

# 2223\_CS-M20\_MSc Project

## Developing Classifiers to classify extremist content online

(Applying Sentiment Analysis to classify Extremist Content Shared Online)

Project Dissertation submitted to Swansea University in Partial Fulfilment for the Degree of  
Master of Science. Department of Data Science,  
Swansea University

5<sup>th</sup> January 2023



By:  
**Pallav Shukla**  
MSc. Data Science  
2154638  
2223\_CS-M20\_MSc Project

Mentor:  
**Dr. Muneeb Ahmad**  
Lecturer  
Department of  
Computer Science

# 1. DECLARATION

This work has not been previously accepted in substance for any degree and is not being concurrently submitted in candidature for any degree.

Signed ..... *Pallav Shukla* ..... (candidate)

Date .....5/01/2023

## Statement 1

This work is the result of my own independent study/investigations, except where otherwise stated. Other sources are clearly acknowledged by giving explicit references. I understand that failure to do this amounts to plagiarism and will be considered grounds for failure of this work and the degree examination as a whole.

Signed ..... *Pallav Shukla* ..... (candidate)

Date .....5/01/2023

## Statement 2

I hereby give my consent for my work, if accepted, to be archived and available for reference use, and for the title and summary to be made available to outside organisations.

Signed ..... *Pallav Shukla* ..... (candidate)

Date .....5/01/2023

## 2. Chapter 1: ABSTRACT:

Extremists frequently use the internet to spread propaganda, encouraging people to engage in radical content. Most social media platforms act as influencers for the young youth to get involved in this propaganda; usually, it is Facebook, Twitter and WhatsApp. To stop this unfortunate use of social media, many researchers in the Artificial intelligence and machine learning community are working on making a system that can predict these events. After getting an Engagement report of social media platforms - the Twitter platform was found to be highly engaging in terrorism scenarios. The prediction is achieved by developing models that predict whether or not the content shared is extremist. Therefore in this project, we built a model after exploring previously developed models based on various machine learning algorithms like (SVM) support vector machine, LSTM (long short-term memory) and CNN convolutional neural network, RNN (Recurrent neural network), PCA principal component analysis. Etc. Seeing the accuracy of the Newly formed model, we tried to improve the accuracy by using the transfer learning algorithm technique, which gave an outstanding accuracy for the prediction model. This project will be a gateway for future researchers to classify the Data as extremist and non-extremist, In a textual manner, images as well as in video form

### 2.1 Keywords

Social Media, Extremist Sentiments, terrorism, Deep Learning, Sentiment Classification, Twitter, machine learning.

### 3. ACKNOWLEDGEMENT

I have great pleasure in submitting this project entitled **Developing Classifiers to classify extremist content online** in partial fulfilment for the degree of Master of Science in Advanced Computer Science. While submitting this project report, I take this opportunity to thank those who are directly or indirectly related to the project work.

I would like to express my gratitude to my supervisor **Dr. Muneeb Ahmad** who has provided the opportunity and organised the project for me. With his active cooperation and guidance, I was able to complete this challenging task on time.

Sincere thanks to the technical and support staff in the department of Computer Science at **Swansea University** for their continuous help and guidance during this project.

I am grateful to my teachers for their excellent teachings that have left no stones unturned in enlightening me. I am at this stage of my life because of their efforts in empowering me.

While submitting this project, I wish to acknowledge my mother **parents and my brother** for their wonderful support and encouragement throughout the completion of this project. This project is an outcome of focused and sincere efforts that could be given to the project only due to great support from them.

I wish to thank my family members, friends and all those who have helped directly or indirectly in the successful completion of the project work.

**Pallav Shukla**

Student No.: 2154638

## 4. TABLE OF CONTENTS

|       |  |    |
|-------|--|----|
| 1     | Declaration .....  | 2  |
| 2     | Chapter 1: Abstract .....  | 3  |
| 2.1   | Keywords.....  | 3  |
| 3     | Acknowledgement.....   | 4  |
| 4     | Table of Contents .....  | 5  |
| 5     | Table of Figures .....   | 6  |
| 6     | Chapter 2: Introduction.....   | 7  |
| 6.1   | Introduction and Background Research for project.....                  | 7  |
| 6.2   | Aim.....   | 8  |
| 6.3   | Objectives .....   | 9  |
| 6.4   | Motivation.....  | 10 |
| 6.5   | About the Data Set.....  | 11 |
| 6.6   | Relevance and Scope .....  | 11 |
| 7     | Chapter 3: Literature Review - Related Works Project Description ..... | 12 |
| 8     | Chapter 4: METHODOLOGY .....   | 18 |
| 8.1   | Libraries Used.....  | 19 |
| 8.2   | Data Collection.....   | 20 |
| 8.3   | Cleaning the data .....  | 20 |
| 8.3.1 | Tokenization .....   | 21 |
| 8.3.2 | Remove Stop Words .....  | 23 |
| 8.3.2 | Remove URLs.....   | 24 |
| 8.3.2 | Lemmatizing .....  | 24 |
| 8.4   | Pre-processing.....  | 25 |
| 8.5   | Data Visualization .....   | 26 |
| 8.5.1 | Word Cloud .....   | 26 |

|        |                             |    |
|--------|-----------------------------|----|
| 8.6    | Model Building .....        | 27 |
| 8.6.1. | LSTM .....                  | 27 |
| 9      | Result and Discussion ..... | 32 |
| 10     | Conclusion.....             | 40 |
| 11     | References.....             | 41 |

## 5. TABLE OF FIGURES

|   |                  |    |
|---|------------------|----|
| 1 | Figure 1 .....   | 12 |
| 2 | Figure 1.a.....  | 15 |
| 1 | Figure 1.b ..... | 18 |
| 2 | Figure 2.....    | 19 |
| 3 | Figure 3.....    | 22 |
| 4 | Figure 4.....    | 23 |
| 3 | Figure 5.....    | 24 |
| 4 | Figure 6.....    | 25 |
| 5 | Figure 7.....    | 25 |
| 6 | Figure 8.....    | 26 |
| 5 | Figure 9.....    | 26 |
| 6 | Figure 10.....   | 27 |
| 7 | Figure 11.....   | 28 |
| 8 | Figure 12.....   | 29 |
| 9 | Figure 13.....   | 30 |

## 6. Chapter 2: INTRODUCTION

### 6.1. Introduction and Background Research for project

Over the past two decades, online social media platforms have given millions of users a way to interact with each other through various social media platforms like Facebook, Twitter, WhatsApp, Instagram, Reddit and many more. [21] With the increase in social media, there has been an increase in extremist content shared online. It may be in the form of comments, tweets, pictures, and videos that increase a person's negative behaviour, leading towards radicalisation. Learning from all the current research, it was found that social media platforms act as the fastest catalyst for spreading extremist content propaganda. [26] [27] Using such web apps Creates concern about their usage and also gives a chance for a hate group to get into participate [24] [25].

Extremism can be of any type from the various examples. One of the most famous tends to be terrorism Which in the past recent years has given inflation to the militant groups that start the involvement of propagation of terrorist content.[23] [28]. Instead of all the other types of extremism, the most important issue in the present scenario is the rise of various militant groups and associations that spread hate speech and extremist comments throughout the world with the help of these social media platforms.[28] This starts by creating groups that actively work at local levels and then starting to form at the community level, which is further preceded by the involvement of social media platform to increase the spread at a tremendous rate.

+

These social media platforms are filled with young youth generation, who are easily manipulated and get addicted to the growing rage with the extremism content acting as a source of brainwashing of their moral etiquettes. And as these social sites are much more approachable and user-friendly that serve as an excellent platform for propaganda that results in increasing of community strengthening of the group which in turn helps in in the spread. There has been a great impact on people's feelings of these because of these social media sites regular usage And also as these sites work as a venue for fundraising as they have a significant influence on people's mind.

The data information present on the site gives a sentimental clue about the Habits and actions of people and about their activities which gives a valuable window to understand content. With the help of this data open golden opportunity on understanding people sentiment and opinion words any activity that helps the researchers to understand the content. Scientists from all around the world are trying to develop new methods on treating or tool For analyzing these content to counteract And also to identify extremism on the networking site these scientists are specialists from various subjects like computer science social science and psychology.

In this project are main aim was to detect extremist content that can be in any form like hate speech or radical comment understanding the sentiment of the user by analyzing the data from the extremist group by doing so we can the make aware of these criminal activities to the required authorities That can help in reducing the extremism in the Peaceful world. In our research we found an extraordinary Project Which was built for an aim at identifying the difference in attitude of the people that can help to prevent any extremist activity. The name of this German project was MOTRA ("Monitoring System and Transfer Platform Radicalization"). [30]

Most of the research is done by counting the frequency of a term and by using a lexicon based dictionary that uses SentiStrength and SentiWordNet [1] [2]. Using this technique will give an output that is not a reliable solution as in a language a sentence may consist of semantic operations and which cannot be correctly predicted by using Just the Frequency count. Thus, the normal filtering tweets techniques for finding an extremist and non extremist coming was not scalable for the researchers and hence new techniques were developed by using machine learning algorithms[5] [6] instead of just approaching by traditional dictionary techniques. [7] [8]. The type of technique is widely used for identification of key players, link prediction and detection of groups. [3] For detecting the extremist community and there extreme behavior and activities on various social Media platform techniques like sentiment analysis and opinion mining using various machine learning algorithms are quite famous for these Networking platform [4] [5]

## 6.2. Aim

Very specifically our aim states that, to get a higher accuracy model to predict the extremist content shared on Online platforms like twitter, that can help various agencies to Curb any extremist activity for example terrorism before it happens.



## 6.3. Objectives

Our main objective leads to the practical implementation of our aim to be achieved, that is described above. This will include analyzing the data, Twitter comments with the steps of cleaning, labeling and processing it through various techniques of the model to get a higher accuracy. Applying different methods to enhance the model to achieve higher on different machine learning algorithms like LSTM.

