An Investigation of Fake News Classification via Hybrid Feature Selection

by Olumide Omololu, B.Eng.

Submitted to The University of Nottingham
September 2021
in partial fulfilment of the conditions for the award of the degree of
Master of Science in Computer Science

I declare that this dissertation is all my own work, except as indicated in the text

Contents

List of Tables

List of Figures

Abstract

There are several ways to identify fake news on social media. However, The approaches investigated tend to be elaborate, complex and resource intensive. In this Thesis we implement a machine learning pipeline using classical methods such as logistic regression and naive Bayes on a dataset consisting of both text and propagation path data to evaluate the trade-offs of using a hybrid approach in comparison to a purely textual method of fake news detection.

Keywords — Natural Language Processing, Fake News, Machine Learning, Graph Representation

Acknowledgements

I would like to thank my supervisor, Dr Jeremie Clos, for providing guidance and feedback throughout this project. I would also like to thank my family, for supporting me during the execution of this project.

Chapter 1

Introduction

1.1 Background

The advent of social media platforms, such as the well-known microblogging platform, Twitter, has been a transformative force in the domain of both large scale and personal communication. However, like many transformative technologies, it is not without pitfalls. Specifically, false information which frequently tends to be substantially polarizing can be propagated on social media. This false information or **fake news** is particularly problematic as it is frequently centred around persons and affairs of current interest as observed in (Allcott & Gentzkow, 2017).

A new and unique challenge introduced by social media is the use of automated accounts, also known as **bots**, which serve as a means of delivering false and polarizing information on a large scale. The propagation of fake news is also greatly assisted by the **echo chamber** effect, which is a result of the incremental customization of social media presented to users by most social media platforms based on the user's preferences in order to drive engagement. A variety of approaches have been taken in order to identify the fake news and its spread on social media using a variety of models and features. The chosen approaches, despite their accuracy, are usually complex and resource intensive due to their reliance on computationally expensive deep learning methods.

1.2 Objectives

The objective of this study is the implementation and evaluation of a fake news classification pipeline which utilizes classical natural language processing and machine learning methods in tandem with fake news propagation data and subsequently assess its effect on accuracy

in comparison to a textual approach. The utilization of **shallow** learning methods, such as linear and probability-based classifiers, introduces an additional requirement of feature selection and preprocessing techniques.

Chapter 2

Literature Review

This literature review of this study consists of two sections. The first section gives an overview of the conceptual principles and performance of previously implemented fake news classification projects as presented in the cited research papers. The latter half discusses previous attempts at classifying fake news using the FakeNewsNet dataset while comparing key insights obtained from these studies and identifying additional research objectives.

2.1 Fake News Classification Approaches

An extensive survey of fake news detection methods by (Zhou & Zafarani, 2018) concluded in the classification of fake news detection strategies into four categories. The survey insists that fake news can be identified by its content, writing style, propagation path and credibility. However, In this review, we are primarily concerned with carrying out a comparative study of content and propagation based approaches.

2.1.1 Content-Based Approaches

A study by (Esmaeilzadeh, Peh, & Xu, 2019) established that the use of neural text summarization methods as a feature generation process can increase the accuracy of fake news classification models. (Yang, Yang, Niven, & Kao, 2019) were able to achieve a similar objective using ensemble of Natural language inference models in tandem with the leading language representation model originally developed by (jacob devlin, mingwei chang, kenton lee, & kristina toutanova, 2018) on dataset based on Chinese news sources.
An alternative approach taken by (X. Zhang et al., 2020) explored the use of extracting features from the relationship signals between publisher and viewer (i.e. posts and their

associated comments) sentiment on social media. The approach taken in this study also resulted an increased level of accuracy in fake news classification when compared to conventional text classification methods after testing on English and Chinese datasets.

2.1.2 Propagation-Based Approaches

Propagation-based approaches at fake news classification are language agnostic compared to content-based fake news classifiers (Monti, Frasca, Eynard, Mannion, & Bronstein, 2019). They are more practical for multilingual datasets as a result.

(Tacchini et al., 2017) carried out a study of hoax detection on a dataset of users and posts from the popular social media platform, Facebook, using interaction metrics (i.e. likes) as training data for a logistic regression model. This approach proved to be accurate, robust and scalable.

