Intelligence and Humanized Computing*, 12:5721–5731, 2021.

[64] A.N.Soni. Spam e-mail detection using advanced deep convolution neuralnetwork algorithms. *JOURNAL FOR INNOVATIVE DEVELOPMENT IN PHARMACEUTICAL AND TECHNICAL SCIENCE*, 2(5):74–80, 2019.

[65] E. Castillo, S. Dhaduvai, Peng Liu, K. S. Thakur, A. Dalton, and T. Strzalkowski. Email Threat Detection Using Distinct Neural Network Approaches. In *Proceedings of the Workshop on Social Threats in Online Conversations: Understanding and Management (STOC-2020)*, pages 48–55, 2020.

[66] Y. Fang, C. Zhang, C. Huang, L. Liu, and Y. Yang. Phishing Email Detection Using Improved RCNN Model With Multilevel Vectors and Attention Mechanism. *IEEE Access*, 7:56329–56340, 2019.

[67] M. Manaswini and DR. N. Srinivasu. Phishing Email Detection Model using Improved Recurrent Convolutional Neural Networks and Multilevel Vectors. *Annals of R.S.C.B.*, 25(6):16674–16681, 2021.

[68] R. Vinayakumar, K. P. Soman, P. Poornachandran, S. Akarsh, and M. Elhoseny. *Deep Learning Framework for Cyber Threat Situational Awareness Based on Email and URL Data Analysis*. Advanced Sciences and Technologies for Security Applications (Springer), 2019.

[69] M. Hiransha, A. U. Nidhin, R. Vinayakumar R, and K. P. Soman. Deep Learning Based Phishing E-mail Detection CEN-Deepspam. In *Proceedings of the 1st AntiPhish- ing Shared Pilot at 4th ACM International Workshop on Se- curity and Privacy Analytics (IWSPA 2018)*, 2018.

[70] G. Chetty, H. Bui, and M. White. Deep Learning Based Spam Detection System. In *2019 International Conference on Machine Learning and Data Engineering (iCMLDE)*, 2019.

[71] L. Nguyen, B. To, H. Nguyen, and M. Nguyen. Detecting phishing websites: A heuristic URL-based approach. In *2013 International Conference on Advanced Technologies for Communications (ATC 2013)*, pages 597–602, 2013.

[72] Y. Zhang, J. I. Hong, and L. F. Cranor. Cantina: a content-based approach to detecting phishing websites. In *Proceedings of the 16th International Conference on World Wide Web*, page 639–648, 2007.

[73] A. K. Jain and B. B. Gupta. Phishing Detection: Analysis of Visual Similarity Based Approaches. *Security and Communication Networks*, 2017:1–20, 2017.

[74] A. Subasi, E. Molah, F. Almkallawi, and T. J. Chaudhery. Intelligent phishing website detection using random forest classifier. In *2017 International Conference on Electrical and Computing Technologies and Applications (ICECTA), Ras Al Khaimah, United Arab Emirates*, pages 1–5, 2017.

[75] S. Hutchinson, Z. Zhang, and Q. Liu. Detecting Phishing Websites with Random Forest. In *Third International Conference, MLICOM*, pages 470–479, 2018.

[76] R. Anagora, Rudini, R. Rohmat Taufiq, A. Jubaedi, R. Wirawan, and A. Putra. The Classification of Phishing Websites using Naive Bayes Classifier Algorithm. *International Journal of Science, Technology Management,*, 3(2):553–562, 2022.

[77] Waleed Ali. Phishing Website Detection based on Supervised Machine Learning with Wrapper Features Selection. *(IJACSA) International Journal of Advanced Computer Science and Applications*, 8(9), 2017.

[78] Altyeb Altaher. Phishing Websites Classification using Hybrid SVM and KNN Approach. *(IJACSA) International Journal of Advanced Computer Science and Applications*, 8(6), 2017.

[79] D. Opitz and R. Maclin. Popular ensemble methods: An empirical study. *Journal of Artificial Intelligence Research*, 11:169–198, 1999.

[80] R. Polikar. Ensemble based systems in decision making. *IEEE Circuits and Systems Magazine*, 6(3):21–45, 2006.