However, There are many challenges associated with using the machine learning algorithms on Twitter dataset as it consists of different categorical data which make it difficult for applying the algorithm directly onto the data so we need to clean the data to get valuable information. This step is required for classification of the data in accordance with the most important features. Furthermore to enhance the quality of a result we will apply transfer learning algorithm that comprises of implementation of the model trained on two datasets that gives higher accuracy.

We have implemented LSTM (long short term memory) algorithm for the classification of the data to be extremist and non extremist. and then proceeded with the improvement of accuracy. That will help us to get an improved prediction filling the gaps of the previously done work.

The aims and objectives will be fulfilled by rigorous methodology and approach our goals the primary objectives will be as follows:

- (i). Searching a dataset: The data set used have a number of Comments with Structured and unstructured data
- (ii). Data cleaning and Data processing: this is the most important part of the project during the initial phase as this includes the meaning of textual data so that it can be processed carefully while applying the machine learning algorithms which will result in a higher accuracy for the model to be trained as well as during the testing time. In order to clean the data it requires multiple steps, like removing the blank data, removing special characters, links, hashtags etc. Data preprocessing involves multiple steps as well that helps to avoid overlap and ambiguity caused from it. This can be achieved by various methods such as Text Blob, NLTK (Natural language Toolkit), and the lexicon based sentiment analysis library Found in Python.
- (iii). Sentiment Analysis: This part comprises the main machine learning algorithm. For example (LSTM) Long Short Term Memory Is a type of Recurrent Neural Network that is RNN. LSTM Can be used over sequential data and is suited to be worked on analysis of sequences of words or characters. A general outline of applying LSTM on the dataset of sentiment analysis over tweets comprises of the following steps:

- (a). Preprocessing: Removing unnecessary characters and formatting the data.
- (b). Tokenization : Breaking processed tweets into words termed as tokens
- (c). Encoding : Tokens encoded as numericals
- (d). Model Training : After splitting, model is trained to predict sentiment
- (e). Model Evaluation: Finding the accuracy
- (f). Model Deployment : After evaluation the model can be deployed on new tweets to classify them as extremist and non extremist.

## 6.4. Motivation

Through this work we aim to analyze the data shared online on various social media platforms in the form of tweets comments and then predict whether the content shared is extremist or non extremist, and hence we will be able to make aware the required authorities for the emerging trend of negative influential extremism through the propaganda created by random terrorist community. This will help reduce hatred amongst people and hence resulting in creating a peaceful loving Society. This will also lead to the reduction of the spread of propaganda at an earlier stage and turning to an end of radicalisation in a safe manner.

With the help of this project many researchers will be able to understand the pattern of radicalisation at an early stage and will be able to curb the heated and agony created by the negative influential people like politicians and terrorists. And also to use this model for not only textual but also in images and video format. The main motivation for doing this project is to stop that hatred among human beings, countries and also to decrease the depression, anxiety and pressure created over young generations by brainwashing them through the propaganda of hatred speech.

## 6.5. About the data set

During the research we found that most of the analysis was done on Twitter dataset as Millions of users used Twitter in daily life and as a wide usage of Twitter in the young generation leads to a tremendous use by militant groups to influence the youth, for the involvement in terrorism. In a study it was found that about 100k accounts were being used for terrorist activities in 2015 [9] [10] [11]. Twitter being actively used as an engaging platform for the terrorist groups serves researchers as a golden opportunity to apply different automated systems on the data information to identify extremism.[9] As Twitter has comments in very short text called as Tweet. The maximum size of a tweet is of about 250 characters which was increased by 140 characters in the year 2017. It is a very challenging task to apply sentiment analysis on short tweets as on a very small text we have very less contextual information that is quite opposite as compared to the other social media platforms.[28] Twitter comments are hard to analyze also because some of the comments are unstructured and some of them are structured that are needed to be cleaned and refined before using in a model. Many of the comments are misleading as by comparing a normal day-to-day scenario with an extremist word. During this research project, the dataset used was from Twitter and has been worked on for data processing and cleaning and then applying sentiment analysis. No particular community or group was targeted during the analysis part. All the information that is used in this project is kept safe as this project consists of sensitive information that is being publicly available by Twitter on kaggle [12] for researchers to work upon. One of the data sets used was unlabelled and we tried to label it by analyzing the hate speech present in the comments.

## 6.6. Relevance and Scope

With the help of this classifier People will be able to understand the correctness of any extremist data whether the comment is extremist or non extremist content. Through sentiment analysis an approach will be created for both personal as well as business, For example in personal life people can understand the propaganda approach of any news and for business mind this can be valued, by the the news agencies whether to classify the content as extremist and non extremist of fake news. With the model created, people can understand the relevance of the data and can analyze the approach of terrorists groups on how they engage young generation to get involved in the radicalisation and can be secured from the brain wash system created by the the militant groups. This model will also be an initial step for the future researchers to work on analyzing on online social Media platform comments.

## 7. Chapter 3: LITERATURE REVIEW – RELATED WORKS:

We have reviewed some relevant studies conducted on the classification of social media-based content which reflects extremist connections. This gave a better understanding of the existing work already achieved for identifying content as extremist or non-extremist.

NLP ( Natural Language Processing ) and SA (Sentiment Analysis) have advanced gradually year by year. Figure 1 [41] depicts how machine learning has progressed over time in order to better analyze the extreme contents from the web. K Nearest Neighbors, Naive Bayes, data clustering , EDA , DNN (Deep Neural Network) and GBDT ( Gradient Boosted Decision Tree ) are few of the most popular techniques for extremism detection on social media sites [42-48].

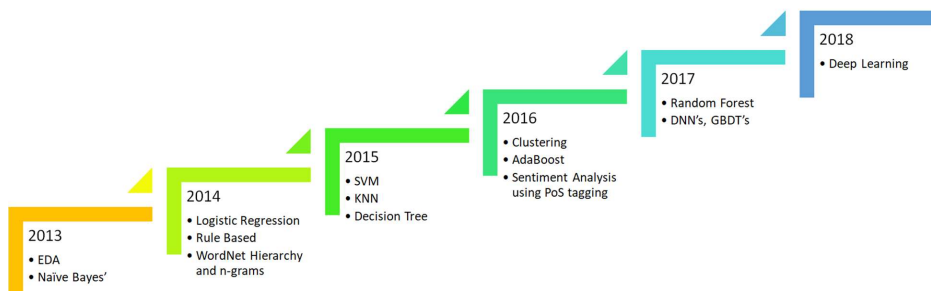


Figure 1 - Machine Learning techniques used over time.

Wei et al. [32] uses a machine learning based classification system for content which signifies extremism on Twitter. Many features are analyzed for finding unusual behavior on tweets on twitter given by users via KNN classifier. Similarly Azizan and Aziz[19] conducted a study for the detection of extremist content using NBA(Naive Bayes algorithm) . This algorithm shows best results among other machine learning classifiers. Here the authors have applied ML classifier with classical features. They were able to classify user reviews into positive and negative sentiments of the extremist groups. Classification into positive and negative does not give us an accurate way of distinguishing between extremist and non extremist content. The classification did not include all the dependencies related to a sentence in one record. Hence this model could not be much useful for our classification with current design.

We investigated some deep learning-based sentiment analysis research which seem to be very promising when in different fields such as speech, vision and text analytics [33][34]. In this research they proposed a multi-channel convolutional neural network-long short-term memory (CNN-LSTM) model consisting of two parts: multi-channel CNN and LSTM to analyze the sentiments English tweets from Twitter. Unlike a conventional CNN, they have applied a multi-channel strategy using several filters of different length to extract active local n-gram features in different scales. LSTM then sequentially composes this information. This paper overcomes the limitation of [19] by combining both CNN and LSTM. The model was able to consider local information within tweets and long-distance dependency across tweets in the classification process.

We came across an interesting research work by Matthias Hartung et al [35]. They used Support Vector Machines with a linear kernel [36] and were able to train their model to detect right-wing extremist users in German Twitter profiles. Their work supported manual monitoring aiming at identifying right-wing extremist content in German Twitter profiles. They did profile classification (based on textual cues), traits of emotions in language use, and linguistic patterns. They were able to reduce 25% of manual labor with their achievement of results. Although the work could have used better deeper methods of NLP in order to be able to address more fine grained aspects.

Patil et al [37], explores feature vectors from LSA (Latent Semantic Analysis) and CNN (Convolutional Neural Networks) classifier. Various text classification techniques such as Deep learning, Ensemble learning as well as effective document representation methods are trending to improve the accuracy of text classifiers [38]. Terrorist extremist contents spread jihadist propaganda and increase their believers via social media. The work [39] proposes a deep learning approach to detect extremist contents and the results provided better accuracy for identifying such cases for classifying the contents automatically, which have terrorist activity contents. Some unigram methods used for feature extraction and for sentimental classification they used SVM and NB algorithms which classifies that user's opinion is positive or negative [40] and thus are able to observe client's opinion on social media.