(Zhao et al., 2018) were able to establish that, in its early stages, fake news propagates in a manner that is notably distinct from real news after an extensive review of key dispersal metrics such as characteristic distance, heterogeneity and layer ratio. (Liu, Liu, Wu, & Wu, 2018) proposed a model of early fake news detection which represented propagation paths as multivariate time-series which were classified using an assortment of deep learning methods. This approach permitted the identification of fake news within five minutes of propagation at an accuracy of 85% on data obtained from the Twitter social network

(Monti et al., 2019) presented a novel approach at fake news classification which utilized geometric deep learning. This approach resulted in a robust and accurate classifier which relies on user and propagation based features.

2.1.3 Combined Approaches

(Ruchansky, Seo, & Liu, 2017) proposed a model which classifies news based on social media based on temporal user activity, background user behaviour and news content. However, the model performs these classifications in a modular fashion and relies on a combined score of the three selected features to classify news permitting more robust and accurate classification.

2.2 Approaches Utilized on FakeNewsNet and Related Datasets

The FakeNewsNet data repository was created by (Shu et al., 2018) for the purpose of facilitating information credibility research. It consists of two datasets which are both made up of linguistic and temporal dispersion data about posts on the social media platform, Twitter, from well known fact checking websites.

In (Shu, Mahudeswaran, Wang, & Liu, 2019), the authors were able to establish the fact that temporal features are more discriminative than linguistic and structural features in the course of their investigation of fake news classification methods via hierarchical propagation networks. (Chandra et al., 2020) presented a novel fake news detection framework which utilizes a graph neural network based classifier to identify fake news by factoring in the online demographic of users who interact with it in addition to its content and dispersal path.

The cited works provide a deep, comprehensive and novel investigations of deep learning based attempts at fake news classification. However, this study aims to highlight the impact of fake news propagation information on fake news classification when used in combination with a conventional and less computationally expensive text mining technique, such as the use of term-document matrices.

Chapter 3

Methodology

3.1 Data Extraction and Transformation

The approach taken in this study is primarily concerned with the utilization of two kinds of features. These are graph measures, which represent the propagation of the news data across a selected social media platform, and word vectors, which represent the semantic content of the news itself. The fake news classification approach taken in this study will consist of data extraction from the two different sources, transformation and reduction of the chosen data and finally, classification using the selected classifiers as illustrated in Figure 3.1.

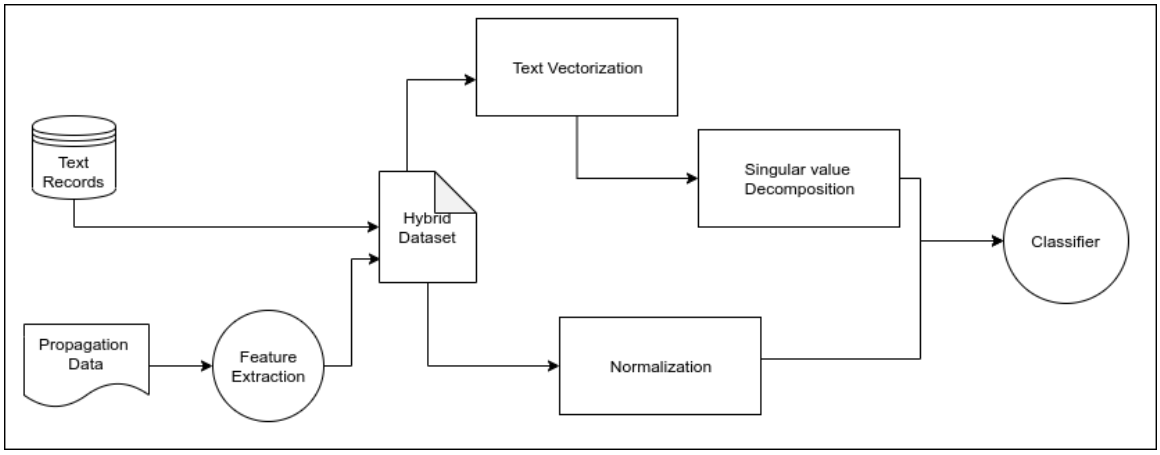

Figure 3.1: Fake News Classification Pipeline

3.1.1 Graph Metrics

News propagation paths in social networks are represented via the use of directed graphs. The graphs are characterized by the strict usage of directed edges as seen in Figure 3.2 and Figure 3.3. These graphs can also be regarded as trees given their directed and acyclic nature. (Vicario et al., 2016) were able to establish notable differences in diffusion patterns of factual and fabricated online content.

