FAKE NEWS DETECTION AND CLASSIFICATION

Enrollment Number(s)- 16103044, 16103053, 16103127

Name of Student(s)- Prakhardev Singh, Siddhi Shree, Vaibhav Kaushik

Name of Supervisor(s)- Ms. Sakshi Agarwal

December 2019

Submitted in partial fulfillment of the Degree of

Bachelor of Technology in

Computer Science Engineering

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING AND INFORMATION TECHNOLOGY,

JAYPEE INSTITUTE OF INFORMATION TECHNOLOGY, NOIDA

(I)

TABLE OF CONTENTS

(II)

DECLARATION

We hereby declare that this submission is our own work and that, to the best of our knowledge and belief, it contains no material previously published or written by another person nor material which has been accepted for the award of any other degree or diploma from a university or other institute of higher learning, except where due acknowledgment has been made in the text.

Place:
Date:

Signature:
Name:
Enrollment No:

Signature:
Name:
Enrollment No:

Signature:
Name:
Enrollment No:

(III)

CERTIFICATE

This is to certify that the project titled “Fake News Classification”, submitted by “Prakhar Dev Singh, Siddhi Shree, Vaibhav Kaushik” in partial fulfillment for the award of the degree of Bachelor of Technology in Computer Science of Jaypee Institute of Information Technology, Noida has been carried out under my supervision. This work has not been submitted partially or wholly to any other University or Institute for the award of this or any other degree or diploma.

Signature of Supervisor ……………………..

Name of Supervisor ……………………..

Designation ……………………..

Date ……………………..

(IV)

ACKNOWLEDGEMENT

We would like to express our sincere gratitude to our evaluation committee - Ms. Kirti Aggarwal and Ms. Kavita Pandey for providing a just and fair evaluation of our efforts and understanding the core of our ideologies kept in mind while working for this project. We would especially like to thank our supervisor - Ms. Sakshi Agarwal for providing her invaluable guidance, comments, and suggestions throughout the course of the project.

Signature of the Student(s)
Name(s):
Date:

Signature of the supervisor(s)
Name(s):
Date:

(V)

SUMMARY

Social media has been drastically changing the way news content is produced, propagated and consumed by the general public. It is definitely opening many opportunities, but at the same time, it is creating complex challenges, specifically regarding the separation of real news from a rumor. A key problem today is that social media has become a place for campaigns of misinformation that affect the credibility of the entire news ecosystem. The majority of existing detection algorithms focus on finding clues from news content, which are generally not effective because fake news is often intentionally written to mislead users by mimicking true news. We need to explore auxiliary information to improve detection. Therefore, we aim to propose a novel solution to the problem of fake news detection and classification by exploring various "user engagement features" in addition to exploring the textual features.

Signature of the Student(s)
Name(s):
Date:

Signature of the supervisor(s)
Name(s):
Date:

(VI)

List of figures

(VII)

List of Tables

1. Introduction

1.1 General Introduction

Social media for news consumption is a double-edged sword. On the one hand, its low cost, easy access, and rapid dissemination of information leads people to seek out and consume news from social media. On the other hand, it enables the widespread of “fake news”, i.e., low-quality news with intentionally false information. The extensive spread of fake news has the potential for extremely negative impacts on individuals and society. Therefore, fake news detection on social media has recently become emerging research that is attracting tremendous attention. Detecting fake news is an important task, which not only ensures users receive authentic information but also helps maintain a trustworthy news ecosystem.

1.2 Problem Statement

The project’s objective is to develop a fake news detector and classifier,using Artificial Intelligence and Supervised Machine Learning techniques.media. To accomplish this goal, we explore several types of features extracted from news stories, including content and posts from social media. We present a new set of features and measure the prediction performance of current approaches and features for automatic detection of fake news. The results reveal interesting findings on the usefulness and importance of features for detecting false news.In this project, we conclude that the hybrid features set consisting of both, textual features as well as social context features would constitute a much better feature set for the classification models in comparison to the models which use either textual features or social context features.

1.3 Significance of the problem

Fake news can become extremely influential and has the ability to spread exceedingly fast. With the increase of people using social media, they are being exposed to new information and stories every day. Misinformation can be difficult to correct and may have lasting implications. In addition, since false information is able to spread so fast, not only does it have the ability to harm people, but it can also be detrimental to huge corporations and even the stock market. The motivation behind this project is to help a naive user identify a credible information source, and not get misled by clickbaits, hoaxes or intentional fake propaganda.

1.4 Challenges faced

Detecting fake news on social media presents unique challenges.

First, fake news pieces are intentionally written to mislead consumers, which makes it not satisfactory to spot fake news from news content itself. For example, fake news may cite true evidence within the incorrect context to support a non-factual claim.. Thus, existing hand-crafted and data-specific textual features are generally not sufficient for fake news detection.Thus, we need to explore information in addition to news content, such as user engagements and social behaviors of users on social media.For example, a credible user's comment that “This is fake news” is a strong signal that the news may be fake.

Second, exploiting this auxiliary information actually leads to another critical challenge: the quality of the data itself. The lack of comprehensive and community-driven fake news datasets has become one of the major roadblocks.. Not only existing datasets are scarce,they do not contain a myriad of features often required in the study such as news content, and social context. In addition, users' social engagements with fake news produce data that is big, incomplete, unstructured, and noisy. Effective methods to differentiate credible users, extract useful post features and exploit network interactions are an open area of research and need further investigations.

1.5 Brief description of the solution approach

Our approach aims to rigorously analyse a “twitter” dataset containing various news articles. First,we extract the textual features from the content itself using various types of vectorizers, thereafter we extract the features related to the “social context” of the news i.e various “user engagement features” on twitter from the associated metadata. We finally apply the techniques of supervised machine learning to come up with a model to classify the news articles as fake or real.