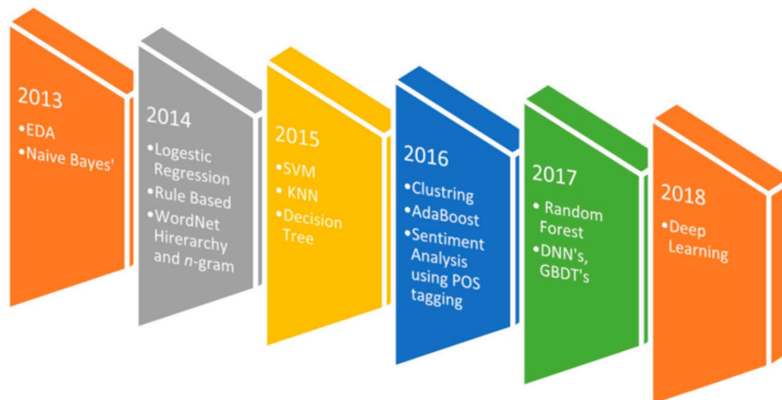
In [49], Authors have combined sentiment analysis with Social Network Analysis to analyze radical groups on YouTube. They have crawled from a group of 700 YouTube accounts. Then the analysis was done on different topics aligning to different polarity to identify sign of extremism and intolerance. The authors had used a lexicon based module to determine the main topic. Sentiment analysis was used on top of this to identify the opinion of users towards these topics. Two different results for men and women are drawn to identify the most positive and most negative topic in both categories. As per the result, women were found to be more positive toward Al-Qaeda and negative towards Judaism whereas for men, higher positivity on Islam was identified as per the results.

Another approach which was presented in [50], where the tweets were grouped into different groups depending on certain special keywords such as "Al-Qaida", "Jihad", "Terrorist Operation"

through a lexicon based approach. They created their own dictionary of semantics based on hashtags in tweets. Classification was done on vectorized tweets based on dictionary related words. Customized set of rules were made for each category and the tweet were classified into that category.

In this section we will add on to our background research and have gone through a number of relevant studies based on the social media classification over the content in the form of comments that reflect extremist links. From this literature review we will get to know a clear and better understanding of the already done work on identifying the content as extremist and non extremist. We have mainly focused on various ways to depict the classification of the data mainly achieved through machine learning algorithms And how they are improvised to get higher accuracy from the model.

Over the past decade there has been a great advancement in NLP ( Natural Language Processing and SA ( Sentiment Analysis) that shows how algorithms from machine learning have revolutionized in order to get better analyses of the extreme comments from social media platforms This can be seen in the following figure 1. [13]



**Figure 1.** Machine learning techniques used for online radicalization and terrorist detection.

*Figure 1.a*

Wei et al. [14] Need a system that signifies extremism on the Twitter dataset beef on the classification machine learning model During the desert unusual behavior from the comments of Twitter tweets was done by why using a classifier KNN ( k - nearest neighbor ) in this many features were analyzed forgetting the findings. In another research performed by using NBA(Naive Bayes algorithm)The otter got the best result among rest of the machine learning classifiers the study was conducted for finding the extra content and was done by Azizan and Aziz [5].

Using this model they were able to classify the data into positive and negative sentimental comments in accordance with the extremist groups as they have applied classical features from the applied machine learning classifiers. Receiving an output as positive and negative comments does not give an accurate result for extremist and non extremist comments. Due to the lack of dependencies in this model for a sentence this classification will not be able to survive all the comments of extremism. Therefore this model will not be useful for our classification design.

In another research very exciting result was found in a variety of fields such as speech, Vision and Text Analytics was been predicted using deep learning based sentiment analysis.[15][16] In this research a combined model was created Using multi-channel convolutional neural network CNN and long short term memory LSTM model that consisted of two parts ( CNN and LSTM ) to analyze the English tweet's sentiment from the Twitter dataset. Instead of using the normal LSTM and normal CNN model they used a strategic model consisting of a multi-channel approach to extract n-gram features Of different length by using discrete features and at different levels. LSTM then forms the information in sequential manner. This approach was quite interesting and was seemingly a better approach for the project as it gave local information a chance, and during the classification process was able to show long-distance dependency across the tweets. This literature wasn't advancement over the last [5] research paper

Matthias Hartung et al [17] gave an interesting research by using a Support Vector Machine SVM with a kernel of linear[18] system and the model trained was able to detect Right wing extremist users in German Twitter profiles. The whole classification was based on textual clues and was detecting profiles based on the emotions in language and their traits, and also using the linguistic patterns available. The main aim for their research was to identify right - wing German Twitter profiles by manual monitoring. From this research they were able to achieve a result of, a reduction of, 25% manual labor. They could have achieved a better achievement by using Natural Language Processing NLP which would have been able to get a higher achievement because of a more refined and precise model.

We came across an interesting research Patil et al [19], In which, classifier explorers feature vectors with help of Convolutional Neural Network CNN and LSA Latent Semantic Analysis. For improving the accuracy of text classifiers various techniques are being used such as Ensemble learning, Deep Learning and also various effective document representation methods are trending. [31] Jihadist propaganda spreads the comments of extremists which increases and give a rise to the believers of extremism through social Media platform. For finding the extremist comment automatically and then classifying them and to get a better result, in this work [32] deep learning technique was used to detect the extremists comments that comprises terrorist activities. In this research the author used unigram methods, Support Vector machine SVM and NB Navies Bias algorithm Was used for feature extraction sentiment classification and also so for the identification of users opinion as positive and negative [33] respectively. Therefore it was a good approach for finding users opinion on on multimedia platform.

In another research[34] based upon men and women over the topic Al-Qaeda, Judaism and Islam On the platform of YouTube Where they have combined sentiment analysis on the basis of radical groups with the help of social network analysis near about 700 YouTube accounts checked. The research was pleased on the identification of which side as a sign of extremism or intolerance. the authors used lexicon model to identify the heading, and to find the opium the topic sentiment analysis was being used as a result over men and women, woman were found to be involved

more in Al-Qaeda And less involved in Judaism whereas men were found to more linked towards Islam

In another approach[35] various groups were performed according to the keywords like “Al-Qaida”, “Jihad” and “Terrorist Operation”. In this approach tweets were classified, organized / vectorized according to the dictionary. The dictionary in this approach was formed by the researcher from their hashtags of tweets with the proper set of rules for each category then they were classified into their respective category. This was an excellent approach for creating their own dictionary, made them more reliable and discreet.

The research [36] For finding Sunni Extremist Propaganda with Deep Learning gave a better understanding of the project approach. The paper was linked to the finding of spread of jihadist propaganda and the recruiting of new members on social media platforms like the internet, Darknet etc. The approach is simple to find the sub-links of web pages and social media content that contains extremist content using the neural network and deep learning techniques. For the approach the head used word frequency feature selection techniques and and preparation remove contacts in semantics then developed “word2vec” and “doc2vec” for a better outcome. The F1 score was 0.93 and for the classifier the accuracy was 93.2 percentage In this paper a future work was mentioned as to combine two datasets and then applying the technique which can be a better approach for our present extremists and non extremist finding Project.[36]

In the paper[37] The author tries to find the prevention of extremist and terrorist activities by quickly detecting the content of extremism with the help of machine learning algorithms and automatic recognition. At first the NB (Naive Bayesian classifier) and point-to-point mutual information (PMI) method was used for classification. The classes used were drugs, violence, nationalism and extremism, etc.[37]. Which gave some result but this approach was not satisfactory, so they tried a different approach based on lexical features the algorithms used this time were SVM and KNN Support Vector Machine and k-Nearest Neighbour respectively. In this Twitter was used as a social Media platform listing the tweets of about 10,000K as a training sample. The accuracy achieved was 90%. This type of approach seems good for a simple dataset for achieving higher accuracy other methods like Transfer learning algorithm could have been used. From this paper we got an idea, that higher accuracy can be achieved for resulting in more accurate results.

During a research[38] we found an extra ordinary paper that aims at finding medical content over social Media platform using the textual psychological and behavioral factors the whole study was classified into three categories first Analyzation of propaganda that was based on text based model of radical content second Model for psychological properties and third evaluation of these model on Twitter dataset to identify online radical tweets.



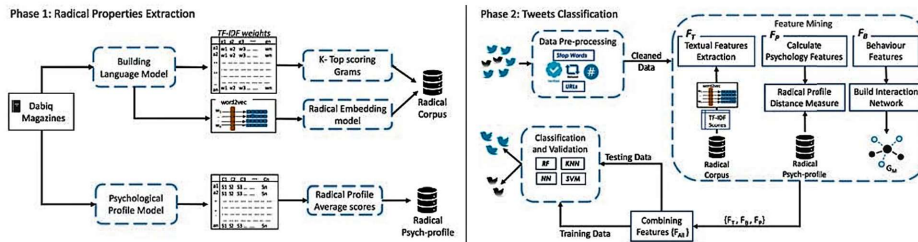


Fig. 1: Approach overview

Figure1.b: The process used in this approach[38].

The figure shows the approach applied in this research. Many classification algorithms were used during the research for example random Forest neural network support vector machine nearest neighbor and as a result random forest and neural network perform the best with an accuracy of with an accuracy of 91% and 100% respectively. The approach of drawback the model was not trained on Twitter comments that can work as a loophole of the system. [38]

In another research[39]A new approach was being used known as Composite based method.The otter Inn at real-time monitoring of the extremist activities and find it in an automated system through classification this is called composite technique in this both semantic and syntactic features of textual contents of a web page are considered. A classification model was developed using J48 decision algorithm They can be classified into appropriate results The model used give an accuracy of 96% success rate in classifying the web pages[39] In this approach 7500 web pages That would be classified as pro extremist neutral and anti extremists were taken as dataset. In this approach a lexical resource Sentistrength Was used to generate a sentiment value for each page under the sentiment analysis section. Whereas in the composite method a custom written script was merged with the semantic features derived from the sentiment analysis and the syntactic features received from Posit. Surprisingly the accuracy of this model was wayahead to 96%.[39]

In another approach[40] through sentiment analysis classification was done. At first TENE web crawler was initially used for manual classification, Then automated system for classification of extremist websites Were done. A feature selection algorithm was able to reduce 26 original features and to attain an overall performance of 94% For the classification of web data[40] Methods involved Sentistrength, feature extraction process, supervised data mining algorithm for developing a regression model in a tree format J48 decision tree classification algorithm was used. The paper consisted of some drawbacks as as the model train in the initial phase work done through manual approach.[40]

From the above given literature reviews we understood the pattern of classification algorithm being used, how they are being implemented, what's the accuracy, and the process step-by-step method that varied from every paper to paper, this gives an idea about how to implement the algorithm on our dataset and how to improvise the accuracy Finding the extremist and non extremist content over online social media platforms.

