Extremism Analysis for Twitter using Multi–Class Classification

Mazin Sherif
University of Wollongong in Dubai, Dubai, UAE
Email: ms679@uowmail.edu.au

Omar Taher
University of Wollongong in Dubai, Dubai, UAE
Email: ot756@uowmail.edu.au

Taha Afsar
University of Wollongong in Dubai, Dubai, UAE
Email: ta171@uowmail.edu.au

Zain Syed
University of Wollongong in Dubai, Dubai, UAE
Email: zs089@uowmail.edu.au

A R T I C L E I N F O

Keywords:
Natural Language Processing, Machine Learning, Twitter, Extremism, Radicalization, Feature Extraction, Word2Vec, Random Oversampling, Multiclass Classification, Random Forest, XGBoost, Hyperparameter Optimization, Bayesian Optimization, Optuna, Evaluation Metrics

A B S T R A C T

With the immense escalation of the popularity of social media platforms in today's digital world, individuals all around the globe are attracted to these spaces for the purposes of seeking information or expressing their opinions on trending topics, current issues, or ongoing debates. Due to the unrestricted right of speech people possess on such online platforms, one example being Twitter, there is an increasing amount of contentious content being circulated. This has eventually gravitated toward large-scale intersocietal and intra societal conflicts as a result of extremist views or collective radicalization.

The automatic flagging of extremism and radicalization requires refined and sophisticated Natural Language Processing (NLP) methods. The present paper comprehends a study on both theoretical and practical material focusing on the main challenges of extremism identification and methods as to how various subjects of extremist tweets can be classified to resolve the problem in question. Random Forest and XGBoost machine learning classifiers were used to classify a tweet into its particular category using multiclass classification. Additionally, the Optuna framework was employed for hyperparameter optimization. The accuracies retrieved out of the entire process were discovered to be 94.64% and 94.87% for Random Forest and XGBoost respectively.

1. Introduction

Social media refers to methods of communication where individuals produce, publish, and exchange knowledge and concepts in online groups. There are millions of platforms that are used on a daily basis by billions of individuals in the current world. From the many platforms available today, 'Twitter' is one of the few most engaging and accustomed programs accessible today. Twitter is the most well-known social media platform with more than 500 million members [1]. This service allows acquaintances to interact and keep in touch by sending brief and frequent messages to one another. On this microblogging website, users also have the opportunity to submit tweets, or status updates with no more than 280 characters [1]. Tweets are instantly available to the user's community of followers as well as anyone who does not have a Twitter account; as a result, the great majority of material posted on Twitter is publicly accessible.

Twitter presents its users with a great room of freedom, where they are free to choose any alias they desire and protect their identities. This is leverage to extremists that can openly display their views to any extent they desire. Several extremist groups use Twitter to publish comments and false news releases, spread propaganda, and urge or justify assaults. Groups utilize Twitter to spread their ideas and misinformation to a wide audience, whether their narrative implies that the West is islamophobic, the government is overstepping its limits, or that particular religions or ethnicities are inferior. This method of marketing is to openly disparage opposing viewpoints and data from other Twitter accounts. There are several barriers to understanding extremism and collective radicalization; one example is the differentiation between who is actually engaged in such activities and who is just casually speaking about it. Thus, there is a need for accurate systems to identify true extremists and filter them out.

With an emphasis on categorizing tweets into extremist and non-extremist classifications, this work intends to propose a framework for content analysis connected to radicalization. Existing studies for tweet classification utilize traditional techniques which result in non-rigorous analysis. However, the focus of this study will provide an in-depth exploration for the categorization of an extremist tweet. This report examines the development of a tweet classification model based on machine learning techniques to primarily identify whether a tweet is extremist or not and then classify each extremist tweet into the following 12 categories: 'Covid 19', 'Anti-Asian Racism', 'Xenophobia', 'Anti-Black Racism', 'Islamophobia', 'White Identity Politics', 'Pro-Trump', 'Pro-Wall', 'Elections', 'Antifa', 'Calls for Action', 'Non-US', 'Uncategorized'.

This report consists of a total of 8 sections that discuss the problem and the implementation of the solution. Section 2 highlights the problem statement with a research question to identify what the report aims to resolve. Section 3 details a literature review consisting of summarized research studies that have conducted similar experiments in the past using various machine learning techniques. The methodology proposed by the authors in this paper is elaborated in detail across Section 4. The evaluation of the models is detailed using several evaluation metrics in Section 5. Section 6 discusses the results obtained from both models. Additionally, a comparative analysis was performed in Section 7. In Section 8, the study comes to an end with a final conclusion and recommendations for future improvement of the proposed model.

1.1 Random Forest Classifier

Before understanding the framework of how a Random Forest model works, it is important to recognize the structure of a decision tree. Sequential questions have answers that lead you along a certain path in a specific decision tree. The model behaves under an 'if this, then that' circumstance, eventually producing a certain outcome [2]. This is clearly illustrated in Fig. 1 below, where the arrow follows a continuous decision to be implemented.

A significant number of decision trees are built during the training phase of the random forests ensemble learning approach, which is used for classification, regression, and other additional tasks. The mean or average forecast of each individual tree is returned for regression tasks. The tendency of decision trees to overfit their training set is corrected by implementing random forest models. Although they frequently outperform decision trees, gradient-enhanced trees are more accurate than random forest, however, their effectiveness may be impacted by data peculiarities [3].

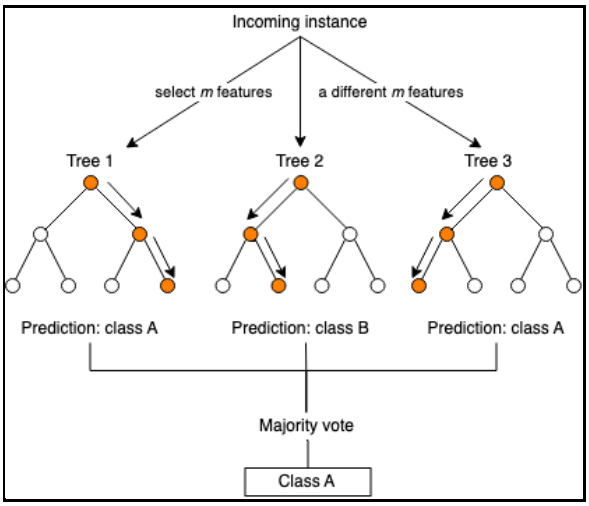

Fig. 1: Structure of Decision Tree [3]

- *A large number of relatively uncorrelated models (trees) operating as a committee will outperform any of the individual constituent models [4].*

1.2 XGBoost Classifier

Boosting is an ensemble approach, which implies a method to combine predictions from several models into a single forecast. It does this by modeling each prediction sequentially depending on the inaccuracy of its predecessor, allowing an increase in weight to the predictors that perform better. A particular kind of boosting called gradient boosting, minimizes the loss of function by employing a gradient descent technique [5].