1.6 Comparison of existing approaches to the problem framed

Existing approaches for fake news classification focus on either of the two procedures:

1.6.1 Considering only text features- lexical features,semantic features etc.

1.6.2 Considering only social context features- user engagement features.

In comparison to the above two standard approaches ,we sought to extract useful user engagement features and combine these two broad types of features to build a “hybrid” feature set..

2. Literature Survey

2.1 Summary of the papers studied

Our task of coming up with the best suitable model for classifying news as fake or real began with researching about news propagation on social media platforms like twitter.n. A key driving force behind the diffusion of information is its spreaders. People tend to spread information that caters to their interests and/or ts their system of belief.

Moreover, the main question was - "When a message, such as a piece of news, spreads in social networks, how can we classify it into categories of interests, such as genuine or fake news?". This question motivated us to trace the most important features which could be used alongside the text content of the related post,which proved to be a good starting point for fake news detection.We focused our analysis on understanding how various features can be used to discriminate true and fake news.

On a coarse-grained level, features for fake news detection can be roughly categorized into two categories. First, Features extracted from news content(e.g., language processing techniques) ie. Textual Features consisting of the information extracted from the news text, including the text body, the headline, and the text message used by the news source. For example - Language Features, Lexical Features, Semantic Features etc.Most existing methods regard fake news detection as a text categorization problem and mainly focus on using content features, such as words and hashtags.s. However, for many emerging applications like fake news and rumor detection, it is very challenging, if not impossible, to identify useful features from content. For example, intentional spreaders of fake news may manipulate the content to make it look like real news. To address this problem,a second type, Features extracted from the environment (e.g., social network structure) have been recognised. Environment Features consist of statistics of user engagement and temporal patterns from social media (i.e., Twitter).These include engagement features like number of likes, shares, and comments from twitter users. Moreover, the number of comments within intervals from publication time and the rate of posting comments can be used to capture temporal patterns .

After figuring out the various features required,there were adequate features available which could give a good insight about how the fake news articles are different from real news,how do the users get convinced to believe on such hoax articles and how text can be leveraged to perform the classification. The next topic for research was processing of textual features.

The processing of news content of articles can be traced back to text classification, which was achieved in two ways: manual and automatic classification. In the former, a human annotator interprets the content of text and categorizes it accordingly. This method usually can provide quality results but it's time-consuming and expensive. The latter applies machine learning, natural language processing, and other techniques to automatically classify text in a faster and more cost-effective way. Instead of relying on manually crafted rules, text classification with machine learning learns to make classifications based on past observations. By using pre-labeled examples as training data, a machine learning algorithm can learn the different associations between pieces of text and that a particular output (i.e. tags) is expected for a particular input (i.e. text).

The first step towards training a classifier with machine learning is feature extraction: a method is used to transform each text into a numerical representation in the form of a vector. One of the most frequently used approaches is bag of words, where a vector represents the frequency of a word in a predefined dictionary of words.One of the major disadvantages of using BOW is that it discards word order thereby ignoring the context and in turn meaning of words in the document. For natural language processing (NLP) maintaining the context of the words is of utmost importance. This problem led our research towards another approach called Word Embedding. It represents words in a coordinate system where related words, based on a corpus of relationships, are placed closer together.

Then, the machine learning algorithm is fed with training data that consists of pairs of feature sets (vectors for each text example) and tags (e.g. sports, politics) to produce a classification model.Once it’s trained with enough training samples, the machine learning model can begin to make accurate predictions.One of the most popular machine learning algorithms for creating text classification models include the “naive bayes algorithm”. One of the members of that family is Multinomial Naive Bayes (MNB). One of its main advantages is that we can get really good results when data available is not much (~ a couple of thousand tagged samples) and computational resources are scarce.

Further research led to the realization that Text classification algorithms like the naive bayes has its own limitations,like, Naive Bayesian classifiers assume that the effect of an attribute value on a given class is independent of the values of the other attributes. This assumption is called “class conditional independence”. This assumption generally doesn’t hold true in many situations and when we are dealing with complex text content of fake and real news,it becomes all the more clear that naive bayes could not be a very accurate model to build our classifier.

Hence we researched other classification algorithms like decision trees,SVM,Logistic regression etc,but to attain even better accuracies Data scientists and artificial intelligence systems experts have been shifting towards “Ensemble learning” which have proven to be extremely useful in the past couple of years.Ensemble methods can be used to increase overall accuracy by learning and combining a series of individual (base) classifier models.Bagging,boosting, and random forests are examples of ensemble methods.

Further Research led to the discovery of a few limitations to ensemble learning which are as follows:

1) Reduction in model interpretability- Using ensemble methods reduces the model interpret-ability due to increased complexity and makes it very difficult to draw any crucial business insights at the end.

2) Computation and design time is high- It is not good for real time applications.
3) The selection of models for creating an ensemble is an art which is really hard to master.

After learning about the ensemble learning we also came across a different class of algorithms known as “passive aggressive classifiers”.Passive Aggressive Algorithms are a family of online learning algorithms (for both classification and regression).In computer science, online machine learning is a method of machine learning in which data becomes available in a sequential order and is used to update our best predictor for future data at each step, as opposed to batch learning techniques which generate the best predictor by learning on the entire training data set at once.The idea of passive aggressive algorithms is very simple and their performance has been proved to be superior to many other alternative methods like Online Perceptron.They are a family of algorithms for large-scale learning.

Cross-validation is a common method to evaluate the performance of a text classifier.Further Research led us towards performance metrics that are useful for a quick assessment on how well a classifier works- Accuracy,Precision,Recall and F1 Score.

2.2 Integrated summary of the literature studied

There have been many researches that aim to address the problem of efficient fake news classification but most of them fall short on one of the following parameters:

2.2.1 Feature Consideration

2.2.2 Feature Extraction

2.2.3 Classification algorithm used

Our approach aims to consider both the features, i.e textual as well as the social context features.Secondly, we try to identify the best suitable vectorizer for feature extraction and the most appropriate classification algorithm amongst the three studied above.