## 8. Chapter 4: METHODOLOGY

This section would cover the methodology which has been used for the analysis of Tweets gathered. We have used multiple techniques of NLP used for this project, such as vectorization of data and using LSTM. NLP stands for Natural language processing and is a branch of AI or artificial Intelligence. NLP is based on combining rule-based modelling of human language with various machine learning and deep learning models (IBM). The goal of using NLP is to train the computer program "to understand" the natural language, that is, human language with sentiment and intention. Here, in methodology section we are covering the implementation we have done on the tweets data which we have gathered. We have used multiple libraries in our code which we will explain in methodology, such as text blob, word cloud[53], and matplotlib (Brownlee, 2021). Initially we implement the various ETL techniques on one dataset which is a small manual dataset. We build the model for this data and then we apply the transfer learning algorithm to combine the datasets manually with larger dataset and then check the efficiency of the model. We check the polarity of our classifier based on input given to classify the data into "Extremist" or "Non-Extremist"

An overall design of our model looks like:

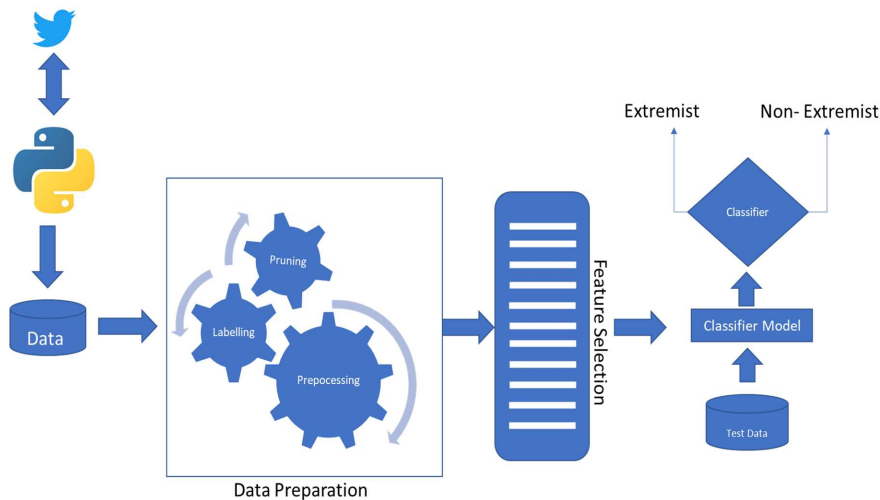


Figure 2: Proposed design [51,52,41]

Here we will explain the steps of our analysis. These steps are data collection, data pruning, pre-processing, feature extraction and EDA. Figure 1 above represents the proposed process for our work.

Some sub – sections that we will cover in the methodology.

- data collection
- data cleaning,
- data pre-processing
- data analysis
- model building.

## 8.1 Libraries used

Various libraries are used classification of the extremist content. text blob(Textblob) is one of python's natural language processing library which uses a NLTK ( natural language toolkit ). This helps in speech tagging. We do the wordnet integration using the “word” property from textblob library. (<https://textblob.readthedocs.io/en/dev/quickstart.html>). Textblob has other features as well like extraction of the noun phrase, tagging the parts of speech, splitting data in words tokenization, frequencies of words, and n-grams used to convert the word into the vector ( 42. Even more features of textblob are - lemmatization and pluralization(from singular to plural or vice-versa) or text blob can also be used for spell correction. We use textblob to get the WordNet.

We use textblob library for word cloud. These are the features used for visualisation of the frequency of the text or word. The pattern of the dataset can be found out using word cloud. NumPy and Pandas are the mostly used libraries. These standard libraries are very powerful and useful for open - source data handling. Although these libraries' first task is mostly importing tabular data into python. Various types of pandas functions handles mostly issues like - handling the missing data, data which is inserted, delete data or merging or joining, slicing, etc. NumPy is mostly used for manipulation of data via arrays. Matplotlib library is used for visualization or plotting the graph. NLTK This library uses for tokenization or spitting the text into words. That is used to filter the stop words such as is, in, etc. Stemming is yet another feature of NLTK which reduces the size of the word by using the root of the word, for example 'coming' becomes 'come' , 'running' becomes 'run' , etc. 'tag' is another package present in nltk which tags data into various parts of speech . This is also called part-of-speech tagging, POS-tagging, or simply tagging. [41]). We have used sklearn library for encoding the data [43]

For model building we've further utilized various other packages of keras and sklearn libraries which we'll discuss further.

## 8.2 Data collection:

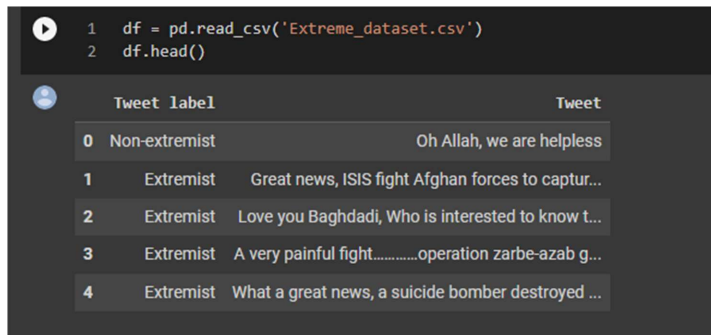
First dataset we have about 21 thousand unique entries and the second dataset we have which is a mixed dataset has around 55 thousand entries. Data is cherry – picked from various sources of twitter data. The dataset contains labels which identifies a record as 'Extremist' or 'Non-

Commented [DL4]:

Commented [DL5]:

Extremist' which will help to train the model. The dataset contains various tweets collected from twitter with consideration of various tweets to maximize the accuracy of the model with the predictions. Polarity of the tweets for our model defines the sentiment of the tweets. Twitter dataset was used for data collection, as this trending social media platform has variety of tweets for training our model with many people's sentiments. Social media has always been the datahub for the researchers for raw data.

Twitter has approximately 450 million users per month who definitely contribute in large amount of data. Thus, twitter becomes an excellent and significant source to extract the data. Data is usually gathered by API, Application Programme Interface. API is the mediator that processes the input request received from the user into the required response expected. Twitter API allows the user to extract tweets by a developer. The data is extracted by using the programme. Different types of APIs can be developed to extract the data. The data collected is too raw to be used, hence we have to clean the data to make our model as efficient as possible.



```
1 df = pd.read_csv('Extreme_dataset.csv')
2 df.head()
```

|   | Tweet label   | Tweet  |
|---|---------------|--|
| 0 | Non-extremist | Oh Allah, we are helpless                          |
| 1 | Extremist     | Great news, ISIS fight Afghan forces to captur...  |
| 2 | Extremist     | Love you Baghdadi, Who is interested to know t...  |
| 3 | Extremist     | A very painful fight.....operation zarbe-azab g... |
| 4 | Extremist     | What a great news, a suicide bomber destroyed ...  |

Figure 3 Raw Data collected

### 8.3 Cleaning the data:

Pruning[58] goes back to 1990 in Yann LeCun's paper [57]. It denotes the removal of not relevant terms in the data set which may ultimately reduce the performance of the classifiers. Such data is often referred to as "noise" or "undesired data". Therefore such words must be removed from the corpus. Below are the few steps which shall be taken for noise removal.

Data Cleaning is always recommended required to while training any model be it supervised or un - supervised. Dataset can consist of different forms, it can be an image dataset or a video dataset, it can be audio dataset, text dataset etc. Usually when the dataset consists of text, like in our case, natural language processing and the combination of machine learning techniques are applied for semantic analysis. Sentiment analysis allows us to understand the dataset's pattern which is a reflection of user's sentiments. With data cleaning we aim to clean the data of

unwanted data using various methods. These methods we will remove the punctuation and stop words.

This is our first phase of text cleaning progression. We will have be removing all hash ( # ) mentions, all HTML tags if any, punctuations, non - alphabets that are numeric, and other characters that are not needed. Regular Expression ( RE ) , is one of the most common cleaning technique used to clean the data or prune the data. Therefore, the many available symbols in English, stop words, basically everything which will not affect the weight of our classification shall be cleaned from the data. Ambiguous sentences makes it difficult for the classifier.

Sometimes the sentences contain words which might have different meanings. The same word could mean different depending on different sentiments. Even existence of apostrophe ( ' ) words such as , "Messi's" might also impact the model's training and predictions.

```
Removing mentions and hashtags

[ ] 1 import re
    2 def remove_hash_mentions(x):
    3     x = re.sub("@[A-Za-z0-9_]+", "", x)
    4     x = re.sub("#[A-Za-z0-9_]+", "", x)
    5     return x

[ ] 1 print(remove_hash_mentions("@Demonslayer what's up man !!!!! #cool"))

    what's up man !!!!!

[ ] 1 df['Tweet'] = df['Tweet'].apply(lambda x: remove_hash_mentions(x))
```

*Figure 4 Removing Hash mentions*

### 8.3.1. Tokenization:

A tokenizer breaks down unstructured data or raw text into pieces of information that can be considered as individual elements. In machine learning, tokenizer can be used to turn unstructured text into numerical data structure suitable for machine learning. For example if we can use the NLTK tokenizer on a sentence such as if this sentence "This is the last ray for humans ! #Hope #hoom.N" is tokenized (one of the many ways) then it would be ["This", "is", "the", "last", "ray", "for", "humans", "Hope", "hoom.N"]

### Tokenization

Now, with the help of a tokenizer we'll break down all the sentences/words of the text into small parts called tokens.

We need to convert the text into an array of vector embeddings. This is needed so that our machine learning model understands the inputs. Word embeddings provide a beautiful way of representing the relationship between the words in the text.