XGBoost is a gradient boosting method that utilizes decision trees as its weak predictors. Additionally, its implementation was exclusively adapted for the best performance and efficiency. The following example demonstrates the procedure of how XGBoost works [6]:

Eq. 1 represents a database that has ***m*** features and a ***n*** number of examples,

$$DS = \{(x_i, y_i): i = 1 \dots n, x_i \in \mathbb{R}^m, y_i \in \mathbb{R}\}$$

(Eq. 1)

y_i is the predicted output of the tree model in Eq. 1.

$$Å \cdot_i = \emptyset(x_i) = \sum_{k=1}^{K} f_k(x_i), f_k \in \mathcal{F}$$

(Eq. 2)

K stands for the number of trees in the model and f_k is the K-th tree. In order to resolve Eq. 2, it is important to determine the optimum set of functions by reducing the loss and regularization objectives.

$$\mathcal{L}(\emptyset) = \sum_i l(y_i, Å._i) + \sum_k \Omega(f_k)$$

(Eq. 3)

The difference between the projected output l and the actual output y_i is represented by the loss function, or 'l'. This helps prevent overfitting of the model even if Ω is a measure of the model's complexity which is determined by using (Eq. 4):

$$\Omega(f_k) = \gamma T + \frac{1}{2} \lambda || w ||^2$$

(Eq. 4)

In Eq. 5, T represents the number of leaves on the tree, and w for each leaf's weight. This technique works by continuously adding

additional functions while the model is trained. Consequently, in the next iteration, the following new tree is added as shown in Eq. 6 .

$$\mathcal{L}^{(t)} = \sum_{k=1}^{K} l(y_i, \text{Å}_i^{(t-1)} + f_k(x_i)) + \Omega(f_t)$$

(Eq. 5)

$$\mathcal{L}_{split} = \frac{1}{2}\left[\frac{(\sum_{i\in IL} g_i)^2}{\sum_{i\in IL} h_i + \lambda} + \frac{(\sum_{i\in IR} g_i)^2}{\sum_{i\in IR} h_i + \lambda} + \frac{(\sum_{i\in I} g_i)^2}{\sum_{i\in I} h_i + \lambda}\right] - \gamma$$

$$g_i = \partial_{\text{Å}^{(t-1)}} l(y_i, \text{Å}^{(t-1)})$$

$$h_i = \partial^2_{\text{Å}^{(t-1)}} l(y_i, \text{Å}^{(t-1)})$$

(Eq. 6)

1.3 Text Vectorization

The significance of data has proliferated in every aspect of our lives. Amongst the different representations of data, the textual form is ceaselessly the most widespread, especially considering its usage in communication across diverse platforms. It is imperative to utilize the ability of machines to generate imperceptible insights by transmuting the data in a form that is conveniently transparent to the computers. One of the preliminary methods involved in converting text documents in a way that is machine-readable is vectorization [7].

Text-Vectorization is almost a prerequisite to any Natural Language Processing (NLP) task that involves solving text-based problems mathematically. These vectorizers can range from very lucid to really complex models [7]. An unsophisticated and basic approach to building such vectors would be to generate a numerical representation of all words in the lexicon using binary encoding. This method is called One-hot Encoding and should be avoided as it would lead to dimensional disaster and data sparsity. Some of the more common methods to represent a vector make use of word frequency for each term in a document. Although most of these approaches still fail to solve problems such as dimensionality reduction and feature extraction of text semantics, they are commonly used to solve most classification problems [8]. A brief explanation of such methods is as follows:

Bag Of Words (BOW):

In general classification of documents, a Bag of Words (or commonly known as BoW) is a vector representation of term frequencies. It retrieves word occurrences for each sentence or document while considering word duplicates but disregards grammar and its sequence. This form of numerical representation is usually enforced in classification methods where the features used are based on the frequency of each term [9]. Consider an example of a paragraph containing two sentences that are as follows:

"The moon is bright. It seems to be really bright."

The Bag of Words representation for this corpus is shown in Table 1.

1.4 Hyperparameter Tuning

Hyperparameters are a deliberate group of choices that precisely influence the training and outcome of machine learning models resulting in a substantial increase in the model's performance. Since the choices are dataset specific and there are no conventional set of parameters that would yield maximum performance, it is essential for an ML model to have the right hyperparameter setting. Most parameters have an extensive scale of values that can be specified. These can be selected based on default values proposed by the developer of the concerned ML algorithm or by finding values suggested by research authors working on a project within a similar domain. For either case, the model may not perform well considering the differences in each set of data. Another approach can be carried out by performing manual search. This requires prior knowledge and experience with the algorithm. Generally, these preconditions are strenuous to accomplish. Therefore, a systematic gradual tuning scheme that requires no expertise or profound knowledge would be a far more favorable alternative [10]. Some of the widely used hyper-parameter optimization techniques

Documents	be	bright	is	it	moon	really	seems	the	to
The moon is bright	0	1	1	0	1	0	0	1	0
It seems to be really bright	1	1	0	1	0	1	1	0	1

Table 1: Bag of Words representation

include Grid Search, Random Search, Bayesian Hyper-parameter Optimization with HyperOpt [11].

2. Research Question

"How can machine learning algorithms be used to detect and mitigate extremist content on Twitter?"

The proliferation of extremist content on social media platforms, such as Twitter, has raised concerns about the potential impact on public safety, online communities, and the spread of harmful ideologies. Current approaches to detecting and removing extremist content on social media often rely on manual review, which is time-consuming and may not be effective in identifying all instances of extremist content. As a result, there is a need to develop more efficient and effective methods for detecting and removing extremist content on Twitter. It is one of our greatest aids to counter the spread of such harmful ideas online. This can not only improve the overall safety and user experience on Twitter but can also help in preventing the radicalization of some individuals and the escalation of real-world violence.

3. Literature Review

It has become increasingly common for extremists and individuals who support politicalradicalism to propagate their beliefs and principles by posting extremist radicalism-promoting tweets. It is practically impossible for Twitter moderators or an intelligence and security analyst to manually identify such tweets as there are millions of tweets posted everyday [12]. Due to this reason, there is a crucial need for extremism identification systems to be developed. There are several techniques already employed using both machine learning and deep learning models for these purposes. Some of these techniques include NER (Named-Entity Recognition) Classifiers, Rule-Based Matching, Multilabel, Multiclass, and Binomial Text Classification.

Mariam Nouh et al. [13] proposed a model in 2019 to detect and classify radical messages extracted from known pro-ISIS Twitter accounts from 2015. Their novel approach included detecting several various signals in the form of textual, psychological, and behavioral properties, taking into regard the identification of psychological signs that are not consciously transmitted when communicating. All these things combine to provide an all-rounded approach to accurately extract radical properties from tweets. There were three different datasets combined into one which was then preprocessed through multiple steps, some of which include reducing noise, removing stop words, URLs, and emojis, performing tokenization to prepare the text for lemmatization, etc. Two different feature selection techniques were used in the study: (i) Term-Frequency Inverse Document-Frequency (TF-IDF) using uni-, bi-, and trigrams, and (ii) Word embedding using a word2vec model [3]. The classification technique used in the study is binomial or binary classification, which basically classifies the tweets into either one of two categories: radical or normal. The dataset used in the study was imbalanced, therefore, the appropriate oversampling and undersampling techniques were applied to balance it. Random Forest, Neural Network, Support Vector Machine, and K-Nearest Neighbor were the classification models used in the study, out of which it was

concluded that the Random Forest and Neural Network classifiers performed the best. L. Kaati et al. [14] wrote a paper in 2015 that aims to detect twitter users that are involved in "Media Mujahideen" - the supporters of jihadist groups who disseminate propaganda content online. Two feature sets were used for the purpose of this study - data-dependent and data-independent. The data-independent feature set was incorporated so that the proposed method could be used as a baseline to classify extremism from various kinds of sources. The AdaBoost classifier, also known as Adaptive Boosting, is an ensemble boosting technique used to perform classification accurately which was used in the study. The experiment was conducted on tweets in English and Arabic. The model was found to perform with a high accuracy score of 99.5% for the English tweets.