3. Requirement Analysis and Solution Approach

3.1 Overall description of the project

The project aims to help mitigate the negative effects caused by fake news–both to benefit the public and the news ecosystem,by identifying the best suitable method for fake news detection and classification. After constructing a feature-rich dataset, we process the textual content by extracting features using different types of vectorizers.Thereafter, we apply various text-classification algorithms and try to identify the best suitable approach using cross validation.

3.2 Requirement Analysis

3.2.1 Functional Requirements

3.2.1.1 The model is able to extract textual and user engagement features from a given news article.

3.2.1.2 The model is able to classify a given article as fake or real.

3.2.1.3 The model can be used for clickbaits and spam detection.

3.2.2 Non-Functional Requirements

3.2.2.1 The model is able to process a large dataset i.e. Scalable model.

3.2.2.2 The model is able to identify fake news with great accuracy.

3.2.2.3 The model is able to give precise results

3.3 Solution Approach

Our approach aims to rigorously analyse a “twitter” dataset containing various news articles. First,we extract the textual and “user engagement” features from the content using data crawling. Later, We apply vectorisation using three different types of vectorizer - Count Vectorizer, Tf-Idf vectorizer and Word2Vec.Thereafter, we apply normalisation to the dataset and construct a correlation heat map to verify social context features. We finally apply the text classification techniques - Naive Bayes, Ensemble and Passive Agressive to come up with the best suitable model to classify the news articles as fake or real.

4. Modeling and Implementation Details

4.1 Design Diagrams

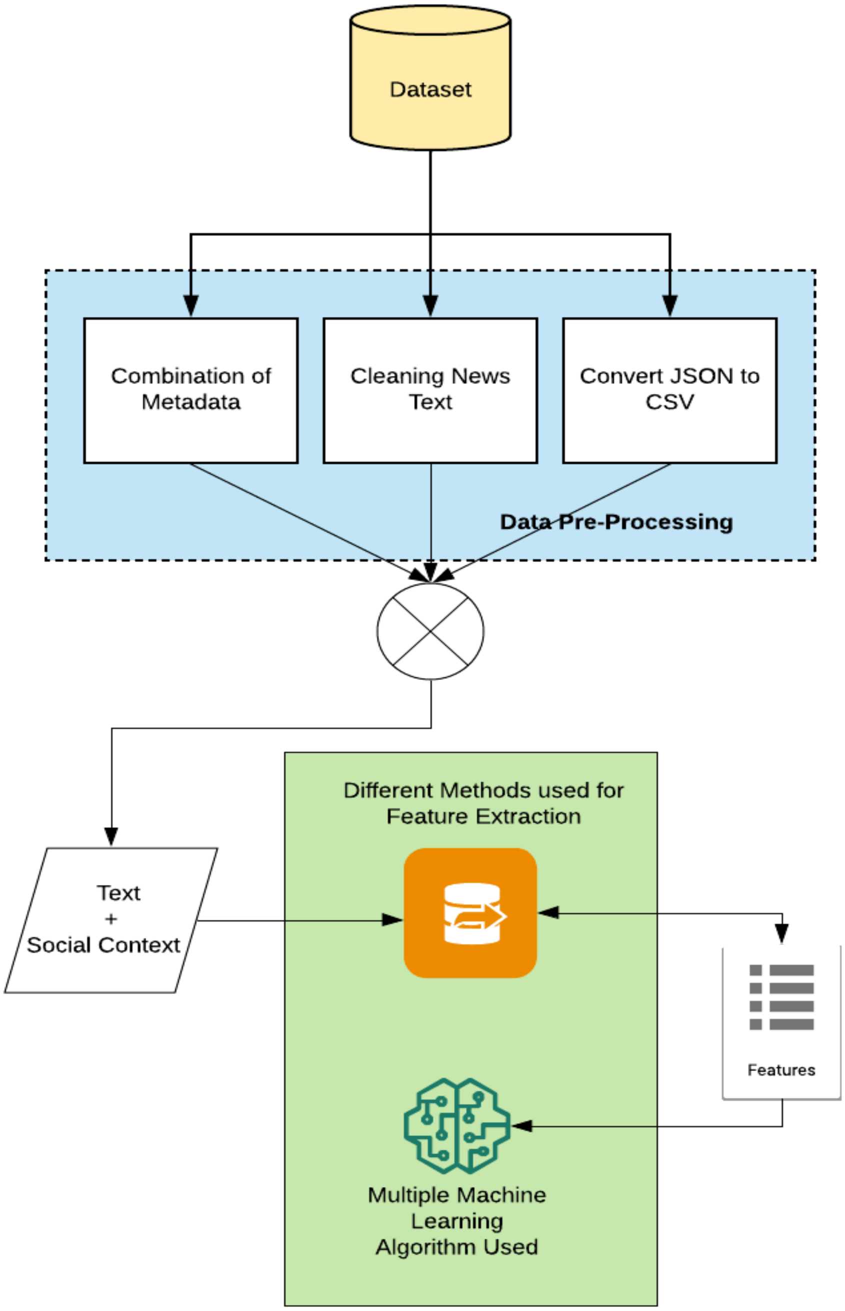

Figure 1:An abstract view of how the project works:

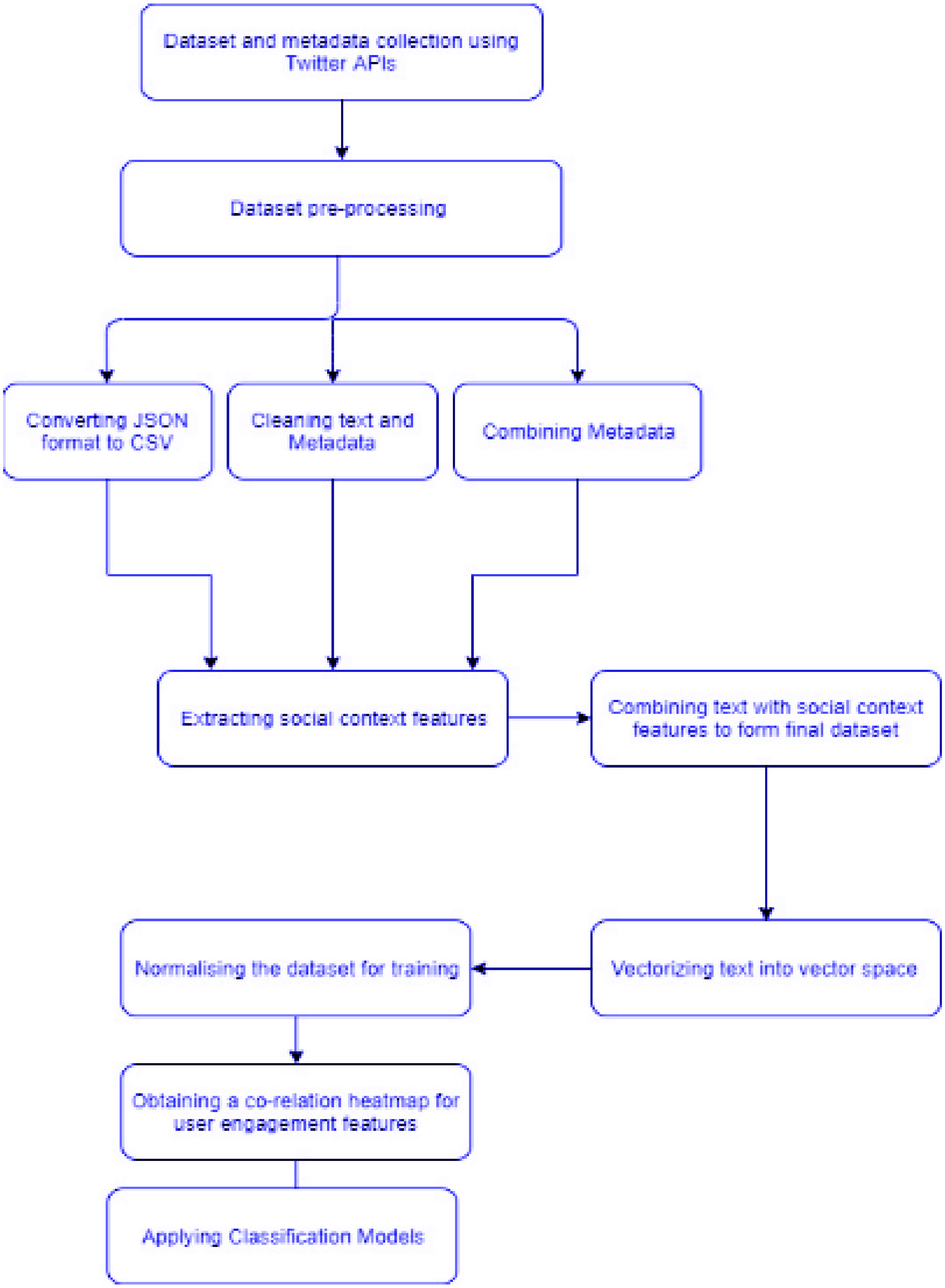
Dataset and metadata collection using Twitter APIs
Dataset pre-processing
Converting JSON format to CSV
Cleaning text and Metadata
Combining Metadata
Extracting social context features
Combining text with social context features to form final dataset
Vectorizing text into vector space
Normalising the dataset for training
Obtaining a co-relation heatmap for user engagement features
Applying Classification Models