[81] L. Rokach. Ensemble-based classifiers. *Artificial Intelligence Review*, 33(1–2):1–39, 2010.

[82] K. Nagaraj, B. Bhattacharjee, A. Sridhar, and G. Sharvani. Detection of phishing websites using a novel twofold ensemble model. *Journal of Systems and Information Technology*, 20(3):321–357, 2018.

[83] A. A. Ubing, S. K. B. Jasmi, A. Abdullah, N. Jhanjhi, and M. Supramaniam. Phishing website detection: An improved accuracy through feature selection and ensemble learning . *International Journal of Advanced Computer Science and Applications* , 10(1):252–257, 2019.

[84] Y. Chandra and A. Jana. 6th International Conference on Computing for Sustainable Global Development (INDIACom) IEEE. In *Improvement in phishing websites detection using meta classifiers*, page 637–641, 2019.

[85] S.O. Folorunso andF.E. yo and K. A. Abdullahand P.I. Ogunyinka. Hybrid vs ensemble classification models for phishing websites. *Iraqi J. Sci*, 61:3387–3396, 2020.

[86] G. Tsakalidis and K. A. Vergidis. Systematic Approach Toward Description and Classification of Cybercrime Incidents. . *IEEE Trans. Syst. Man Cybern. Syst.*, 49:710–729, 2019.

[87] M. Ganesan and P. Mayilvahanan. Cyber Crime Analysis in Social Media Using Data Mining Technique. *Int. J. Pure Appl. Math.*, 116:413–424, 2017.

[88] S. Prabakaran and S. Mitra. Survey of Analysis of Crime Detection Techniques Using Data Mining and Machine Learning. *J. Phys. Conf. Ser.*, 1000, 2018.

[89] A. Kigerl. Cyber Crime Nation Typologies: K-Means Clustering of Countries Based on Cyber Crime Rates. *Int. J. Cyber Criminol.*, 10:147–169., 2016.

[90] T. R. Soomro and H. Mumtaz. Social Media-Related Cybercrimes and Techniques for Their Prevention. *Appl. Comput. Syst.*, 24:9–17, 2019.

[91] S. Banerjee, S. Giri, D. Das, and P. K. Mondal. An Approach to Predict the Location of Crime Using Machine Learning. In *Advances in Intelligent Systems and Computing book series (AISC,volume 1397)*, page 943–951, 2021.

[92] S. Giri, S. Das, S.B. Das, and S. Banerjee. SMS Spam Classification–Simple Deep Learning Models With Higher Accuracy Us- ing BUNOW And GloVe Word Embedding. *Journal of Applied Science and Engineering*, 26(10):1501–1511, 2022.

[93] S. Giri and S. Banerjee. Performance analysis of annotation detection techniques for cyberbullying messages using word-embedded deep neural networks. *Social Network Analysis and Mining*, 13(1):1–12, 2023.

[94] S. Giri, S. Banerjee, K. Bag, and D. Maiti. Comparative Study of Content-Based Phishing Email Detection Using Global Vector (GloVe) and Bidirectional Encoder Representation from Transformer (BERT) Word Embedding Models. In *2022 First International Conference on Electrical, Electronics, Information and Communication Technologies (ICEEICT)*, pages 1–6, 2022.

[95] Govt. Of India. The child labour (prohibition and regulation) act,1986@ONLINE. https://labour.gov.in/sites/default/files/act_3.pdf, March 2022.

[96] MINISTRY OF LAW and JUSTICE. The juvenile justice (care and protection of children)act, 2015 @ONLINE. https://cara.nic.in/PDF/JJ%20act%202015.pdf, March 2022.

[97] Govt. Of India. prohibition-child-marriage-act-2006 @ONLINE. https://www.india.gov.in/prohibition-child-marriage-act-2006, March 2022.

[98] MINISTRY OF LAW and JUSTICE. The protection of children from sexual offences@ONLINE. https://wcd.nic.in/sites/default/files/POCSO%20Act%2C%202012.pdf, March 2022.