Tokenize and converting the tweets into numerical vectors.

- Num\_words – This hyperparameter refers to the number of words to keep based on the frequency of words.
- Split – This hyperparameter refers to the separator used for splitting the word.
- pad\_sequences() function is used to convert a list of sequences into a 2D NumPy array.

NOTE: 0 is a reserved index that won't be assigned to any word. (from

[https://www.tensorflow.org/api\\_docs/python/tf/keras/preprocessing/text/Tokenizer](https://www.tensorflow.org/api_docs/python/tf/keras/preprocessing/text/Tokenizer)) The `fit_on_texts()` method helps to create an association between the words and the assigned numbers. This association is stored in the form of a dictionary in the `tokenizer.word_index` attribute.

Now we would replace the words with their assigned numbers using the `text_to_sequence()` method.

[https://www.tensorflow.org/api\\_docs/python/tf/keras/preprocessing/text/text\\_to\\_word\\_sequence](https://www.tensorflow.org/api_docs/python/tf/keras/preprocessing/text/text_to_word_sequence)

```
[ ] 1 tokenizer = Tokenizer(num_words=500, split=' ')
    2 tokenizer.fit_on_texts(df['Tweet'].values)
    3 X = tokenizer.texts_to_sequences(df['Tweet'].values)
    4 X = pad_sequences(X)
```

[Tokenizer](#)

*Figure 5 Tokenization applied*

Before tokenization we even convert all our required dataset to all lower. Model cannot differentiate between upper and lower case and will consider "APPLE" and 'apple' as 2 different words. Therefore to increase the efficiency of the model we convert our dataset to all lower.

## ▾ Cleaning

<https://catriscod.com/2021/05/01/tweets-cleaning-with-python/>

## ▾ To lower case

Converting all to lower case

```
1 #lowercase
2 df['Tweet'].str.lower()

0          oh allah, we are helpless
1  great news, isis fight afghan forces to captur...
2  love you baghdadi, who is interested to know t...
3  a very painful fight.....operation zarbe-azab g...
4  what a great news, a suicide bomber destroyed ...
...
21181  baghdadi... our last hope, i simply love you #isis
21182      we condemn a suicide attack in peshawar today
21183          oh allah, destroy us and israel
21184  a very painful fight.....clean up operation gav...
21185      we condemn a suicide attack in peshawar today
Name: Tweet, Length: 21186, dtype: object
```

*Figure 6: Converting to lower case*

### 8.3.2 Remove stop words:

Natural language almost all the time contains stop words such as “the”, “an”, “a”, “am”, “to”, “how”, etc. These words are unrelated and lead to increase the dimensionality of features. The computational complexity of classification models is increased because of this increasing dimensionality. These are a set of frequent words which carry less important meaning. Therefore, to reduce the training time and memory overhead, we remove such non-informative words. Therefore, we can use the stop-word list for English from the source [59].

Not always stop-words need to be removed, for example, when music lyrics are the data set then stop word removal is not beneficial as we might lose semantics for the quotes. Majority of the cases dropping the stop – words does not cause losing semantic meaning. One of the ways is to count the number of words and note their frequencies which will help to remove the duplicated words. Then we can use STOPWORDS from the ‘wordcloud’ library to filter the dataset.

#### ▼ Removing punctuations and alphanumeric

```
[ ] 1 def remove_punc_alphanumeric(x):  
2     x = re.sub('[()!@]', ' ', x)  
3     x = re.sub('[\.\*?\']', ' ', x)  
4     x = re.sub("[^a-z0-9]", " ", x)  
5     return x  
  
[ ] 1 df['Tweet'] = df['Tweet'].apply(lambda x: remove_punc_alphanumeric(x))
```

*Figure 7 Removing Punctuations and alpha - numeric characters*

### 8.3.3 Removing URLs

We'll use Beautiful soup library to remove the html tags as well as to remove URLs we'll develop our own function to remove any links using the same "re" library ( regular expression ) .

#### ▼ Removing Links

```
1 def remove_links(x):  
2     x = re.sub(r"http\S+", "", x)  
3     x = re.sub(r"www.\S+", "", x)  
4     return x  
  
1 df['Tweet'] = df['Tweet'].apply(lambda x: remove_links(x))
```

*Figure 8 Removing the links and URLs*

### 8.3.4 Lemmatizing:

Lemmatization reduces the word to its root form, such as "coming" is converted to the string "come". This helps to reduce the dataset by cleansing caused as a result of lemmatization. Stemming stems or removes the last few characters from a word which might often lead to incorrect meanings and spelling. In comparison to stemming, lemmatization is considered as a more promising approach with excellent results. The reason lies in the fact that lemmatization considers the context and then the word is converted to its base form. For example 'Caring' is converted by stemming to 'Car'. We must be very careful of the training accuracy of the model while applying lemmatization. As the words are reduced it can be the case that the real meaning of the sentence is changed which might lead to incorrect predictions. Hence, we should be careful while using lemmatization.



#### LEMMAIZATION

```
[ ] 1 df['Tweet'] = df['Tweet'].apply(lambda x: ' '.join([Word(x).lemmatize() for x in x.split()])))
```

Figure 9 Lemmatization

## 8.4 Pre-processing:

There are many methods of pre-processing the data, here we have applied label encoding as follows, using the sklearn library's preprocessing package.

We define an object for the LabelEncoder() and then utilize the fit\_transform function to encode the labels. This allows to feed in values for the model.

#### Data Pre-processing

```
[ ] 1 df['Tweet label'].unique()
```

```
array(['Non-extremist', 'Extremist'], dtype=object)
```

```
[ ] 1 from sklearn import preprocessing
2 label_encoder = preprocessing.LabelEncoder()
3 df['encoded_label']=label_encoder.fit_transform(df['Tweet label'])
```

```
[ ] 1 df['encoded_label'].unique()
```

```
array([1, 0])
```

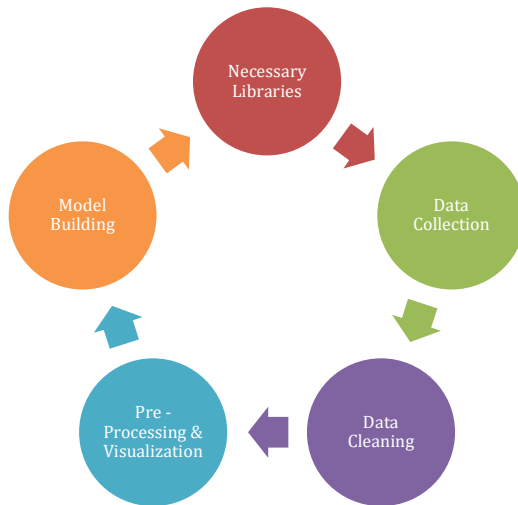
```
[ ] 1 df.head()
```

|   | Tweet label   | Tweet  | encoded_label |
|---|---------------|--|---------------|
| 0 | Non-extremist | Oh Allah, we are helpless                          | 1             |
| 1 | Extremist     | Great news, ISIS fight Afghan forces to captur...  | 0             |
| 2 | Extremist     | Love you Baghdadi, Who is interested to know t...  | 0             |
| 3 | Extremist     | A very painful fight.....operation zarbe-azab g... | 0             |
| 4 | Extremist     | What a great news, a suicide bomber destroyed ...  | 0             |

Hence 1 denotes "Non-extremist" and 0 denotes "Extremist"

Figure 10 Pre - Processing by label encoding

We continuously use new libraries to improve the model and the cycle below continues until we reach the required accuracy of the model.



*Figure 11 Cycle flow Chart of Methodology*

## 8.5 Data visualization:

Data visualization that is good for understudying the proper dataset and pattern of the dataset here below some visualization present that helps understand the pattern of data.

### 8.5.1. Word cloud:

This visualization shows the word's frequency that is greater the size of the word the greater the frequency of the word.

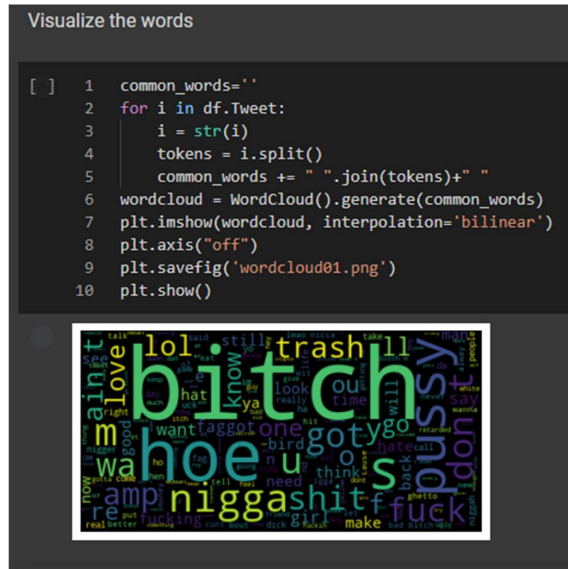


Figure 12 Word Cloud representation - Visualization of data

The visualization shows clearly that large font indicates more frequency of the word, and less size word represents the less frequency as shown in the Figure 11.

## 8.6 Model Building

Extremist or non-Extremist sentiment analysis is being developed to understand and forecast any mis – happenings which we will be able to predict from monitoring the social media tweets. Similarly many organizations are developing similar solutions to understand their client's interests in various products by analyzing the shopping trends with continuous feedback and evaluation.

### 8.6.1. LSTM:

LSTM concept was first proposed in 1997 by Hocheriter&Schmindhuber. LSTM stands - for Long Short-Term Memory (LSTM). LSTM captures a Long-term reliance, which is a type of RNN network. LSTM are widely utilised for a multiple types of projects like natural language processing, subjectivity analysis, text categorization, and so on. Here, we have developed a machine learning model using LSTM Recurrent Neural Network to classify tweet into extremist or non - extremist.