Figure 2:Flow-chart describing workflow

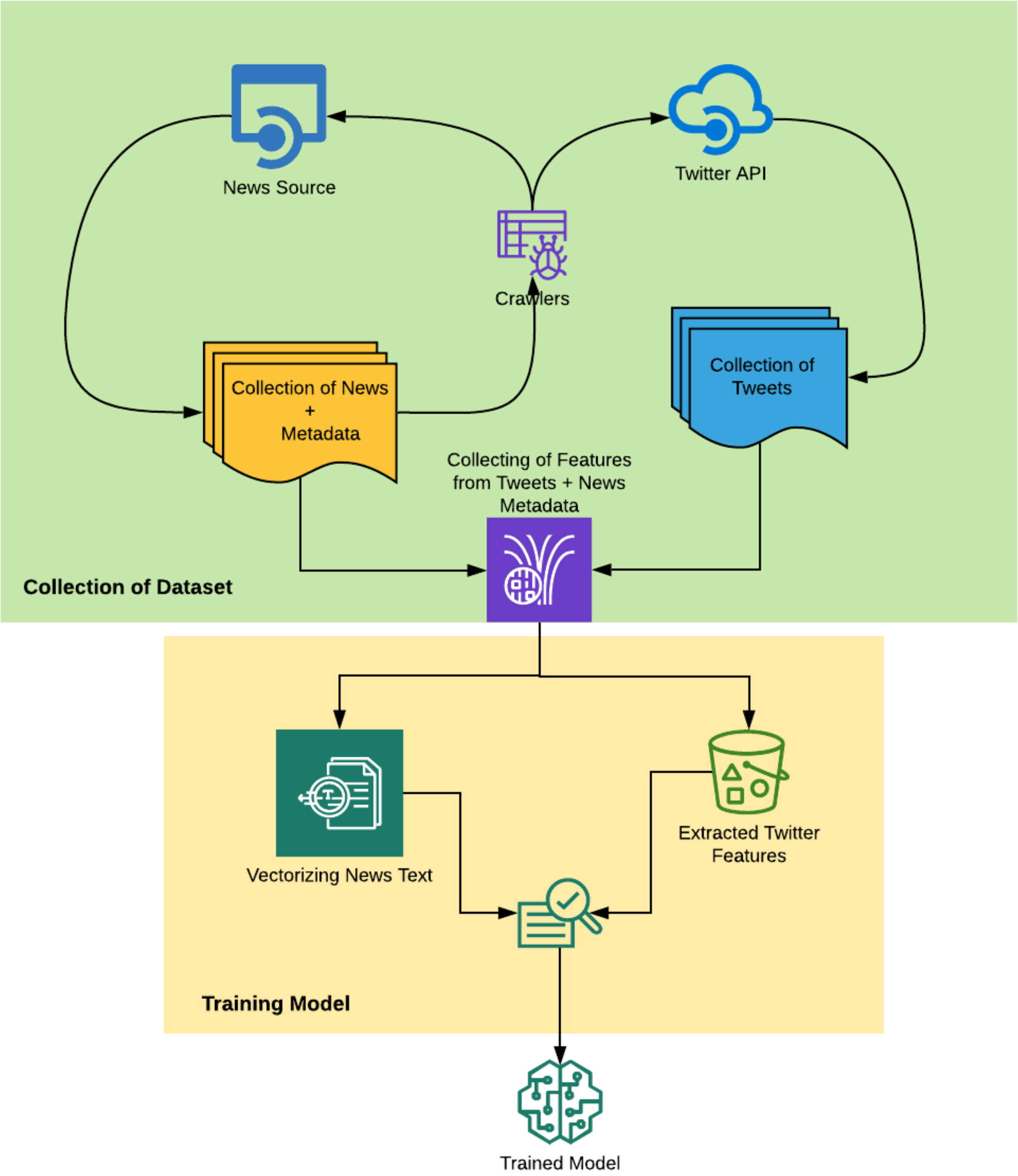

Figure 3. A Representation of the Control flow of Project

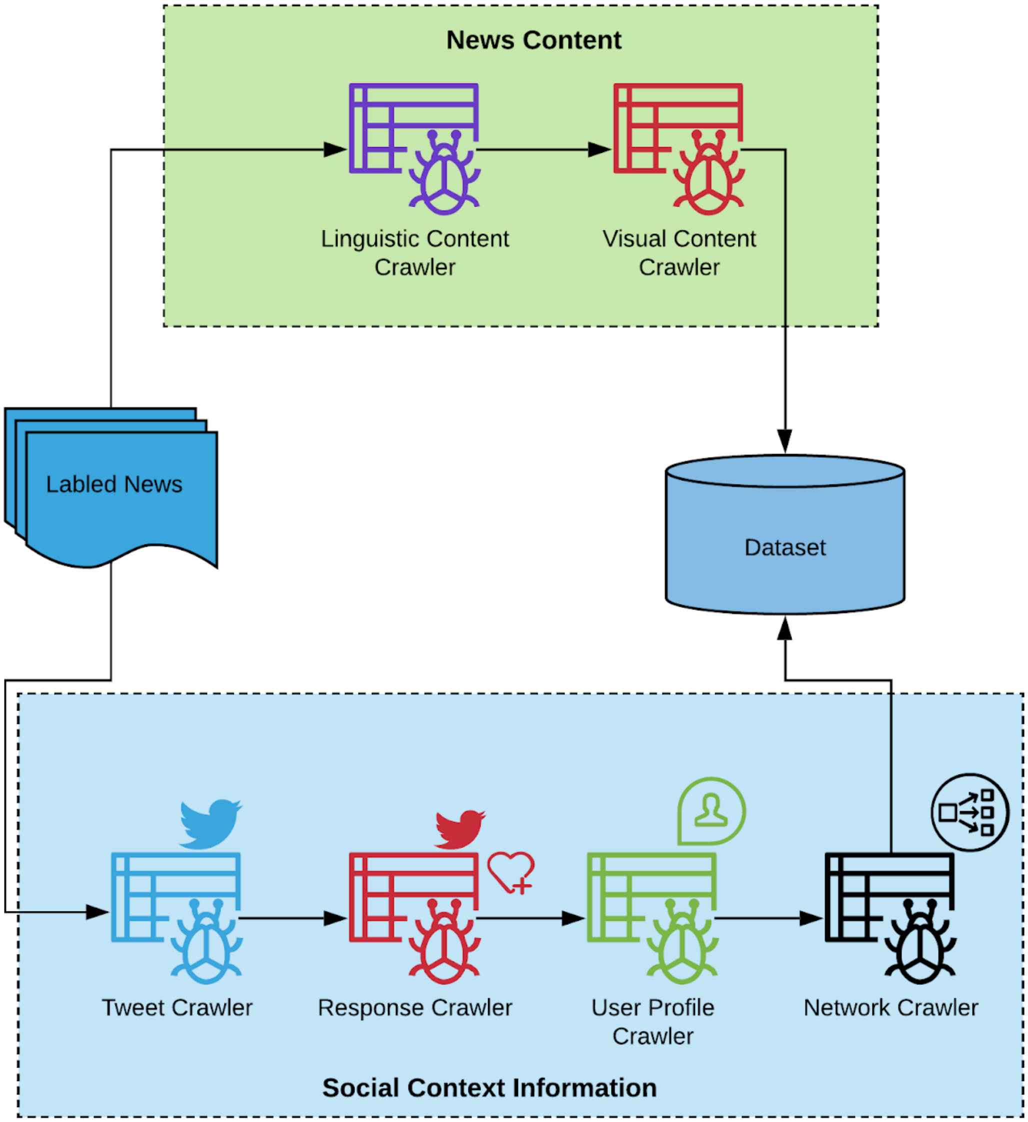

Figure 4. Flowchart of Dataset collection

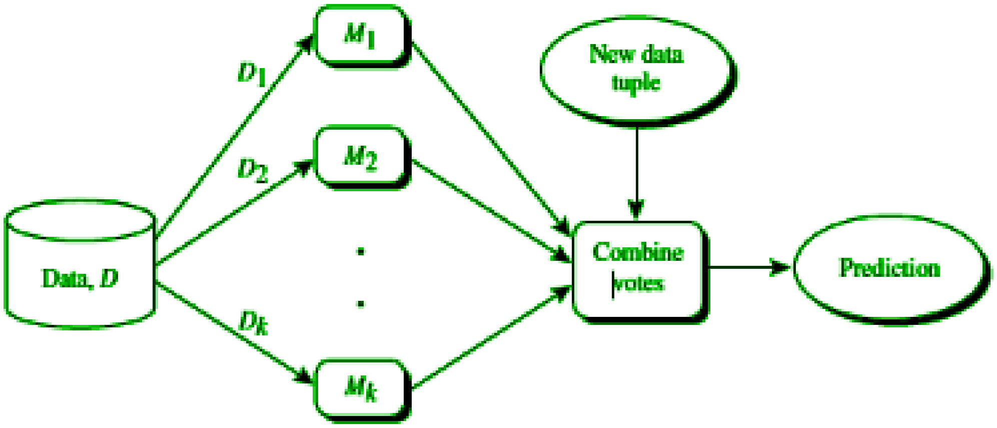

Figure 5: Ensemble Learning

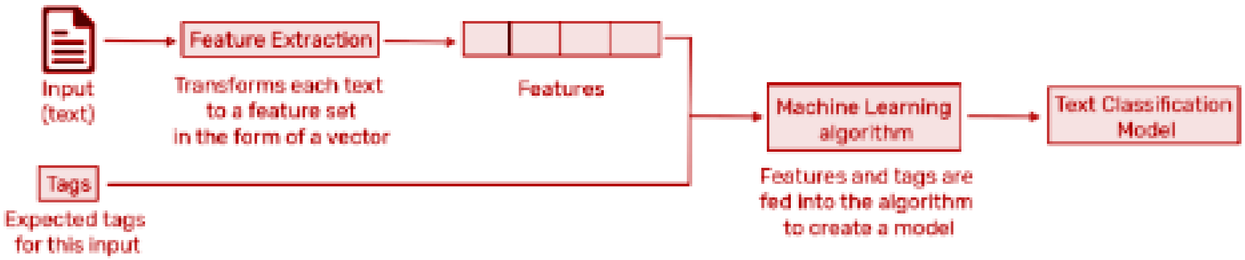
Input (text)
Feature Extraction
Transforms each text to a feature set in the form of a vector
Features
Tags
Expected tags for this input
Machine Learning algorithm
Features and tags are fed into the algorithm to create a model
Text Classification Model

Figure 6: Use Case Diagram

4.2 Implementation Details

4.2.1 Data Collection

In our project, we use a dataset consisting of fake and real news, and try to extract various features associated with each news.This feature extraction process has manifold benefits- First, the rich set of features in the datasets provides an opportunity to experiment with different approaches for fake news detection, understand the diffusion of fake news in social network and intervene in it.Second, the temporal information enables the study of early fake news detection by generating synthetic user engagements from historical temporal user engagement patterns in the dataset.

To collect reliable ground truth labels for fake news, we utilize fact-checking websites to obtain news content for fake news and true news such as GossipCop.The user engagements related to the fake and real news pieces from fact-checking websites are collected using search API provided by social media platforms such as Twitter's Advanced Search API. After obtaining the social media posts that directly spread news pieces,we further fetch the user response towards these posts such as replies, likes, and reposts.The temporal information indicates that we record the timestamps of user engagements, such as the number of comments within intervals from publication time and the rate of posting comments.

4.2.2 Data Pre-Processing

The available dataset was in JSON format and was converted to .csv for processing.Dataset cleaning was performed on the dataset which included removal of certain rows that contained null values and removal of certain columns containing only noise which might have crept in while crawling or parsing the dataset or assembling the dataset to the required format.