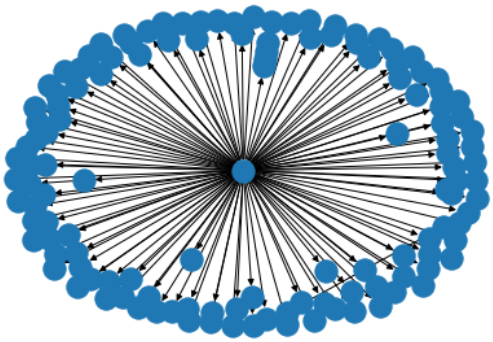

FIGURE 3.2: Propagation Path Sample 1

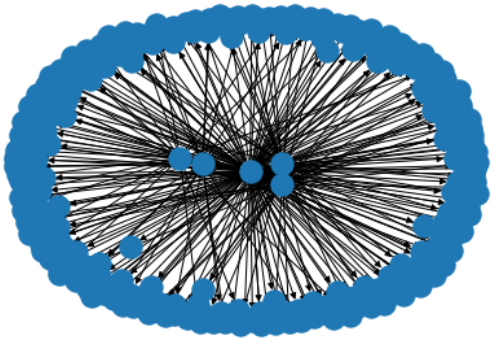

FIGURE 3.3: Propagation Path Sample 2

Centrality

An important graph metric which will be used in this study is the degree centrality of the root node. It is a standardized measure of the total amount of direct links to other nodes

in the network (J. Zhang & Luo, 2017). It can also be thought of as the root node's level of influence or popularity.

Degree

Another graph measure utilized in this study is the degree of the root node. This can simply be defined as the number of edges adjacent to the root node.

Density

The density of the news propagation graph is also used as a measure in this study. It can be defined as the ratio of the number of edges in a graph to the maximum number of possible edges in a graph.

3.1.2 Natural Language Processing Methods

Text Vectorization

Text vectorization is the process of generating vector representations for linguistic data. A simple approach utilized in this study is creating a term-document matrix from the collection of text data available and representing each item within it as a vector consisting of the counts of each word within it as represented by the aforementioned matrix. This is also known as bag of words representation.

Term Weighting

In order to facilitate more accurate representations of document vectors in relation to a corpus, term frequency — inverse document frequency (TF-IDF) weighting is applied. It is able to ensure that text vectors emphasize the most relevant words within the linguistic data they represent by determining the relative frequency of words in a specific text item compared to the inverse proportion of that word over the entire corpus (Ramos, 1999).

3.1.3 Data Transformation and Reduction Techniques

Normalization

Feature scaling is a process by which the values of data are rescaled to values between a certain range. This may be required in order to facilitate or enhance the application of sev-

eral machine learning algorithms. In this study, a variant known as min-max normalization was utilized to place certain input features within a range of one and zero.

Singular Value Decomposition

Singular value decomposition (SVD) is a dimensionality reduction technique that decomposes a matrix into three component matrices. These matrices are the left and right singular vectors and the singular value matrix which is a diagonal matrix. In this study, we utilize a variant of this method known as truncated singular value decomposition, which only generates the number of singular values specified by the user as seen in equation (3.1). In which an approximation of the original matrix is generated using the first k elements of the decomposed components.

$$X \approx X_k = U_k \Sigma_k V_k^\top \tag{3.1}$$

The technique is also referred to as latent semantic analysis when used in a natural language processing context. Its popularity in text processing compared to other dimensionality reduction techniques is a result of the fact that it does not require data centring, which is advantageous when working with sparse matrices such as term-document matrices(Manning, Raghavan, & Schütze, 2008).

3.2 Machine Learning Techniques

Machine learning refers to a set of computational techniques that permit computers to learn from data for a variety of purposes such as identifying patterns and making decisions (aurelien geron, 2017).

3.2.1 Classification Algorithms

Classification is a process by which a machine learning algorithm learns to map a set of input features to a label based on observed relationships between input features and the labels within its training data. The classification algorithms under the review of this study include:

Logistic Regression

Logistic regression is a regression algorithm that is applicable to binary classification problems as it estimates probabilities based on the input features of an item and assigns it to either of the two classes or labels used to train it using sigmoid function as seen in which assigns a value of 1 or 0 to the input data as seen in equation 3.2. The binary values are used to represent the two classes used in logistic regression.

$$\sigma(z) = \frac{1}{1 + e^{-z}} \tag{3.2}$$

K Nearest Neighbours

The K Nearest Neighbours algorithm classifies unknown items based on the labels of the nearest items when represented in vector space. The distance metric used to estimate the proximity and number of neighbouring items to be considered, specified by (k), are the key defining parameters of this model.