[99] MINISTRY OF LAW and JUSTICE. The juvenile justice (care and protection of children)amendment act, 2021@ONLINE. https://egazette.nic.in/WriteReadData/2021/228833.pdf, March 2022.

[100] National Crime Records Bureau. @ONLINE. https://ncrb.gov.in/, March 2022.

[101] H. Hassani, X. Huang, E.S. Silva, and M. Ghodsi. A review of data mining applications in crime. *Stat. Anal. Data Min. ASA Data Sci. J.*, 9(3):139–154, 2016.

[102] E. Ahishakiye, I. Niyonzima, and R. Wario. A Performance Analysis of Business Intelligence Techniques on Crime Prediction. *Int. J. Comput. Inf. Technol.*, 6(2):84–90, 2017.

[103] A. G. Ferguson. Big data and predictive reasonable suspicion. *Univ. PA. Law Rev.*, 163(2):329–346, 2012.

[104] U. Saeed, M. Sarim, A. Usmani, A. Mukhtar, A.B. Shaikh, and S.K. Raffat. Application of Machine learning Algorithms in Crime Classification and Classification Rule Mining. *Res. J. Recent Sci. Res.J.Recent Sci*, 4(3):106–114, 2015.

[105] D. Usha and K. Rameshkumar. A Complete Survey on application of Frequent Pattern Mining and Association Rule Mining on Crime Pattern Mining . *Int. J. Adv. Comput. Sci. Technol.* , 3(4):264–275, 2014.

[106] S. Schneider. *Predicting Crime: A Review of the Research*. 2002.

[107] W. L. Perry, B. McInnes, C. C. Price, S. C. Smith, and J. S. Hollywood. *Predictive Policing: The Role of Crime Forecasting in Law Enforcement Operations*. 2013.

[108] O. Jain, M. Gupta, S. Satam, and S Panda. Has the COVID-19 pandemic affected the susceptibility to cyberbullying in India? *Computers in Human Behavior Reports*, 2, 2020.

[109] National Crime Record Bureau (NCRB). @ONLINE. https://ncrb.gov.in/, May 2021.

[110] Open Government Data (OGD). @ONLINE. https://data.gov.in/;https://www.digitalindia.gov.in/, May 2021.

[111] L. P. Beland, A. Brodeur, J. Haddad, and D. Mikola. COVID-19, Family Stress and Domestic Violence: Remote Work, Isolation and Bargaining Power. *Global Labor Organization*, 47 (3):439–459, 2021.

[112] R. Amin. Mathematical Model of Crime and Literacy Rates. *International Journal of Mathematics Trends and Technology (IJMTT)*, 65(9):2231–5373, 2019.

[113] A. A. Malik. Urbanization and Crime: A Relational Analysis. *IOSR Journal Of Humanities And Social Science (IOSR-JHSS)* , 21(1):68–74, 2016.

[114] D. C. Montgomery, E. A. Peck, and G. G Vining. *Introduction to Linear Regression Analysis, 5th Edition*. Wily, 2012.

[115] M.Awad and R. Khanna. Support vector regression. In *Efficient Learning Machines, Springer*, page 67–80, 2015.

[116] What is text message marketing@ONLINE. https://www.tatango.com/, March 2022.

[117] A. A. Helmy, Y. M.K. Omar, and R. Hodhod. An Innovative Word Encoding Method For Text Classification Using Convolutional Neural Network. In *14th International Computer Engineering Conference (ICENCO2018), Faculty of Engineering, Cairo University, Egypt*, 2018.

[118] J. Pennington, R. Socher, and C. Manning. GloVe: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, page 1532–1543, 2014.

[119] T.A. Almeida and J.M.G Hidalgo. Sms spam collection v.1.@ONLINE. http://www.dt.fee.unicamp.br/~tiago/smsspamcollection/, March 2022.

[120] Dataset@ONLINE. https://archive.ics.uci.edu/ml/datasets/YouTube+Spam+Collection., March 2022.

[121] Z. Jianqiang, G. Xiaolin, and Z. Xuejun. Deep Convolution Neural Networks for Twitter Sentiment Analysis. *IEEE Access*, 6:23253–23260, 2018.