4.2.3 Feature Extraction

Following social context features were extracted from the cleaned dataset:

4.2.3.1 **Tweets**: it describes the total number of tweets related to each news article.

4.2.3.2 **Retweet_count**: Total number of retweets of all the users associated with the news article.

4.2.3.3 **Favourite_count**: Total number of like of all the tweets associated with the news article.

4.2.3.4 **Is_verified_count**: Total number of verified accounts associated with direct tweet of news articles.

4.2.3.5 **mean_time_diff**: The mean time difference between user's account creation and tweet time of the particular news article.

4.2.3.6 **popularity_score**: This score indicates how popular a news is expected to get.It is reflected by the popularity of the users associated with the news articles.This is calculated by first subtracting the "friends_count" from the "follower_count" from the metadata and then summing over all such values for every user associated with a particular news.

4.2.4. Text analysis and Vectorisation

For Text Analysis, raw data as a sequence of symbols cannot be fed directly to the algorithms themselves as most of them expect numerical feature vectors with a fixed size rather than the raw text documents with variable length.In order to address this,first tokenize strings and give an integer id for each possible token, for instance by using white-spaces and punctuation as token separators.It is followed by counting the occurrences of tokens in each document and normalizing and weighting with tokens that occur in the majority of samples / documents in decreasing order of importance.

In this scheme, features and samples are defined by treating each individual token occurrence frequency (normalized or not) as a feature.The vector of all the token frequencies for a given document is considered a multivariate sample.

A corpus of documents can thus be represented by a matrix with one row per document and one column per token.We call vectorization the general process of turning a collection of text documents into numerical feature vectors. This specific strategy (tokenization, counting and normalization) is called the **Bag of Words or "Bag of n-grams" representation**. Documents are described by word occurrences while completely ignoring the relative position information of the words in the document.