S. Agarwal et al. [12] approached the problem of extremist tweet detection by performing one-class or unary-class categorization by creating a statistical model from a training set containing objects from a single class. A number of linguistic features such as negative emotions and offensive terms were utilized to discriminate hate and extremism-promoting tweets from normal, deradicalized tweets. The authors of the study employed a single-class SVM and KNN algorithm for the one-class classification technique. The results included F-scores of 0.60 and 0.83 for the KNN and SVM classifiers respectively.

4. Methodology

This section of the paper meticulously demonstrates the implementation of an approach to classify tweets to its respective category of radicalization, utilizing python and its relevant libraries for NLP, Machine Learning, Data Sampling and Hyperparameter Tuning. In Fig. 2, a flowchart demonstrating the same is illustrated.

4.1 Exploratory Data Analysis (EDA)

Exploratory Data Analysis (EDA) is the process of examining and summarizing a dataset in order to understand its characteristics and underlying patterns. It typically involves the visualization of data, identifying trends, and performing statistical tests which helps in gaining proper insights from the data. Covert patterns and relationships in the data can easily be uncovered with the help of this process. EDA is often the first step in the data analysis process that can lead the way for developing more models and algorithms.

From Fig. 3, generated using the pandas and matplotlib libraries from python, it is depicted that categories 8 and 99 contain the most repeated terms 'china' and 'buildthewall', respectively. Fig. 4, generated using pyplot, also displays significance to both these categories due to them containing the highest number of tweets with category 8 having a total number of 6,251 tweets followed by category 99 having 2,996 tweets. Using this beneficial information, we can identify potential problems with this data. For instance, the observed words 'china' and 'buildthewall' have the highest count in two separate categories; it indicates that these topics are being discussed very frequently which depicts a common topic or on-going theme.

4.2 Lemmatization with POS Tags

Lemmatization is a technique used to combine the analogous words in a way that can be identified as a single element (commonly referred to as the word's "lemma") [15]. Table 2 shows an example of a few words and their lemmas. Using spaCy's tokenizer, each word was tokenized along with its POS tag. After which, they are compared to a user-defined list of tags, consisting of nouns, adjectives and verbs, to be lemmatized to. Due to computational constraints, the number of tags in this study were restricted to only three.

4.3 Word2vec

The Word2vec model is a word vector generation method that makes use of word embeddings and consists of skip-gram and Continuous Bag of Words (CBOW). It is designed to contain a

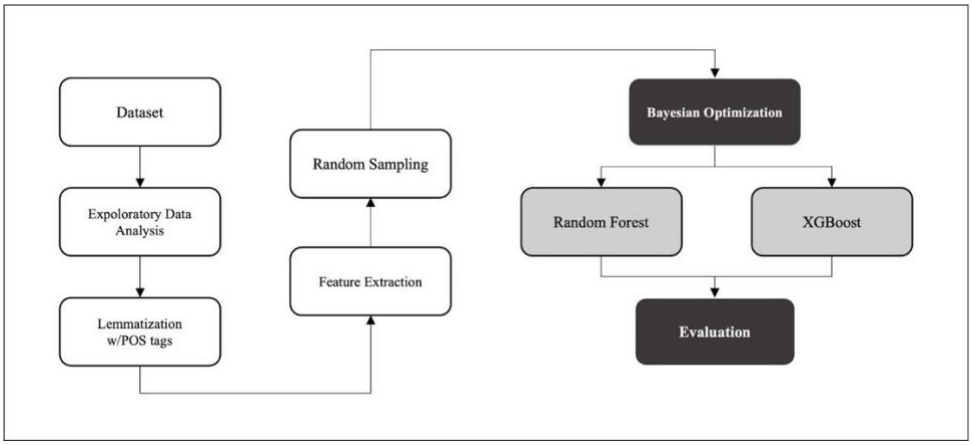

Fig. 2: Flowchart demonstrating the approach

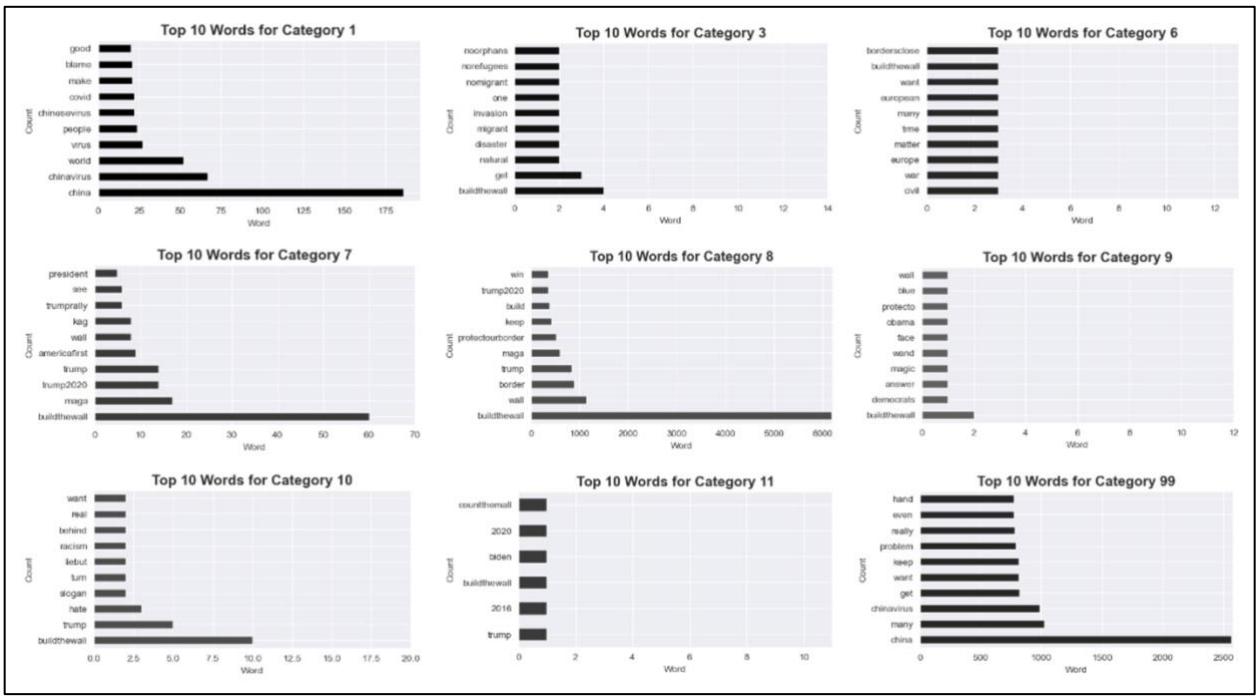

Fig. 3: Top 10 words for Categories 1, 3, 6, 7, 8, 9, 10, 11 and 99

Words	Lemma
Corpora	corpus
Playing	play
Played	play
Plays	play
Different	differ

Table 2: Example of words and their vocabulary form (lemma)