[122] S. M. Rezaeinia, R. Rahmani, A. Ghodsi, and H. Veisi. Sentiment Analysis based on improved Pre-trained Word Embeddings. *Expert Systems with Application*, 117:139–147, 2019.

[123] A. Sharma, P. Malacaria, and M. Khouzani. Malware Detection Using 1-Dimensional Convolutional Neural Networks. In *2019 IEEE European Symposium on Security and Privacy Workshops (EuroS and PW)*, 2019.

[124] H. Raj, Y. Weihong, S. Kumar, B., and P. D. Soomro. LSTM Based Short Message Service (SMS) Modeling for Spam Classification. In *ICMLT '18: Proceedings of the 2018 International Conference on Machine Learning Technologies*, pages 76–80, 2018.

[125] D. Scherer, A. Muller, and Sven Behnke. Evaluation of Pooling Operations in Convolutional Architecture for Object Recognition. In *20th International Conference on Artificial Neural Networks (ICANN),Thessaloniki, Greece*, 2010.

[126] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakutdinov. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research* , 15, 2014.

[127] A. K. Uysa and S. Gunal. The Impact of Preprocessing on Text Classification. . *Information Processing and Management* , 50(1):104 – 112, 2014.

[128] J. Camacho-Collados and M. T. Pileavar. On the Role of Text Preprocessing in Neural Network Architectures: An Evaluation Study on Text Categorization and Sentiment Analysis. In *Proceedings of the 2018 EMNLP Workshop Blackbox NLP: Analyzing and Interpreting Neural Networks for NLP*, pages 40–46, 2018.

[129] S. Weidman. *Deep Learning from Scratch: Building with Python from First Principles*. O'Reilly, 1st edition, 2019.

[130] D. Nordstokke and A. M. Stelnicki. *Encyclopedia of Quality of Life and Well-Being Research*. Springer Nature, 2014.

[131] M. Campbell and S. Bauman. Cyberbullying: definition, consequences, prevalence.Reducing Cyberbullying in Schools: International Evidence-Based Best Practices. *Elsevier, London, UK*, page 3–16, 2018.

[132] L Arseneault and S Shakoor. Bullying victimization in youths and mental health problems: 'much ado about nothing. *Psychol.Med.*, 40(5):717, 2009.

[133] D. Sharma, J Kishore, N Sharma, and Duggal. Aggression in schools: cyberbullying and gender issues. *Asian J Psychiatr* , 29:142–145, 2017.

[134] Pew Research Center. A majority of teens have experienced some form of cyberbullying.@ONLINE. https://www.pewresearch.org/internet/2018/09/27/a-majority-of-teens-have-experienced-some-form-of-cyberbullying/, 2018.

[135] Pew Research Center. The state of online harassment.@ONLINE. https://www.pewresearch.org/internet/2021/01/13/the-state-of-online-harassment/, 2021.

[136] A. E. Bradley. The Use Of The Area Under The Roc Curve In The Evaluation Of Machine Learning Algorithms. *Pattern Recognition*, 30(7):1145–1159, 1997.

[137] Cyberbullying data set (2020) @ONLINE. https://data.mendeley.com/datasets/jf4pzyvnpj/1#__sid=js0, March 2022.

[138] Economic impact of cybercrime - no slowing down @ONLINE. https://www.mcafee.com/enterprise/, March 2022.

[139] Phishing statistics @ONLINE. https://www.tessian.com/blog/phishing-statistics-2020/, March 2022.

[140] H. Rao, X. Shi, A. K. Rodrigue, J. Feng, Y. Xia, M. Elhoseny M, and L. Gu. Feature selection based on artificial bee colony and gradient boosting decision tree. *Applied Soft Computing*, 74:634–642, 2019.

[141] I. A. Nabi and Q. Yaseen. Spam Email Detection Using Deep Learning Technique. *Elsevier B.V*, 184:853–858, 2021.

[142] Email dataset 1: @ONLINE. https://www.kaggle.com/nitishabharathi/email-spam-dataset/, October 2021.

[143] Nazario's Phishing Corpora. Email dataset 2: @ONLINE. https://monkey.org/~jose/phishing/, August 2021.