As the usual models do not store memories, they cannot analyze sequence information. Because of this the input and output is also constant. These systems cannot be used to resolve time series prediction and such related issues. Therefore, Recurrent Neural Networks (RNN) was established. RNN collects serial/series data for a period. In RNN, the structures / layers extend with the preceding state's inputs ( $w_1$ ) and initial state's output ( $w_2$ ). Then, Tanh function separates both. To have output sequence, we simply combine the separate state with a Tanh function output. However, drawback of RNN is its degradation problem, that is it produces significant weight increase which do not let the model train. To overcome this, LSTM was developed.

Recurrent neural network is supervised deep classification technique. The cells in RNN are time-linked to one another. RNN's only goal is to retain the knowledge in primary cells so that these neurons can transfer data amongst themselves in next layers for evaluation. This means that the one-time instance ( $t_1$ ), the data is transferred to the following time instance ( $t_2$ ). Vanishing gradient is one of RNN's major drawback. When any model is trained, during the training phase, the parameters are updated by estimating the loss and then back-propagating it through networks. However, such back propagation is not very easy. The issue lies in calculation of such weights which is quite complex and not accurate. This is because the slope of every single instance must be divided by the weight of the cable network. The gradients become weaker and weaker as we move deeper back in time to calculate the weights. This is the reason of vanishing gradient. The learning rate of an algorithm is directly proportional to gradient descent that insignificant gradient value will have little effect on the learning rate.

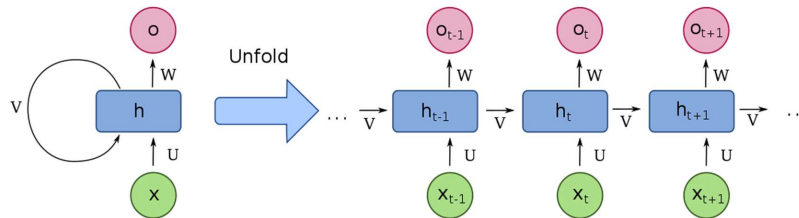


Figure 13 Recurrent neural network unfolds. SVG([En.wikipedia.org](http://en.wikipedia.org). 2021)

In general:

New weight = Old weight – ( Learning rate \* gradient )

Therefore, if the gradient is too insignificant then new weight will be equal as old weight and thus no significant learning. Hence, in long sequences it becomes difficult for RNN to carry information from one instance to next.

LSTM (Long Short-Term Memory) to the rescue ! (<https://medium.datadriveninvestor.com/how-do-lstm-networks-solve-the-problem-of-vanishing-gradients-a6784971a577?qi=7632bdb59f41> )

To overcome the problem of RNN, the architecture of the LSTM is shown below,

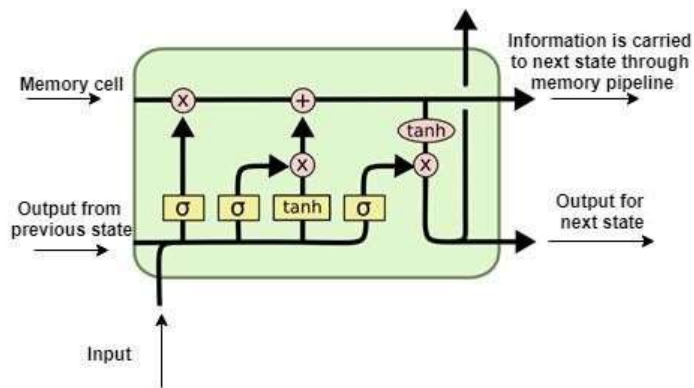


Figure 14 UNDERSTANDING LTMS NETWORK(COLAH.GITHUB.IO)

When contrasted to Figure 12 we can see that here we can recall more information from prior iterations and therefore overcome the drawback of RNN, that is the vanishing gradient problem. Our project therefore uses below model design as mentioned:

```
[ ] 1
2 model = Sequential()
3 model.add(Embedding(500, 120, input_length = X.shape[1]))
4 model.add(SpatialDropout1D(0.4))
5 model.add(LSTM(176, dropout=0.2, recurrent_dropout=0.2))
6 model.add(Dense(2, activation='softmax'))
7 model.compile(loss = 'categorical_crossentropy', optimizer='adam', metrics = ['accuracy'])
8 print(model.summary())
9
```

WARNING:tensorflow:Layer lstm will not use cuDNN kernels since it doesn't meet the criteria. It will use the default LSTM implementation instead.

Model: "sequential"

| Layer (type)                         | Output Shape    | Param # |
|--------------------------------------|-----------------|---------|
| embedding (Embedding)                | (None, 31, 120) | 60000   |
| spatial_dropout1d (SpatialDropout1D) | (None, 31, 120) | 0       |
| lstm (LSTM)                          | (None, 176)     | 209088  |
| dense (Dense)                        | (None, 2)       | 354     |

=====  
Total params: 269,442  
Trainable params: 269,442  
Non-trainable params: 0  
=====  
None

Different activations available (<https://keras.io/api/layers/activations/> )

- relu function
- sigmoid function
- softmax function
- softplus function
- softsign function
- tanh function
- selu function
- elu function
- exponential function

We have used softmax activation.

Optimizer

In machine learning, Optimization is an important process which optimizes the input weights by comparing the prediction and the loss function. Keras provides quite a few optimizers as a module, and they are as follows:

- SGD – Stochastic gradient descent optimizer.
- RMSprop – RMSProp optimizer.

- Adagrad – Adagrad optimizer.
- Adadelat – Adadelat optimizer.
- Adam – Adam optimizer.
- Adamax – Adamax optimizer from Adam.
- Nadam – Nesterov Adam optimizer.

We have used Adam optimizer.

Loss-functions

(Reference: <https://neptune.ai/blog/keras-loss-functions>)

In deep learning, the loss is computed to get the gradients with respect to model weights and update those weights accordingly via backpropagation. Loss is calculated and the network is updated after every iteration until model updates don't bring any improvement in the desired evaluation metric. Some of which are

- Binary Classification
  - Binary Classification
  - Binary Cross Entropy
- Multiclass classification
  - Categorical Crossentropy
  - Sparse Categorical Crossentropy
  - The Poison Loss
  - Kullback-Leibler Divergence Loss

While compiling the model we used categorical\_crossentropy.

The embedding layer vectorizes the data and vectorized data is supplied to the activation function. The activation functions used for nonlinear it in the output activation function, decides whether the neurons are activated or not.

We have used softmax as the activation function. Post the activation function step the next layer is the dropout layer. This prevents overfitting problems in the model we've used dropout as 0.2. Dropout is one of the regularization strategy which reduces overfitting and thus increasing prediction error in deep neural networks.

Optimizer, loss and metrics and learning parameter is used to compile the model. We have considered adam. With increase in number of epochs, the chances of loss are decreased.

## 9. Chapter 5: RESULT and DISCUSSION

When we trained the model on first data set we received the output as :

```

- Fitting the data in model

[ ] 1 batch_size=32
    2 history = model.fit(X_train, y_train, epochs = 5, validation_split=0.2, batch_size=batch_size, verbose = 'auto')

Epoch 1/5
371/371 [=====] - 52s 129ms/step - loss: 0.5740 - accuracy: 0.7187 - val_loss: 0.5368 - val_accuracy: 0.7569
Epoch 2/5
371/371 [=====] - 50s 135ms/step - loss: 0.5219 - accuracy: 0.7642 - val_loss: 0.5307 - val_accuracy: 0.7525
Epoch 3/5
371/371 [=====] - 51s 137ms/step - loss: 0.5091 - accuracy: 0.7716 - val_loss: 0.5286 - val_accuracy: 0.7603
Epoch 4/5
371/371 [=====] - 59s 158ms/step - loss: 0.5018 - accuracy: 0.7744 - val_loss: 0.5415 - val_accuracy: 0.7613
Epoch 5/5
371/371 [=====] - 53s 144ms/step - loss: 0.4903 - accuracy: 0.7782 - val_loss: 0.5340 - val_accuracy: 0.7657

[ ] 1 # model.save('first_trained_model_v2.h5')
    2 model.save('FirstDataset.h5')

- Metrics

Plotting the metrics using the matplotlib.

[ ] 1 history_dict = history.history
    2 print(history_dict.keys())

dict_keys(['loss', 'accuracy', 'val_loss', 'val_accuracy'])

```

Figure 15 Fitting the model on dataset 1



Fitting the model on dataset we received the accuracy of 77 percent.