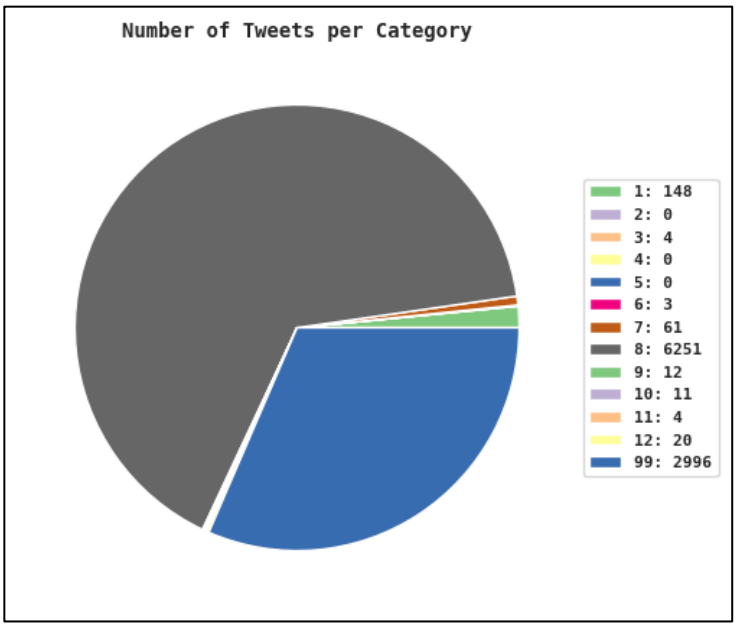

Fig. 4: Distribution of tweets across each category

disseminated emotion of words that map them to vectors with describable dimensions. Since the context is contemplated while training, a less dense dimensional vector with text semantics is produced. Each term in a document is intrinsically established as a word sequence [8] (shown in Eq. 7).

$$d = \left[d^{(1)}, d^{(2)}, \ldots, d^{(j)}, \ldots, d^{(i)}\ \right]$$

(Eq. 7) (Yang et al. 2022)

Where,

d(j) indicates word at jth position

d(i) indicates word at last position

To acquire the word vector, a mapping association should be established between the observed word and its context. This is trained using a neural network where the hidden layer weight procured during training is the word vector itself [8].

Word2vec will be used as the main feature extraction technique for the development of the model used in this study. A 3-dimensional graph which exhibits the top eight similar words for the term 'wall' is illustrated in Fig. 5.

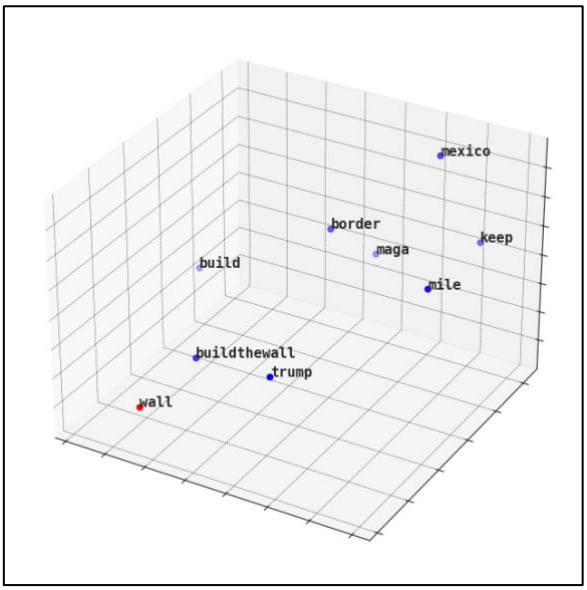

Fig. 5: Top 8 similar terms for the word 'wall'

4.4 Random Oversampler

Data imbalances are common and inherent in the real world. The data often shows a distort distribution with long tails. However, the majority of machine learning algorithms in use

today are designed on the assumption of even distribution across each targeted class. One of the many methods of overcoming this problem would be random oversampling. Random oversampling involves choosing random instances from the minority class with substitution and appending multiple versions of this instance to the training data. This results in a more likely opportunity of a single instance being selected more than once [16].

The Imblearn library is developed primarily to handle imbalanced datasets. It offers a variety of techniques other than oversampling, such as undersampling and SMOTE, to manage and eliminate the datasets imbalance. This library includes a number of ensemble techniques, including boosting classifiers, bagging classifiers and random forest classifiers, which may be used to train models for unbalanced data sets in an accurate and efficient manner [17].

Overall, the random oversampling method endures coherent advantages to researchers for unbiased treatment. A balanced subset is created to carry out an equal potential for representing the complete set as a whole. Besides the multiple precedence this method prevails, it is important to understand the potential drawbacks at hold. When training a model and increasing more instances from the minority class, it is possible for the computing costs to increase, resulting in the model being exposed to the same cases repeatedly. This will affect the accuracy by a slight margin for the balanced output. Nonetheless, the benefits outweigh the drawbacks proving to researchers that the random oversampling method is the most efficient and accurate at use.

4.5 Bayesian Optimization

Function optimization is one of the most primitive aspects of machine learning. Most machine learning algorithms involve the optimization of parameters like weights or coefficients in response to training data. In other words, optimization refers to the process of finding the most optimal set of hyperparameters that configure the training of a machine learning algorithm [18]. Black-box optimization (BBO) refers to the optimization of algorithms where the structure of the objective function and/or the constraints defining the set is unknown, unexploitable, or non-existent [19]. Global optimization is the process of searching for an input that results in the minimum or maximum cost of a given objective function [18]. What makes the problem of global optimization so difficult as compared to local optimization is the fact that there is a presence of local optima, which is extremely easy to locate but the goal of locating a global optima means searching the entire space and retrieving the optimal parameters.

There are a wide variety of algorithms available for black-box global optimization, one such candidate being Bayesian optimization. It is an approach that uses Bayes Theorem to obtain the minimum or maximum of an objective function, usually when the function is complex, non-linear, noisy, and computationally expensive to evaluate. It works by iteratively building a probabilistic model of the objective function, called the surrogate function, by mapping from hyperparameter values to the actual objective function. This probabilistic model captures the behavior of the surrogate function to form a posterior distribution over the objective function, which is then used to form an acquisition function that determines the next point possessing the best improvement probability [18][20]. The operation of the Bayes optimization algorithm results in a trade-off between exploration and exploitation, which is then optimally balanced by algorithms such as Tree-structured Parzen Estimator (TPE) and Gaussian Process Regressor [20].

Optuna is an advanced, open-source hyperparameter optimization framework that implements the Bayesian optimization algorithm by default, specifically, the Tree-Structured Parzen Estimator (TPE). TPE, in short, is an iterative algorithm that utilizes the history of previously evaluated hyperparameters in order to create a probabilistic model which would be able to advocate or recommend the next set of hyperparameters to assess. This helps to speed up optimization time and performance drastically

compared to traditional methods such as GridSearch. Furthermore, it also allows users to plot optimization histories or visualize relationships between the hyperparameters for better interpretability of the model [20]. The implementation of the optimization process using Optuna is elaborated in detail below:

Initially, an objective function needs to be defined by the user. This function contains the fundamental logic of creating a regular model definition, training, and testing processes. After the evaluation of the model, it returns the evaluation metrics mentioned by the user.