Count-Vectorizer: This is the most simple vectorizer which is most commonly used.The main point to note here is that following from the points mentioned above it follows the simple "bag of words" assumption. implements both tokenization and occurrence counting

Tf-Idf vectorizer:Tf means term-frequency while tf–idf means term-frequency times inverse document-frequency ie tf-idf(t,d)=tf(t,d)*idf(t)..It is a better technique than the count vectorizer because of the weighting scheme used which count vectorizer clearly lacks.In a large text corpus, some words will be very present (e.g. "the", "a", "is" in English) hence carrying very little meaningful information about the actual contents of the document. If we were to feed the direct count data directly to a classifier those very frequent terms would shadow the frequencies of rarer yet more interesting terms.In order to re-weight the count features into floating point values suitable for usage by a classifier it is very common to use the tf–idf transform.

One of the major disadvantages of using BOW is that it discards word order thereby ignoring the context and in turn meaning of words in the document. For natural language processing (NLP) maintaining the context of the words is of utmost importance. To solve this problem we use another approach called Word Embedding.

Word Embedding: It is a representation of text where words that have the same meaning have a similar representation. In other words it represents words in a coordinate system where related words, based on a corpus of relationships, are placed closer together.

Word2vec takes as its input a large corpus of text and produces a vector space with each unique word being assigned a corresponding vector in the space. Word vectors are positioned in the vector space such that words that share common contexts in the corpus are located in close proximity to one another in the space.

The basic idea is that semantic vectors (such as the ones provided by Word2Vec) should preserve most of the relevant information about a text while having relatively low dimensionality which allows better machine learning treatment than straight one-hot encoding of words. Another advantage of topic models is that they are unsupervised so they can help when labeled data is scarce.

4.2.5 Normalisation

In order to have the same range of values for every feature set that would be trained,we applied normalisation wherever required,thus converting all values of all the columns in the range of 0 and 1.Countvectorizer and tf-idf vectorizers by default achieve normalisation.In situations like using word2vec for only text features, considering only social context features and considering hybrid of all the features, we have explicitly applied normalisation.

4.2.6 Correlation Heat map

To see whether the extracted social context features contribute significantly to our models and “linearly” correlate well ,we constructed a correlation matrix of all such features,We then plotted the same matrix with the correlation scores corresponding to each combination of features.the feature “mean_time_diff” was the only feature which did not “linearly” correlate well with others,rest all the features proved to be extremely useful for our classification task.

4.2.7 Classification

Various traditional techniques of supervised machine learning have been employed in many applications requiring classification of data.The challenge while dealing with any problem is the selection of a particular classification model that would give the best results without causing any overfitting.Hence we used a number of different models and compare them on various metrics to select the best suitable model.

The classification models were applied in the following stages:

4.2.7.1 Classification using only textual features and compared all three vectorizers used

4.2.7.2 Classification using only the social context features.

4.2.7.3 Classification using “Hybrid” features.

The above results are compared on the basis of their accuracy scores,F1 scores and confusion matrix.Multiple algorithms have been used for classification.

Naive Bayes Classifier:Bayesian classifiers are statistical classifiers. They can predict class membership probabilities such as the probability that a given tuple belongs to a particular class.Bayesian classification is based on Bayes’ theorem.Bayesian classifiers have also exhibited high accuracy and speed when applied to large databases.Naive Bayesian classifiers assume that the effect of an attribute value on a given class is independent of the values of the other attributes. This assumption is called “class conditional independence”.

Ensemble methods:Bagging, boosting, and random forests are examples of ensemble methods.An ensemble combines a series of k learned models (or base classifiers),M1, M2,....,Mk, with the aim of creating an improved composite classification model, M*. A given data set,D,is used to create k training sets, D1, D2,...,Dk, where Di (1 <= i<= k-1) is used to generate classifier Mi . Given a new data tuple to classify, the base classifiers each vote by returning a class prediction. The ensemble returns a class prediction based on the votes of the base classifiers.

An ensemble tends to be more accurate than its base classifiers.The base classifiers may make mistakes, but the ensemble will misclassify X only if over half of the base classifiers are in error. Ensembles yield better results when there is significant diversity among the models. That is, ideally, there is little correlation among classifiers. The classifiers should also perform better than random guessing. Each base classifier can be allocated to a different CPU and so ensemble methods are parallelizable.

Random Forest:It creates a set of decision trees from randomly selected subset of training set. It then aggregates the votes from different decision trees to decide the final class of the test object.In a random forest,each of the classifiers in the ensemble is a decision tree classifier so that the collection of classifiers is a “forest.”The individual decision trees are generated using a random selection of attributes at each node to determine the split.

The generalization error for a forest converges as long as the number of trees in the forest is large. Thus, overfitting is not a problem.The accuracy of a random forest depends on the strength of the individual classifiers and a measure of the dependence between them. Because random forests consider many fewer attributes for each split, they are efficient on very large databases.

XGboost:Its name stands for eXtreme Gradient Boosting.XGBoost is a scalable and accurate implementation of gradient boosting machines and it has proven to push the limits of computing power for boosted trees algorithms. Gradient boosting also comprises an ensemble method that sequentially adds predictors and corrects previous models.This is an ensemble method that seeks to create a strong classifier (model) based on “weak” classifiers. In this context, weak and strong refer to a measure of how correlated are the learners to the actual target variable. By adding models on top of each other iteratively, the errors of the previous model are corrected by the next predictor, until the training data is accurately predicted or reproduced by the model.

Passive-Aggressive Classifier:Passive Aggressive Algorithms are a family of online learning algorithms (for both classification and regression).The idea is very simple and their performance has been proved to be superior to many other alternative methods.They are a family of algorithms for large-scale learning.

5. Testing

5.1 Testing Plan

Cross-validation is a common method to evaluate the performance of a text classifier. It consists in splitting the training dataset randomly into equal-length sets of examples (e.g. 4 sets with 25% of the data). For each set, a text classifier is trained with the remaining samples (e.g. 75% of the samples). Next, the classifiers make predictions on their respective sets and the results are compared against the human-annotated tags. This allows finding when a prediction was right (true positives and true negatives) and when it made a mistake (false positives, false negatives).