[144] Enron Email Dataset. Email dataset 3: @ONLINE. https://www.cs.cmu.edu/~enron/, August 2021.

[145] Data report (2020). @ONLINE. https://datareportal.com/global-digital-overview, December 2021.

[146] Apwg report (2020). @ONLINE. https://apwg.org/reportphishing, December 2021.

[147] A. Acquisti, I. Adjerid, R. Balebako, L. Brandimarte, L. Cranor, S. Komanduri, P. Leon, N. Sadeh, F. Schaub, M. Sleeper, Y. Wang, and S. Wilson. Nudges for Privacy and Security: Understanding and Assisting Users' Choices Online. *ACM Computing Surveys*, 50(3):1–41, 2017.

[148] M. M. Moreno-Fernandez, F. Blanco, P. Garaizar, and H. Matute. Fishing for phishers. Improving Internet users' sensitivity to visual deception cues to prevent electronic fraud. *Computer-Human Behavior*, 69:421–436, 2017.

[149] I. You and K. Yim. Malware Obfuscation Techniques: A Brief Survey. In *International Conference on Broadband, Wireless Computing, Communication and Applications*, pages 297–300, 2010.

[150] H. R. Bonab and F. Can. A Theoretical Framework on the Ideal Number of Classifiers for Online Ensembles in Data Streams. In *Proceedings of the 25th ACM International Conference on Information and Knowledge Management*, pages 2053–2056, 2016.

[151] H. R. Bonab and F. Can. A Comprehensive Framework for the Number of Components of Ensemble Classifiers. *IEEE Transactions on Neural Networks and Learning Systems*, 14(8), 2018.

[152] L. Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.

[153] Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.

[154] D. H. Wolpert. Stacked Generalization. *Neural Networks*, 5(2):241–259, 1992.

[155] M. Ozay and F. T. YarmanVural. A New Fuzzy Stacked Generalization Technique and Analysis of its Performance. *Machine Learning*, 2013.

[156] P. Smyth and D. H. Wolpert. Linearly Combining Density Estimators via Stacking. *Machine Learning Journal*, 36:59–83, 1999.

[157] A. Zamir, H.U. Khan, T. Iqbal, N. Yousaf, F. Aslam, and A. Anjum. Phishing website detection using diverse machine learning algorithms. *The Electronic Library*, 38(1), 2020.

[158] G. Kamal and M. Manna. Detection of Phishing Websites Using Naive Bayes Algorithm. *Proceeding of International Journal of Recent Research and Review*, 11(4), 2018.

[159] M. Al-Sarem, F. Saeed, Z.G Al-Mekhlafi, B.A. Mohammed andT. Al-Hadhrami, M.T. Alshammariand A. Alreshidi, and Alshammari. An Optimized Stacking Ensemble Model for Phishing Websites Detection. *Electronics 2021*, 10:1285, 2021.

[160] V.E. Adeyemo, A.O. Balogun, H.A. Mojeed, N.O. Akande, and K.S. Adewole. Ensemble-Based Logistic Model Trees for Website Phishing Detectio. *Springer Nature Singapore Pvt. Ltd*, pages 627–641, 2021.

[161] O. Akanbi, A. Abunadi, and A. Zainal. Phishing Website Classification: A Machine Learning Approach. *Journal of Information Assurance and Security*, 9(2014):222–234, 2014.

[162] G. H. Lokesh and G. BoreGowda. Phishing website detection based on effective machine learning approach. *Journal of Cyber Security Technology* , 5(1):1–14, 2020.

[163] Y.A. Alsariera, A.V. Elijah, and A.O. Balogun. Phishing website detection: forest by penalizing attributes algorithm and its enhanced variations. *Arab. J. Sci. Eng*, 45(12):10459–10470, 2020.

[164] Data set. @ONLINE. https://archive.ics.uci.edu/ml/machine-learning-databases/00327/, May 2021.

[165] T. Kam. Random Decision Forests. In *Proceedings of the 3rd International Conference on Document Analysis and Recognition*, page 278–282, 1995.

[166] W. Hadi, F. Aburub, and S. Alhawari. A new fast associative classification algorithm for detecting phishing websites. *Applied Soft Computing*, 48:729–734, 2016.