The second step is to create a Trial class, which is used to store a specific combination of hyperparameters that will be used by the machine learning model later on.

Last but not least, a Study object is created and called. This call is where the optimization of the objective function takes place in order to find the leading combination of hyperparameters. The user can define the maximum trial or time until when the study object iterates the trials. Finally, the trial with the best-concluded hyperparameters will be stored in study.best_trial.

The hyperparameters used for the classification models employed in this paper are listed below:

Hyperparameters used for Random Forest Classifier [21]:

1. *max_depth* - The longest path between the root and leaf node.
2. *min_sample_split* - The minimum required number of observations in any given node to split by.
3. *n_estimators* - Number of trees in the forest.
4. *criterion* - The function to measure the quality of a split, ‘gini’ for the Gini impurity and ‘entropy’ for information gain.
5. *min_samples_leaf* - Specifies the minimum number of samples that should be present in the leaf node after splitting a node.
6. *max_features* - Resembles the maximum number of features provided to each tree in a random forest.

Hyperparameters used for XGBoost Classifier [22]:

1. *subsample* - Represents the subsample ratio of the training sample.
2. *max_depth* - Represents the limit of how deep each tree can grow; default = 6.
3. *reg_alpha (alias: alpha)* - The L1 regularization parameter, the higher the value, the more conservative the model; default = 1.
4. *reg_lambda (alias: lambda)* - The L2 regularization parameter; works the same as *alpha*. Default value is set to zero.
5. *booster* - A special purpose hyperparameter that consists of 3 options: ‘dart’, ‘gbtree’ (tree-based) and ‘gblinear’ (Ridge regression).

5. Evaluation

In this section, the evaluation metrics used for analyzing the best classification model are demonstrated. The predictions are evaluated in terms of accuracy, precision, recall, f1 score, log loss, Cohen’s Kappa score and Matthew’s correlation coefficient. Before understanding how each of these metrics work, some fundamental concepts regarding outcomes need to be elaborated. Taking into consideration an example of an email spam classification, the “true” value refers to the binary classification of 1 (spam), whereas “false” refers to 0 (ham). Bearing in mind that positives indicate a correct prediction and negatives indicate the opposite, a

true positive signifies a correct spam classification, and vice versa in terms of true negatives. The same logic is applied to the classification of ham emails as regards to false positives and false negatives. Using these notions, a brief explanation of each metric is given below:

- *Accuracy*

This is the most common metric and perhaps the easiest to understand. It calculates the proportion of results that are correct (refer Eq. 8). [23]

$$Accuracy = \frac{True\ positives + True\ Negatives}{Total\ Predictions}$$

(Eq. 8)

- *Precision*

Precision indicates the ratio of correctly classified outcomes to the total predicted values (refer Eq. 9). [23]

$$Precision = \frac{True\ positives}{True\ positives + False\ Positives}$$

(Eq. 9)

- *Recall*

Recall stipulates the ratio of correctly classified outcomes to just the actual values (refer Eq. 10). [23]

$$Recall = \frac{True\ positives}{True\ positives + False\ Negatives}$$

(Eq. 10)

- F1 Score

Ideally, a model with both high precision and recall would be the desired result. However, there would be a trade-off between the two metrics, meaning the model can either be tweaked to increase precision, but at the expense of weaker recall or vice versa. F1 score is the combination of the two metrics into one by calculating their harmonic mean (refer Eq. 11). [23]

$$F1 = \frac{2 \times precision \times recall}{precision + recall}$$

(Eq. 11)

- *Log Loss*

Log-loss is a metric that stipulates the adjacency of the prediction probability to its corresponding truth value (actual value). The closer the probability converges to the truth value, the lower the value indicated by log-loss (refer Eq. 12). [24]

$$Log\ Loss_i = [y_i \ln(p_i) + (1 - p_i)]$$

(Eq. 12)

Where,

i represents the given document,
y is the actual value,
p denotes the prediction probability, and
ln refers to the natural log (base e).

- *Cohen's Kappa Coefficient*

Since accuracy, precision and recall are not the best measures to resort to in the case of multi-class evaluation, Cohen's kappa score is an exceptional and underutilized metric in statistics that can tackle the problem of multi and imbalanced classes. The score, ranging from -1 to 1, indicates the degree of concurrence between the actual and predicted values. Kappa values beneath 0.4 are generally considered weak, 0.4 - 0.75 is observed as moderate, and any value beyond 0.75 represents an excellent agreement (refer Eq. 13). [25]

$$K = \frac{P_o - P_e}{1 - P_e}$$

(Eq. 13)

- *Matthew's Correlation Coefficient*

Another remarkable metric used to evaluate multi-class classification involves addressing the actual and predicted classes as binary variables and assessing their correlation coefficient. A high correlation between the two values indicates a preferable prediction. This is similar to the statistical concept known as the phi-coefficient (∅) retitled as Matthew's Correlation Coefficient (MCC, refer Eq. 14). [26]

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

(Eq. 14)

6. Results & Discussion

In this section, the results of the aforementioned models are interpreted followed by a comparative analysis of this paper with similar researches. Prior to analyzing the performance of the models, some insights and its interpretation related to the Bayesian Optimization Search (using Optuna) are given below:

Hyperparameter Importances:

Fig. 6 acknowledges the hyperparameters that tend to yield superior results in contrast to the other specifications. As depicted, the max_depth has the most important effect on both the algorithm's performance compared to a low significance for the type of classifier. The Optuna framework only exhibits the intersection of parameters across both models, therefore, the need to further study the parameters individually.

Slice Plots:

This visualization method using Optuna enables the individual study for each hyperparameter and its objective value across all trials. The slice plots for some of the parameters are shown in Fig. 7. This illustrates the distribution of values for reg_alpha, reg_lambda and subsample for the XGBoost model where the darker areas of data points indicate the widely used values across the final trials. These graphs suggest a reduced range of values to be selected for each of the hyperparameters, as it appears to have minimal significance to the objective value of the model. Bearing these insights in mind, future trials can be orchestrated to further increase the performance and efficiency of the optimization models.

Optimization History:

Observing the best line and the distribution of trials with its corresponding f1 scores (objective values) in Fig. 8, it is evident that increasing the trials to a value above 100 would be impractical as Optuna perceived optimal values for the hyperparameters in under 15 trials and was unable to enhance those results.

Classification Results:

The results for the Bayesian Optimization indicating the best hyperparameter choices for each model along with its pre-eminent f1 scores are given in Table 3.

The XGBoost model ostensibly yielded the best scores. Using the predictions and probabilities obtained from the objective function of Optuna, these models are further evaluated (results in Table 4) using the various metrics discussed in Section 5.

On comparing the log loss, kappa and MCC scores of both XGBoost and RandomForest models (from Table 4 and Fig. 9), it can be deduced that the former algorithm produces better predictions compared to the latter.