Accuracy plot:

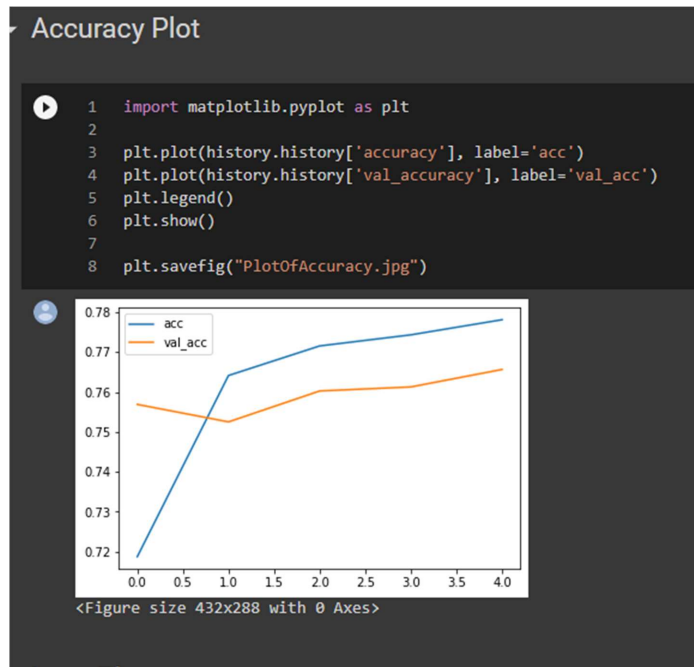


Figure 16 Accuracy plot for dataset 1

Loss Plot:



Figure 17 Loss plot for dataset 1

## Evaluation

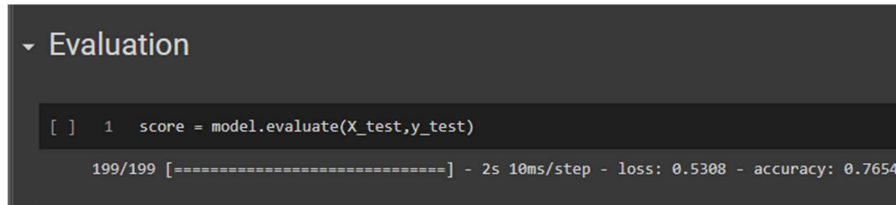


Figure 18 Evaluation of the model for dataset 1

We put a random statement to predict the results for the sample statement:

text = "Weapon fight kill gun missile tank fire death dead chaos"

As expected the result was : "Extremist" 😊

```
[ ] 1 text = "Weapon fight kill gun missile tank fire death dead chaos"
    2 tw = tokenizer.texts_to_sequences([text])
    3 tw = pad_sequences(tw,maxlen=31)
    4

[ ] 1 model.predict(tw)

1/1 [=====] - 0s 24ms/step
array([[0.2527691, 0.7472309]], dtype=float32)

[ ] 1 model.predict(tw).round()

1/1 [=====] - 0s 23ms/step
array([[0., 1.]], dtype=float32)

[ ] 1 model.predict(tw).round()[0].astype(int)

1/1 [=====] - 0s 79ms/step
array([0, 1])

▶ 1 prediction = model.predict(tw).round()[0].astype(int)[1]
  2 print(prediction)

1/1 [=====] - 0s 75ms/step
1

[ ] 1 sentiment_label = df['Tweet label'].factorize()
    2 sentiment_label

(array([0, 1, 1, ..., 1, 1, 0]),
 Index(['Non-extremist', 'Extremist'], dtype='object'))

[ ] 1 print("Predicted label: ", sentiment_label[1][prediction])

Predicted label: Extremist
```

Figure 19 Predicting the output of the model on trained dataset 1

The results of first dataset were not that good due to less training data.

#### APPLYING THE TRANSFER LEARNING ALGORITHM

We merged the dataset and then trained the model again to improve the prediction by mixing the datasets and the results were astonishing ! We can clearly see the accuracy jump in the new transfer learning algorithm :

```

Fitting the 2nd model in same same design model

[ ] 1 batch_size=32
    2 history2 = model2.fit(X_train2, y_train2, validation_split=0.2, epochs = 5, batch_size=batch_size, verbose = 'auto')

Epoch 1/5
/usr/local/lib/python3.8/dist-packages/tensorflow/python/data/ops/structured_function.py:264: UserWarning: Even though the `tf.config.experimental_run_functions_eagerly`
warnings.warn(
/usr/local/lib/python3.8/dist-packages/tensorflow/python/data/ops/structured_function.py:264: UserWarning: Even though the `tf.config.experimental_run_functions_eagerly`
warnings.warn(
972/972 [=====] - ETA: 0s - loss: 0.1899 - accuracy: 0.9305/usr/local/lib/python3.8/dist-packages/tensorflow/python/d
warnings.warn(
972/972 [=====] - 389s 401ms/step - loss: 0.1899 - accuracy: 0.9305 - val_loss: 0.1642 - val_accuracy: 0.9421
Epoch 2/5
972/972 [=====] - 334s 344ms/step - loss: 0.1527 - accuracy: 0.9460 - val_loss: 0.1511 - val_accuracy: 0.9443
Epoch 3/5
972/972 [=====] - 330s 340ms/step - loss: 0.1431 - accuracy: 0.9482 - val_loss: 0.1486 - val_accuracy: 0.9465
Epoch 4/5
972/972 [=====] - 331s 341ms/step - loss: 0.1358 - accuracy: 0.9500 - val_loss: 0.1483 - val_accuracy: 0.9461
Epoch 5/5
972/972 [=====] - 328s 338ms/step - loss: 0.1291 - accuracy: 0.9515 - val_loss: 0.1452 - val_accuracy: 0.9480

took 31 mins 21 sec on GPU
took 28 mins on GPU

[ ] 1 # model2.evaluate(X_test2,y_test2)

1/521 [=====] - ETA: 1:14 - loss: 0.0541 - accuracy: 0.9688/usr/local/lib/python3.7/dist-packages/tensorflow/python
"Even though the `tf.config.experimental_run_functions_eagerly` "
/usr/local/lib/python3.7/dist-packages/tensorflow/python/data/ops/structured_function.py:265: UserWarning: Even though the `tf.config.experimental_run_functions_eagerly` "
"Even though the `tf.config.experimental_run_functions_eagerly` "
521/521 [=====] - 74s 142ms/step - loss: 0.1462 - accuracy: 0.9473
[0.14620698988437653, 0.9473304748535156]

```

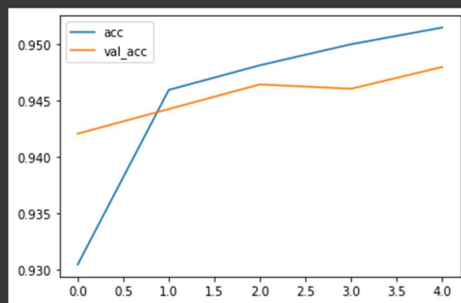
Figure 20 Mixed dataset results with accuracy of 94.73 percent also the Evaluation shown with loss of 14 percent only

In the mixed dataset we were able to train the dataset with more values which resulted in better accuracy for the model to predict various outputs . As compared to dataset one which the mixed dataset performed very well with very less loss and val loss percentage.

The Graphs were as follows for the below :

## Accuracy Plot

```
[ ] 1 import matplotlib.pyplot as plt
    2
    3 plt.plot(history2.history['accuracy'], label='acc')
    4 plt.plot(history2.history['val_accuracy'], label='val_acc')
    5 plt.legend()
    6 plt.show()
    7
    8 plt.savefig("PlotOfAccuracy2.jpg")
```



<Figure size 432x288 with 0 Axes>

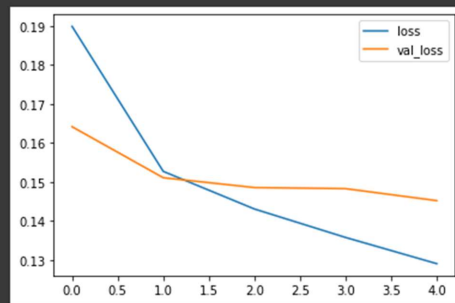
Figure 21 Accuracy plot for dataset 2

Much more better plot as compared to dataset 1

Loss Plot for dataset 2 :

## ▼ Loss Plot

```
[ ] 1 plt.plot(history2.history['loss'], label='loss')
    2 plt.plot(history2.history['val_loss'], label='val_loss')
    3
    4 plt.legend()
    5 plt.show()
    6
    7 plt.savefig("PlotOfLoss2.jpg")
```



<Figure size 432x288 with 0 Axes>

Figure 22 Loss plot for dataset 2, the mixed dataset

Predictions for dataset 2 :

```

1 test_sentence4 = "I hate park when the slide in it are empty."
2 raw = predict_sentiment2(test_sentence4)
3 print(raw)

1/1 [=====] - 0s 149ms/step
Predicted label: Non-extremist
[[0.92635655 0.07364339]]

[ ] 1 test_sentence5 = "I hate park when the slide in it are bombed."
2 raw = predict_sentiment2(test_sentence5)
3 print(raw)

1/1 [=====] - 0s 142ms/step
Predicted label: Non-extremist
[[0.92635655 0.07364339]]

[ ] 1 test_sentence6 = "Many people were killed in the water park."
2 raw = predict_sentiment2(test_sentence6)
3 print(raw)

1/1 [=====] - 0s 140ms/step
Predicted label: Extremist
[[0.01310811 0.9868919 ]]
/usr/local/lib/python3.8/dist-packages/tensorflow/python/data/ops/structured_f
warnings.warn(

```

Figure 23 Predictions done on the dataset 2 the mixed dataset

Moreover, right now the project only focuses on one language therefore we need to find solutions to make it multilingual.

## Future Scope of the project

Some of the future scope points out of endless possibilities for this project would be

- Project in future will be enhanced for analyzing image and video dataset . This will help to alert the authorities in case of any extremist behaviour in advance.
- Project right now is trained on English words , this can be made multi-lingual.
- Browser plug-in adaptation will make the project more dynamic , that is while a portion of text can be selected and with image processing the text can be analysed if the content is extremist or not .

## 10. Chapter 6: CONCLUSION

Social media provides the required freedom for all of us to share our opinions which inspires certain individuals to spread extremist content on the internet. They use this platform to inspire individuals to become a supporter of their cause. Therefore, the task of researchers, psychologists, data mining experts and computer science engineers becomes very crucial for classification of extremist content various such platforms. In our case we aim to curb the same on one of the most popular social media platform Twitter by training our model with utmost accuracy.