[167] L. Machado and J. Gadge. Phishing sites detection based on C4.5 decision tree algorithm. In *In Proceedings of the International Conference on Computing, Communication, Control and Automation (ICCUBEA), Pune, India*, page 1–5, 2017.

[168] Y. Sonmez, T. Tuncer, H. Gokal, and E. Avci. Phishing web sites features classification based on extreme learning machine. In *IEEE 6th International Symposium on Digital Forensic and Security (ISDFS), Antalya,*, page 1–5, 2018.

[169] M. Babagoli, M. P. Aghababa, and V. Solouk. Heuristic nonlinear regression strategy for detecting phishing websites. *Soft Computing*, 23:4315–4327, 2019.

[170] E. Zhu, Y. Chen, C. Ye, X. Li, and F. Liu. Ofs-nn: an effective phishing websites detection model based on optimal feature selection and neural network. *IEEE Access*, 7:73271–73284, 2019.

[171] B. Alotaibi and M. Alotaibi. Consensus and majority vote feature selection methods and a detection technique for web phishing. *Journal of Ambient Intelligence and Humanized Computing*, 12:717–727, 2020.

[172] A. Taha. Intelligent Ensemble Learning Approach for Phishing Website Detection Based on Weighted Soft Voting . *Mathematics*, 9(21):2799, 2021.

[173] M. Dharani, S. Badkul, K. Gharat, A. Vidhate1, and D. Bhosale. Detection of Phishing Websites Using Ensemble Machine Learning Approach. In *ITM Web of Conferences 40*, pages 3–12, 2021.

[174] JUSTICE MINISTRY OF LAW and COMPANY AFFAIRS (Legislative Department). He information technology act, 2000 (no. 21 of 2000)@ONLINE. https://www.meity.gov.in/content/preliminary, May 2021.

[175] National Crime Record Bureau (NCRB). @ONLINE. https://ncrb.gov.in/hi/crime-in-india-table-addtional-table-and-chapter-contents?page=24, May 2021.

[176] Open Government Data (OGD). @ONLINE. https://data.gov.in/, May 2021.

Social Network Analysis and Mining (2023) 13:23
https://doi.org/10.1007/s13278-022-01023-2

ORIGINAL ARTICLE

Performance analysis of annotation detection techniques for cyber-bullying messages using word-embedded deep neural networks

Surajit Giri[1] · Siddhartha Banerjee[1]

Received: 30 July 2022 / Revised: 22 December 2022 / Accepted: 31 December 2022

Abstract

In recent times, online harassment due to cyber-bullying is significantly increased with the growth of social media users. Cyber-bullying is a technique to harass users using electronic messages. Many researchers attack this problem using natural language processing. Most of them detect whether a message is a bully or not. In this paper, multiple deep learning models are introduced to detect not only bullying messages but also the annotation of cyber-bullying. Annotation detection of cyber-bullying assigns a proper description in which category a message belongs. The advantage of annotation detection is to warn the user by giving an alert message with proper annotation when the user sends or posts a message on social media. If this feature is combined with popular social network sites like Facebook, Twitter, WhatsApp, etc., this can be an additional filter to alert the user that they are going to post or send a bullied message of which type. Social media messages are unstructured as it includes text, URL link, emojis, abbreviations, etc. Most of the previous works are conducted to detect bullying messages only considering important words in the text, neglecting the other attributes in the message like URL links, emojis, and abbreviations. In this paper, an advanced pre-processing technique is proposed by considering some of the attributes in the messages like URL, abbreviation, number, emojis, etc., to detect bullying messages. In this work, six models, i.e., three deep learning models combined with two different word-embedding models have been employed for annotation detection. The performances of each of these six models are measured twice, by employing traditional pre-processing, and proposed advanced pre-processing. The experimental results show that the advanced pre-processing works better in the case of all six models.