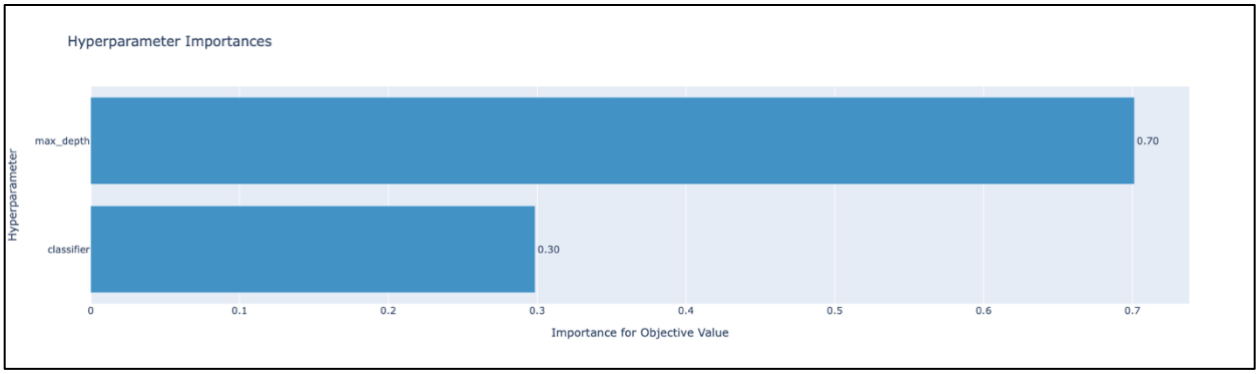

Fig. 6: Hyperparameter Importance for both Algorithms

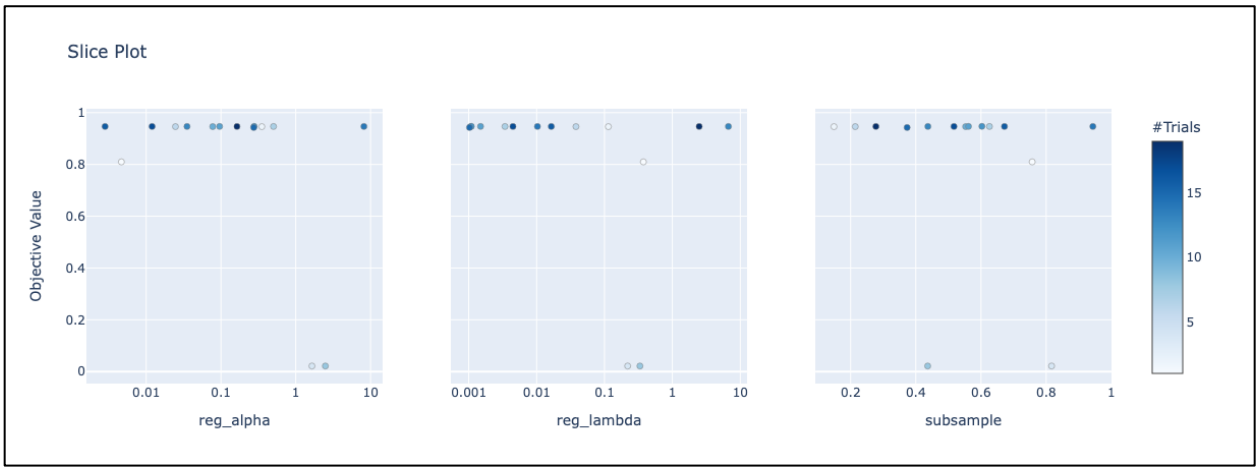

Fig. 7: Slice plots for reg_alpha, reg_lambda and subsample HP's for XGBoost

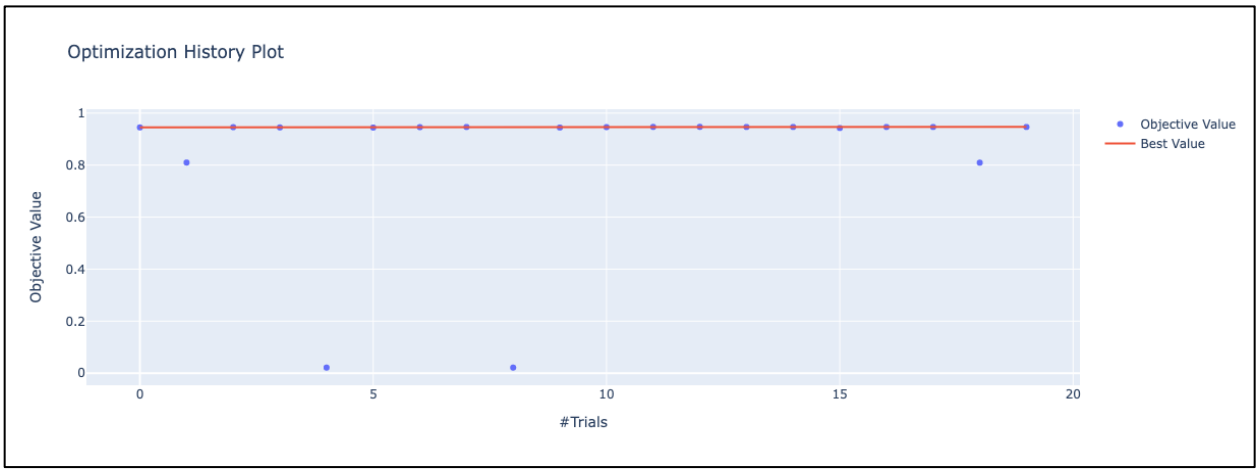

Fig. 8: Slice plots for reg_alpha, reg_lambda and subsample HP's for XGBoost

Model	Duration (In seconds)	Best Parameters	Best F1 Score
XGB	2	'booster': 'gbtree', 'reg_alpha': 0.277462, 'reg_lambda': 0.001093, 'subsample': 0.602853, 'max_depth': 4.493821	0.9471
RandomForest	18	'criterion': 'entropy', 'n_estimators': 137, 'max_depth': 31, 'min_samples_split': 2, 'min_samples_leaf': 11, 'max_features': 2	0.9450

Table 3: Table retrieved by Optuna for best trials

Model	Params	Accuracy	Precision	Recall	F1	Logloss	Kappa	MCC
XGB	'booster': 'gbtree', 'reg_alpha': 0.1702975919423911, 'reg_lambda': 0.12524259351851555, 'subsample': 0.6070142165167571, 'max_depth': 13	0.948	0.957	0.948	0.947	0.130	0.942	0.943
RandomForest	'criterion': 'entropy', 'n_estimators': 137, 'max_depth': 31, 'min_samples_split': 2, 'min_samples_leaf': 11, 'max_features': 2	0.946	0.950	0.946	0.945	0.151	0.939	0.940