## 11. REFERENCES:

1. Chalothorn, T.; Ellman, J. Using SentiWordNet and Sentiment Analysis for Detecting Radical Content on Web Forums. 2012. Available online: [http://nrl.northumbria.ac.uk/13075/1/1\\_\\_\\_\\_Chalothorn\\_Ellman\\_SKIMA\\_2012.pdf](http://nrl.northumbria.ac.uk/13075/1/1____Chalothorn_Ellman_SKIMA_2012.pdf) .
2. Mei, J.; Frank, R. Sentiment Crawling: Extremist Content Collection through a Sentiment Analysis Guided Web-Crawler. In Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 15), Paris, France, 25–28 August 2015; pp. 1024–1027
3. P. Choudhary and U. Singh, “A Survey on Social Network Analysis for CounterTerrorism,” *Int. J. Comput. Appl.*, vol. 112, no. 9, pp. 24–29, 2015.
4. P. Burnap et al., “Tweeting the terror: modelling the social media reaction to the Woolwich terrorist attack,” *Soc. Netw. Anal. Min.*, vol. 4, no. 1, pp. 1–14, 2014.
5. Azizan, S.A.; Aziz, I.A. Terrorism Detection Based on Sentiment Analysis Using Machine Learning. *J. Eng. Appl. Sci.* 2017, 12, 691–698.
6. Berger, J.M.; Morgan, J. The ISIS Twitter Census: Defining and Describing the Population of ISIS Supporters on Twitter. *Brook. Proj. Us Relat. Islamic World* 2015, 30, 20.
7. Mei, J.; Frank, R. Sentiment Crawling: Extremist Content Collection through a Sentiment Analysis Guided Web-Crawler. In Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 15), Paris, France, 25–28 August 2015; pp. 1024–1027.
8. [8] Wadhwa, P.; Bhatia, M.P.S. Classification of Radical Messages on Twitter using Security Associations. In *Case Studies in Secure Computing Achievements and Trends*; Auerbach Publications: Boca Raton, FL, USA, 2014; pp. 273–294.
9. Sharif, W., Mumtaz, S., Shafiq, Z., Riaz, O., Ali, T., Husnain, M., & Choi, G. S. (2019). An empirical approach for extreme behavior identification through tweets using machine learning. *Applied Sciences* (Switzerland), 9(18). <https://doi.org/10.3390/app9183723>

10. Azizan, S.A.; Aziz, I.A. Terrorism Detection Based on Sentiment Analysis Using Machine Learning. *J. Eng. Appl. Sci.* 2017, 12, 691–698.
11. Yadron, D. Twitter Deletes 125,000 ISIS Accounts and expands anti-Terror Teams. Available online: <https://www.theguardian.com/technology/2016/feb/05/twitter-deletes-isis-accounts-terrorism-online> (accessed on 15 May 2019)
12. [11.A] Santos, C.D.; Gatti, M. Deep convolutional neural networks for sentiment analysis of short texts. In *Proceedings of the 25th International Conference on Computational Linguistics: Technical Papers*, Dublin, Ireland, 23–29 August 2014; pp. 69–78.
13. [12]ItsSuru. (2021, February 20). Global terrorism. Kaggle. Retrieved January 4, 2023, from <https://www.kaggle.com/datasets/itsuru/global-terrorism?select=globalterrorism.csv>
- 13.[13] Waqas Sharif 1 , Shahzad Mumtaz 1 , Zubair Shafiq 2 , Omer Riaz 1 , Tenvir Ali 1 , Mujtaba Husnain 1 and Gyu Sang Choi 3, “An Empirical Approach for Extreme Behavior Identification through Tweets Using Machine Learning”
14. Wei Y, Singh L, Marti S (2016) Identification of extremism on Twitter. *Proceedings of the IEEE/ACM international conference on advances in social networks analysis and mining*. IEEE, New Jersey, pp 1251–125
15. Zhang H, Wang J, Zhang J, Zhang X (2017) Ynu-hpcc at semeval 2017 task 4: using a multi-channel cnn-lstm model for sentiment classification. In: *Proceedings of the 11th international workshop on semantic evaluation (SemEval-2017)*
16. Yenter A, Verma A (2017) Deep CNN-LSTM with combined kernels from multiple branches for IMDB review sentiment analysis. In: *2017 IEEE 8th annual ubiquitous computing, electronics and mobile communication conference (UEMCON)*. IEEE. pp 540–546
17. Matthias HartungRoman KlingerFranziska SchmidtkeLars VogelFlavius FrascinarAshwin IttooLe Minh NguyenElisabeth Métais(1 January 2017) Springer International Publishing Identifying Right-Wing Extremism in German Twitter Profiles: a Classification Approach

18. Cortes, C., Vapnik, V.: Support-vector networks. Machine Learning 20, 273–297 (1995)
19. Patil, Sojwal, Aishwarya Gune and Mayura Nene, “Convolutional neural networks for text categorization with latent semantic analysis,” 2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS), 10.1109/ICECDS.2017.8390217, IEEE
20. Sign in to your account. Retrieved January 4, 2023, from [https://canvas.swansea.ac.uk/courses/23936/files/2779243?module\\_item\\_id=1474308](https://canvas.swansea.ac.uk/courses/23936/files/2779243?module_item_id=1474308)
21. Azizan, S.A.; Aziz, I.A. Terrorism Detection Based on Sentiment Analysis Using Machine Learning. J. Eng. Appl. Sci. 2017, 12, 691–698.
22. O'Hara, K. and Stevens, D. (2015), Echo Chambers and Online Radicalism: Assessing the Internet's Complicity in Violent Extremism. Policy & Internet, 7: 401-422. <https://doi.org/10.1002/poi3.88>
23. von Behr, I., Reding, A., Edwards, C., & Gribbon, L. (n.d.). Radicalisation in the digital era: The use of the internet in 15 cases of terrorism and extremism. [www.rand.org](http://www.rand.org)
24. Hao F, Park DS, Pei Z (2018) When social computing meets soft opportunities and insights. Human-centric Comput Inform Sci 8(8):1–18
25. Hao F, Min G, Pei Z, Park DS, Yang LT (2017) k-clique communities detection in social networks based on formal concept analysis. IEEE Syst J. 11(1):250–259
26. Scrivens, R.; Davies, G.; Frank, R. Searching for Signs of Extremism on the Web: An Introduction to Sentiment-Based Identification of Radical Authors. Behav. Sci. Terror. Political Aggress. 2018, 10, 39–59
27. Seib, P.; Janbek, D.M. Global Terrorism and New Media: The Post-Al Qaeda Generation, 1st ed.; Routledge: London, UK, 2010
28. Iskandar B (2017) Terrorism detection based on sentiment analysis using machine learning. J Eng Appl Sci 12–3:691–698

29. Ferrara E, Wang WQ, Varol O, Flammini A, Galstyan A (2016) Predicting online extremism, content adopters, and interaction reciprocity. International conference on social informatics. Springer, New York, pp 22–39
30. Bundeskriminalamt. (2021). Projektbeschreibung. Bundeskriminalamt Accessible at [https://www.bka.de/DE/UnsereAufgaben/Forschung/ForschungsprojekteUndErgebnisse/TerrorismusExtremismus/Forschungsprojekte/MOTRA/Projektbeschreibung/projektbeschreibung\\_node.html](https://www.bka.de/DE/UnsereAufgaben/Forschung/ForschungsprojekteUndErgebnisse/TerrorismusExtremismus/Forschungsprojekte/MOTRA/Projektbeschreibung/projektbeschreibung_node.html)
31. Kilimci, Zeynep H. and Selim Akyokus, “Deep Learning- and Word EmbeddingBased Heterogeneous Classifier Ensembles for Text Classification,” Volume 2018, Article ID 7130146, 10 pages <https://doi.org/10.1155/2018/7130146>.
32. Johnston, Andrew H. and Gary M. Weiss, “Identifying Sunni Extremist Propaganda with Deep Learning ,” 978-1-5386-2726-6/17/\$31.00 c 2017 IEEE.
33. Bhargava, Rupal, Yashvardhan Sharma and Shubham Sharma, “Sentiment Analysis for Mixed Script Indic Sentences,” 2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI), 978-1-5090- 2029-4/16/\$31.00 c 2016 IEEE
34. Bermingham, A.; Conway, M.; Mcinerney, L.; Hare, N.O.; Smeaton, A.F. Combining Social Network Analysis and Sentiment Analysis to Explore the Potential for Online Radicalisation. In Proceedings of the 2009 International Conference on Advances in Social Network Analysis and Mining (ASONAM 2009), Athens, Greece, 20–22 July 2009; pp. 231–236.
35. Wadhwa, P.; Bhatia, M.P.S. Classification of Radical Messages on Twitter using Security Associations. In Case Studies in Secure Computing Achievements and Trends; Auerbach Publications: Boca Raton, FL, USA, 2014; pp. 273–294.
36. IEEE Computational Intelligence Society, & Institute of Electrical and Electronics Engineers. 2017 SSCI proceedings.

37. Mussiraliyeva, S., Bolatbek, M., Omarov, B., Medetbek, Z., Baispay, G., & Ospanov, R. (2020, November 28). On detecting online radicalization and extremism using natural language processing. Proceedings - 2020 21st International Arab Conference on Information Technology, ACIT 2020. <https://doi.org/10.1109/ACIT50332.2020.9300086>
38. Zheng, X., IEEE ITSS, Zhongguo ke xue yuan. Institute of Automation, & Institute of Electrical and Electronics Engineers. (n.d.). 2019 IEEE International Conference on Intelligence and Security Informatics (ISI) : July 1-3, 2019, Shenzhen, China.
39. Owoeye, K. O., & Weir, G. R. S. Classification of Radical Web Text using a Composite-Based Method.
40. Owoeye, K. O., & Weir, G. R. S. Classification of Extremist Text on the Web using Sentiment Analysis Approach.
41. (<https://www.nltk.org/book/ch05.html>)
42. <https://www.analyticsvidhya.com/blog/2018/02/natural-language-processing-for-beginners-using-textblob/> ).
43. ( M. K. Dahouda and I. Joe, "A Deep-Learned Embedding Technique for Categorical Features Encoding," in *IEEE Access*, vol. 9, pp. 114381-114391, 2021, doi: 10.1109/ACCESS.2021.3104357. )
- 44.
- 45.
- 46.
- 47.