Keywords Cyber-bullying · Pre-processing · Annotation detection · Word embedding · Deep learning

1 Introduction

With the advancement in the communication field, online communities and social networks gain popularity to share data. Different online forums, blogs, and social networking sites have been used as common platforms for communication. Some of the users use these platforms in illegal and unethical ways to insult, humiliate, and threaten other users within these communities. Sending or posting illegal text or images over the internet via digital electronic devices to hurt or embarrass another person is termed cyber-bullying. This may include confidential information leakage, photograph manipulation, recording of physical assaults, etc. (Campbell and Bauman 2018). An increased level of anxiety and depression may be occurred due to the occurrence of cyber-bullying in childhood and teenage (Arseneault and Shakoor 2009). Different studies indicate that higher suicidal attempts, poor academic and work performance, degradation of physical and mental strength are directly related to cyber-bullying. A survey in Delhi shows that among 174 students with age between 11 and 15, 8% are involved in performing cyber-bullying and 175 are victimized by such events (Sharma et al. 2017). Another study found that 59% of youth in the United State are suffered from some form of cyber-bullying (Pew Research Center 2018). Not only teenagers,

✉ Siddhartha Banerjee
sidd_01_02@yahoo.com

Surajit Giri
girisurajit@gmail.com

[1] Department of Computer Science, Ramakrishna Mission Residential College (Autonomous), Narendrapur, Kolkata, India

Published online: 14 January 2023

SMS Spam Classification–Simple Deep Learning Models With Higher Accuracy Using BUNOW And GloVe Word Embedding

Surajit Giri, Sayak Das, Sutirtha Bharati Das, and Siddhartha Banerjee*

Department of Computer Science, Ramakrishna Mission Residential College, Narendrapur, West Bengal, India

* *Corresponding author. E-mail: sidd_01_02@yahoo.com*

Received: Mar. 16, 2022; Accepted: Nov. 30, 2022

Unwanted text messages are called Spam SMSs. It has been proven that Machine Learning Models can categorize spam messages efficiently and with great accuracy. However, the lack of proper spam filtering software or misclassification of genuine SMS as spam by existing software, the use of spam detection applications has not become popular. In this paper, we propose multiple deep neural network models to classify spam messages. Tiago's Dataset is used for this research. Initially, preprocessing step is applied to the messages in the data set, which involves lowercasing the text, tokenization, lemmatization of the text, and removal of numbers, punctuations, and stop words. These preprocessed messages are fed in two different deep learning models with simpler architectures, namely Convolution Neural Network and a hybrid Convolution Neural Network with Long Short-Term Memory Network for classification. To increase the accuracy of these two simple architectures, BUNOW and GloVe word embedding techniques are incorporated with deep learning models. BUNOW and GloVe are popular choices in sentiment analysis, but in this work, these two-word embedding techniques are tried in the context of text classification to improve accuracy. The best accuracy of 98.44% is achieved by the CNN LSTM BUNOW model after 15 epochs on a 70% - 30% train-test split. The proposed model can be used in many practical applications like real-time SMS spam detection, email spam detection, sentiment analysis, text categorization, etc.

Keywords: SMS Spam; Machine Learning; CNN; CNN-LSTM; Word Embedding; GloVe; BUNOW

http://dx.doi.org/10.6180/jase.202310_26(10).0015

1. Introduction

Short Message Service (SMS) is one of the most popular forms of telecommunication service around the world because of its affordability. According to [1], around 5 billion people can send and receive SMS and more than 200 thousand SMS are sent every second. 83% of SMS messages are read within 90 seconds. On average, SMS messages have a 98% open rate compared to 20% of emails. Text messages have a 209% higher response rate than emails. Because of these statistics, marketers prefer SMS as the primary form of advertising which leads to spam-ming. SMS spamming has become a serious problem for mobile subscribers. Spam SMS refers to unwanted text messages that are usually either someone promoting a product or service or someone attempting to scam a subscriber into providing personal information. It incurs substantial costs in terms of lost productivity, network bandwidth usage, management, and raid of personal privacy. The main reason to stop spamming is that it costs a lot more to its receivers than its senders. Because of these reasons and many others, SMS spam detection is very necessary.

SMS spam is a kind of problem that doesn't have an algorithmic definite solution. Existing SMS spam filtering methods are not very robust. The machine learning method comes to be the most popular choice for spam classification;