Table 4: Best Scores and parameters for each model

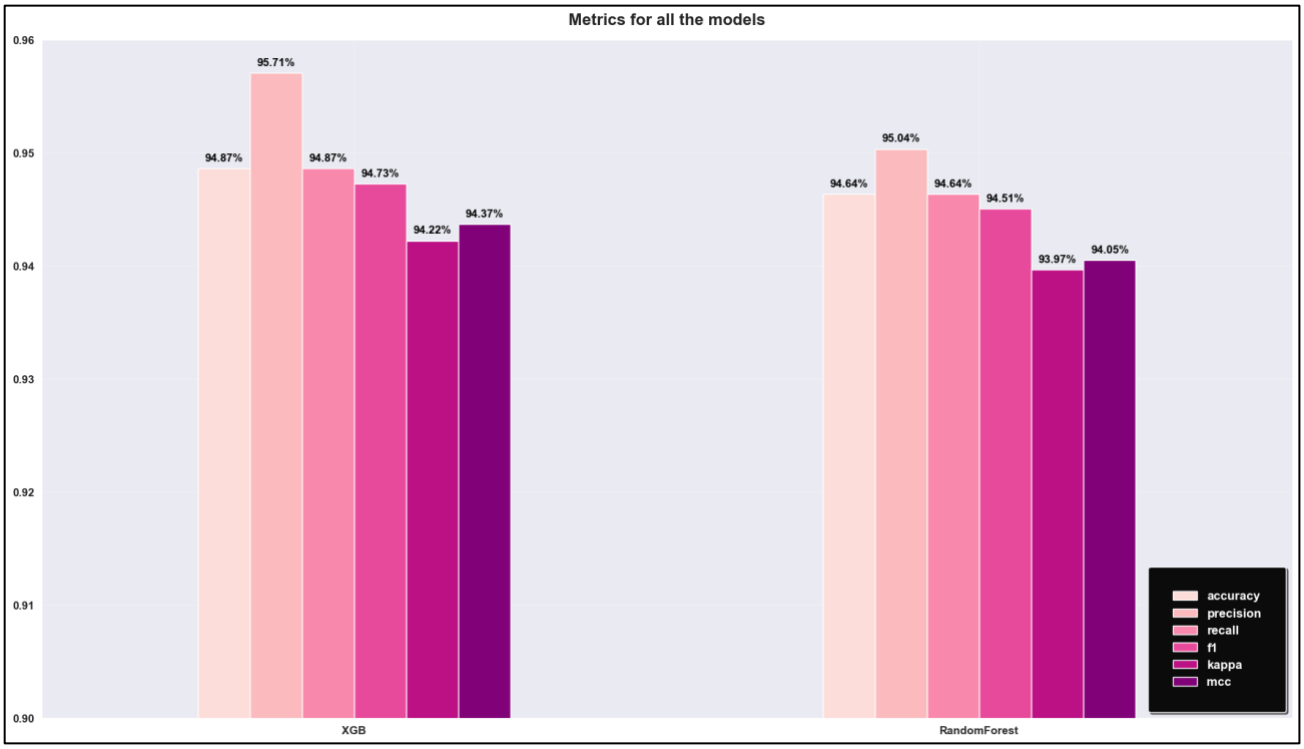

Fig 9: Accuracy, Precision, Recall, F1, Kappa and MCC scores for both models

6. Comparative Analysis

The approach that was proposed in [13] is a suitable candidate for comparison with this study as it shares a number of similarities with this paper. The feature extraction methods employed were Word2Vec and TF-IDF, whereas in this study, the former is the only technique used in this study. The classification technique, however, did not share any similarity. The approach in [13] was to classify tweets into only two different categories, either radical or normal, which is a binary classification task. The study in this paper performs multiclass classification. Lastly, the classification models used upon the dataset in [13] consisted of Random Forest, Neural Network, Support Vector Machine, and K-Nearest Neighbour classifiers whereas this study only made use of the Random Forest and XGBoost models. Their main feature set outputs an accuracy of 80%. The models used in this study were only Random Forest and XGBoost and the accuracy obtained were 94.64% and 94.87% respectively.

Furthermore, [14] conducted text pre-processing contrasting to this study. They used two feature sets, namely, data-dependent and data-independent. This benefitted in classifying extremism from different sources. The pre-processing conducted in this study contained the traditional lemmatization method. The chosen classifier was AdaBoost or Adaptive Boosting. The experiment conducted by the model resulted in an accuracy of 99.5% whereas both the models in this study, Random Forest and XGBoost, resulted in an accuracy score of 94.64% and 94.87% respectively.

8. Conclusion & Future Recommendations

Twitter extremism is a growing concern and tackling it using machine learning is one of the most effective approaches. There is a dire need for systems that are able to accurately identify extremism or collective radicalism. Several machine learning techniques have been trained for these purposes.

In this study, the pre-processing phase consisted of lemmatizing the dataset. Exploratory Data Analysis (EDA) was performed to examine and summarize the dataset in order to understand the characteristics and underlying patterns. The word2vec feature extraction technique was used to capture the semantic meaning of the words. Due to some imbalances that were found in the dataset, the RandomOverSampler method was utilized to address and resolve the issue. It randomly replicated samples from the minority class until the number of samples in each class became equal. This helped in improving the performance by ensuring that the data was balanced, and the algorithms had an equal number of examples to learn from for each class.

Popular machine learning algorithms such as the Random Forest and XGBoost models were used to perform the classification. Hyperparameter tuning was executed to obtain the best results possible out of both models. This was done through Bayesian optimization using an advanced automatic hyperparameter optimization framework called Optuna.

As expected from any research study, there is always plenty of room for improvement in the future. There are a number of recommendations that can be carried out for countering extremism on Twitter and help reduce its dissemination. Some of these implementations include:

- Developing more advanced machine learning algorithms for identifying extremist content on Twitter. This could involve using more sophisticated techniques, such as deep learning and transfer learning to improve the performance of the algorithms. Deep learning algorithms are designed to learn and make predictions based on large amounts of data. They use a hierarchical structure of interconnected “neurons” that can perform comparatively better than traditional machine learning algorithms that are often less effective when working on a huge dataset and may struggle to make accurate predictions or extract useful insights from the data.

- Expanding the training dataset to include a greater diversity of extremist content, such as different types of extremist ideologies. This can help the algorithms to analyze new and unseen data.

- BERT (Bidirectional Encoder Representations from Transformers) is a type of deep learning model that helps algorithms understand the meaning and intent behind words and sentences from a document. Utilizing it as a Feature Extraction method can potentially make it more effective at identifying hate speech and violent language compared to other models.

- Using more models and batches in hyperparameter tuning can help in improving classification by allowing the optimizer to identify the most significant parameters in addition to exploring an extensive range of hyperparameter values to attain the most favorable results. Moreover, the utilization of sophisticated computational resources can further aid in speeding up the optimization process.

- Conducting research to evaluate more of the psychological and social factors that can potentially contribute to extremism and using this knowledge to develop more effective interventions for preventing radicalization. This could involve using NLP techniques to analyze social media data and identify potential risk factors for radicalization. As discussed in Section 3, [9]'s proposal suggested developing a psychological profile for extremist Twitter users by detecting signals that indicate that the user carries the potential to spread propaganda regarding any form of violent extremism or collective radicalisation. This proposal would greatly help in identifying and more importantly, preventing the spread of such content as it considers the psychological aspects of the users' behavior.

These endeavors, in the end, would allow law enforcement and intelligence agencies to investigate and successfully intercept individuals or groups who are thought to follow suspicious radicalization routes into extremism.