5.2 Testing Methods

With these results,we can build performance metrics that are useful for a quick assessment on how well a classifier works:

5.2.1 Accuracy: the percentage of texts that were predicted with the correct tag.

5.2.2 Precision: the percentage of examples the classifier got right out of the total number of examples that it predicted for a given tag.

5.2.3 Recall: the percentage of examples the classifier predicted for a given tag out of the total number of examples it should have predicted for that given tag.

5.2.4 F1 Score: the harmonic mean of precision and recall.

5.3 Limitations of the Solution

5.4.1 News Source features like political bias, credibility and source trustworthiness could have been helpful in improving the accuracy of results.
5.4.2 News propagation path features can be used for early detection of fake news.

6. Findings, Conclusion and Future Work

6.1 Findings

	F1 Score			Accuracy		
Model Used	Text	Social Context	Hybrid	Text	Social Context	Hybrid
Naive-Bayes-count-vectorizer	0.719	–	–	0.721	–	–
Naive-Bayes-tf-idf-vectorizer	0.728	–	–	0.732	–	–
Naive-Bayes		0.509	0.691		0.539	0.703

Table 1: Naive Bayes Classifier

	F1 Score			Accuracy		
Model Used	Text	Social Context	Hybrid	Text	Social Context	Hybrid
Random Forest-count-Vectorizer	0.760	–	–	0.761	–	–
Random Forest-tf-idf-Vectorizer	0.750	–	–	0.750	–	–
Random Forest-Word2Vec	0.546	–	–	0.547	–	–
Random Forest		0.787	0.847		0.788	0.847

Table 2: Ensemble: Random Forest

	F1 Score			Accuracy		
Model Used	Text	Social Context	Hybrid	Text	Social Context	Hybrid
Xgboost-count_vectorizer	0.734	-	-	0.736	-	-
Xgboost-tf-idf-vectorizer	0.725	-	-	0.727	-	-
Xgboost-Word2Vec	0.501	-	-	0.555	-	-
Xgboost		0.804	0.855		0.805	0.855

Table 3:Ensemble: XgBoost

	F1 Score			Accuracy		
Model Used	Text	Social Context	Hybrid	Text	Social Context	Hybrid
Passive Aggressive-count-vectorizer	0.747	-	-	0.747	-	-
Passive Aggressive-tf-idf-vectorizer	0.764	-	-	0.765	-	-
Passive Aggressive-Word2vec	0.335	-	-	0.505	-	-
PassiveAggressive		0.531	0.800		0.573	0.800

Table 4: Passive Aggressive Classifier

6.2 Conclusions

In our project, we analyzed various categories of classification algorithms under a number of situations and from the findings we conclude that

1) Text features, when used along with all the relevant social context features to form a set of hybrid features, prove to be best suitable for fake news classification problem
2) Ensemble learning classification algorithms like Xgboost and Random Forest show the best accuracies and are best suitable when used with hybrid features.

6.3 Future Work

Even though the problem of fake news detection has been solved, we plan to overcome the limitations of the solution by considering news source and news propagation path features.This will not only help improve the accuracy of the results, but also will make the model more generalizable and robust in early detection of fake news since it only relies on common user characteristics which are more available, reliable and robust in the early stage of news propagation than linguistic and structural features.

Also,in the future, we plan to investigate whether user characteristics can help us identifying users who are easy to believe and spread fake news, and which features affect users' tendency to spread fake news most significantly, which are important problems in the prevention and debunking of fake news.

References:

1. Lazer, David & Baum, Matthew & Benkler, Yochai & Berinsky, Adam & Greenhill, Kelly & Menczer, Filippo & Metzger, Miriam & Nyhan, Brendan & Pennycook, Gordon & Rothschild, David & Schudson, Michael & Sloman, Steven & Sunstein, C. & Thorson, Emily & Watts, Duncan & Zittrain, Jonathan. (2018). The science of fake news. Science. 359. 1094-1096. 10.1126/science.aao2998.
2. H. Allcott, M. Gentzkow, J. Econ. Perspect. 31, 211 (2017)
3. Gottfried, E. Shearer, News use across socialmedia platforms 2017, Pew Research Center, 7September 2017; www.journalism.org/2017/09/07/news-use-across-social-media-platforms-2017/.
4. Senate Judiciary Committee, Extremist contentand Russian disinformation online: Working with tech to find solutions (Committee on the Judiciary,2017); www.judiciary.senate.gov/meetings/extremist-content-and-russian-disinformation-onlineworking-with-tech-to-find-solutions.
5. Rubin, V., Chen, Y. and Conroy, N. (2015). Deception detection for news: Three types of fakes. Proceedings of the Association for Information Science and Technology, 52(1), pp.1-4.
6. Shu, K., Sliva, A., Wang, S., Tang, J. and Liu, H. (2017). Fake News Detection on Social Media. ACM SIGKDD Explorations Newsletter, 19(1), pp.22-36.
7. Wang, William Yang. “"Liar, Liar Pants on Fire": A New Benchmark Dataset for Fake News Detection.” ACL (2017).
8. Mitra, Tanushree and Eric Gilbert. “CREDBANK: A Large-Scale Social Media Corpus With Associated Credibility Annotations.” ICWSM (2015).
9. Santia, Giovanni C. and Jake Ryland Williams. “BuzzFace: A News Veracity Dataset with Facebook User Commentary and Egos.” ICWSM (2018).
10. J. C. S. Reis, A. Correia, F. Murai, A. Veloso and F. Benevenuto, "Supervised Learning for Fake News Detection," in *IEEE Intelligent Systems*, vol. 34, no. 2, pp. 76-81, March-April 2019.