9. References

[1] *Twitter and Violent Extremism* n.d., viewed 7 December 2022, *<https://cops.usdoj.gov/RIC/Publications/cops-w0741-pub.pdf>.*

[2] Ali, J, Khan, R, Ahmad, N & Maqsood, I 2012, *Random Forests and Decision Trees*, ResearchGate, unknown, viewed 7 December 2022, *<https://www.researchgate.net/publication/259235118_Random_Forests_and_Decision_Trees>.*

[3] Wood, T 2020, *Random Forests*, DeepAI, DeepAI, viewed 7 December 2022, *<https://deepai.org/machine-learning-glossary-and-terms/random-forest>.*

[4] Yiu, T 2019, *Understanding Random Forest - Towards Data Science*, Medium, Towards Data Science, viewed 7 December 2022, *<https://towardsdatascience.com/understanding-random-forest-58381e0602d2>.*

[5] Mello, A 2020, *XGBoost: theory and practice - Towards Data Science*, Medium, Towards Data Science, viewed 7 December 2022, *<https://towardsdatascience.com/xgboost-theory-and-practice-fb8912930ad6>.*

[6] Ibrahem Ahmed Osman, A, Najah Ahmed, A, Chow, MF, Feng Huang, Y & El-Shafie, A 2021, 'Extreme gradient boosting (Xgboost) model to predict the groundwater levels in Selangor Malaysia', *Ain Shams Engineering Journal*, vol. 12, no. 2, pp. 1545–1556, viewed 7 December 2022, *<https://www.sciencedirect.com/science/article/pii/S2090447921000125>.*

[7] Singh, AK & Shashi, M 2019, 'Vectorization of Text Documents for Identifying Unifiable News Articles', International Journal of Advanced

Computer Science and Applications, vol. 10, no. 7.

[8] Yang, X, Yang, K, Cui, T & He, L 2022, *A Study of Text Vectorization Method Combining Topic Model and Transfer Learning*, ResearchGate, unknown, viewed 8 December 2022, *https://www.researchgate.net/publication/358585623_A_Study_of_Text_Vectorization_Method_Combining_Topic_Model_and_Transfer_Learning*.

[9] Wisam Abdulazeez Qader, Musa M.Ameen & Bilal Ismael Ahmed 2019, *An Overview of Bag of Words;Importance, Implementation, Applications, and Challenges*, ResearchGate, unknown, viewed 8 December 2022, *<https://www.researchgate.net/publication/338511771_An_Overview_of_Bag_of_WordsImportance_Implementation_Applications_and_Challenges>.*

[10] Pannakkong, W, Thiwa-Anont, K, Singthong, K, Parthanadee, P & Buddhakulsomsiri, J 2022, 'Hyperparameter Tuning of Machine Learning Algorithms Using Response Surface Methodology: A Case Study of ANN, SVM, and DBN', in K-H Chang (ed.), *Mathematical Problems in Engineering*, vol. 2022, pp. 1–17, viewed 8 December 2022, *<https://www.hindawi.com/journals/mpe/2022/8513719/>.*

[11] Shekhar, S, Bansode, A & Salim, A n.d., *A Comparative study of Hyper-Parameter Optimization Tools*.

[12] Natarajan, R, Barua, G & Manas Ranjan Patra 2015, *Distributed Computing and Internet Technology*, Springer, pp. 444–455, viewed 6 December 2022, *<https://www.researchgate.net/profile/Krithivasan-Ramamritham/publication/269230368_Putting_smart_meters_to_work/links/5645f6b308ae9f9c13e72985/Putting-smart-meters-to-work.pdf#page=444>.*

[13] Nouh, M, Nurse, J & Goldsmith, M 2019, *Understanding the Radical Mind: Identifying Signals to Detect Extremist Content on Twitter*.

[14] Kaati, L, Omer, E, Prucha, N & Shrestha, A 2015, 'Detecting Multipliers of Jihadism on Twitter', *2015 IEEE International Conference on Data Mining Workshop (ICDMW)*.

[15] Divya Khyani & S, SB 2021, *An Interpretation of Lemmatization and Stemming in Natural Language Processing*, ResearchGate, unknown, viewed 8 December 2022, *<https://www.researchgate.net/publication/348306833_An_Interpretation_of_Lemmatization_and_Stemming_in_Natural_Language_Processing#:~:text=What%20is%20Lemmatization%3F,adds%20meaning%20to%20particular%20words.>.*

[16] Pykes, K 2020, *Oversampling and Undersampling - Towards Data Science*, Medium, Towards Data Science, viewed 7 December 2022, *<https://towardsdatascience.com/oversampling-and-undersampling-5e2bbaf56dcf>.*

[17] Soni, P 2020, *Handling Imbalanced Datasets With imblearn Library - TheCyPhy - Medium*, Medium, TheCyPhy, viewed 7 December 2022, *<https://medium.com/thecyphy/handling-imbalanced-datasets-with-imblearn-library-df5e58b968f4>.*

[18] Brownlee, J 2019, *How to Implement Bayesian Optimization from Scratch in Python - MachineLearningMastery.com*, MachineLearningMastery.com, viewed 6 December 2022, *<https://machinelearningmastery.com/what-is-bayesian-optimization/>.*

[19] Alarie, S, Audet, C, Gheribi, AE, Kokkolaras, M & Le Digabel, S 2021, 'Two decades of blackbox optimization applications', *EURO Journal on Computational Optimization*, vol. 9, p. 100011, viewed 6 December 2022, *<https://www.sciencedirect.com/science/article/pii/S2192440621001386>.*

[20] Lim, Y 2022, *State-of-the-Art Machine Learning Hyperparameter Optimization with Optuna*, Medium, Towards Data Science, viewed 6 December 2022, *<https://towardsdatascience.com/state-of-the-art-machine-learning-hyperparameter-optimization-with-optuna-a315d8564de1>.*

[21] Chauhan, A 2021, Random Forest Classifier and its Hyperparameters - Analytics Vidhya - Medium, Medium, Analytics Vidhya, viewed 6 December 2022, *<https://medium.com/analytics-vidhya/random-forest-classifier-and-its-hyperparameters-8467bec755f6>.*

[22] Alam, M 2021, *A guide to XGBoost hyperparameters - Towards Data Science*, Medium, Towards Data Science, viewed 6 December 2022, *<https://towardsdatascience.com/a-guide-to-xgboost-hyperparameters-87980c7f44a9>.*

[23] Chowdhury, S & Schoen, MP 2020, *Research Paper Classification using Supervised Machine Learning Techniques*, ResearchGate, unknown, viewed 8 December 2022, *<https://www.researchgate.net/publication/346853360_Research_Paper_Classification_using_Supervised_Machine_Learning_Techniques>.*

[24] Gaurav Dembla 2020, *Intuition behind Log-loss score - Towards Data Science*, Medium, Towards Data Science, viewed 7 December 2022, <*https://towardsdatascience.com/intuition-behind-log-loss-score-4e0c9979680a*>.

[25] Pykes, K 2020, Cohen's Kappa - Towards Data Science, Medium, Towards Data Science, viewed 7 December 2022, *<https://towardsdatascience.com/cohens-kappa-9786ceceab58#:~:text=agree%20by%20chance.-,Evaluating%20Cohen's%20Kappa,less%20agreement%20than%20random%20chance.>.*

[26] Boaz Shmueli 2019, *Matthews Correlation Coefficient is The Best Classification Metric You've Never Heard Of*, Medium, Towards Data Science, viewed 7 December 2022, *<https://towardsdatascience.com/the-best-classification-metric-youve-never-heard-of-the-matthews-correlation-coefficient-3bf50a2f3e9a>.*