Support Vector Classifier

Support Vector Machines Serve as a classification method via the establishment of decision boundaries with wide margins in vector space based on clustering patterns of a subset of training instances. Unlabelled instances of data are classified based on their location in relation to the defined decision boundaries. The support vector machine is noted to be effective in classifying high dimensional data.

Naive Bayes Classifier

The Naive Bayes classifier classifies data based on probability distributions of features with the assumption that they are independent. It is an application of Bayesian statistics to machine learning. This study utilizes a variation of the Naive Bayes classifier known as multinomial naive Bayes which is ideal for data with discrete features, such as text vectors.

Random Forrest Classifier

The random forest classifier is an ensemble method that utilizes a collection of decision trees to make decisions collectively in order to greatly increase the accuracy of the baseline method. The approach of using the average outcome of the individual decision to make a

decision is referred to as bagging. This method is preferred over individual decision trees as the bagging process offsets the variance the results from the use of a single tree.

3.2.2 Hyperparameter Tuning

In order to identify the best performing version of a classification model, we relied on hyperparameter tuning. Hyperparameter tuning is a process whereby a model is trained and assessed using various configurations of its defining parameters to establish which configuration results in the highest accuracy. This study specifically relied on cross-validation and a Bayesian parameter search process to achieve the best results.

Cross Validation

Cross-validation is a means of screening a machine learning model for overfitting by partitioning the training data into blocks which are iteratively used as test data while the rest of the data is used to train the classifier. Each iteration is referred to as a fold. The performance metric of the cross-validation process is the average accuracy of all the iterations.

Bayesian Hyperparameter Optimization

Hyperparameters are configurable attributes of machine learning models which are not learned via training and require an optimization method to tune. Bayesian Hyperparameter Optimization is a means of efficient hyperparameter tuning which aims to reduce tuning time selecting parameters based on specified distributions. It uses past evaluation results to form a probabilistic model by mapping Hyperparameters to a probability score on an objective function. This approach has been proven to be effective on a variety of models as shown in (Snoek, Larochelle, & Adams, 2012).

3.2.3 Model Evaluation Methods

In this study, the performance of our models will be evaluated using the following metrics. These performance measures are ratio of true positives(TP), true negatives(TN), false positives (FP) and false negatives (FN).

Accuracy

This is the ratio of correct classifications to the total number classifications. While this is a commonly used metric it is not an effective measure of a machine learning models performance on imbalanced datasets.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.3}$$

Precision and Recall

Precision is the proportion of valid positive classifications to the total number of positive classifications while recall is the proportion of valid positive classifications to the total number of positive instances.

$$Precision = \frac{TP}{TP + FP} \tag{3.4}$$

$$Recall = \frac{TP}{TP + FN} \tag{3.5}$$

F1 Score

The F1 score is the harmonic mean of the classifier's precision and recall. This metric is required due to the fact that precision and recall have an inverse relationship and cannot individually describe the classifier's performance.

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} = \frac{2 * TP}{2 * TP + FP + FN} \tag{3.6}$$

Confusion Matrix

The Confusion matrix is a visual representation of the amount of accurately classified and falsely classified instances.

3.2.4 Statistical Testing

In order to establish significant differences between the performance of the approach proposed in this thesis and classifier reliant solely on text features, a null Hypothesis test is required. A paired t-test will be relied on for this purpose as it can adequately assess for differences in predictive ability.

Chapter 4

Comparative Evaluation of Classifiers

4.1 Data Processing and Analysis

4.1.1 Procurement of FakeNewsNet Data

In order to execute the approach prosed in the Methodology we obtained the FakeNewsNet dataset from a publicly available repository. The dataset takes the form of a collection of CSV files which hold the titles of news posts made on Twitter and the JSON files detail the propagation path of each tweet as it is shared among users.

Extraction of Graph Features

The extraction of definitive metrics from the graph data was facilitated via the use of the **Networkx** library. The following features were selected:

- degree
- density
- centrality
- number of nodes
- number of edges

Correlation Check

An initial correlation check of the selected features revealed a high degree of correlation between certain features as seen in Figure 4.1.

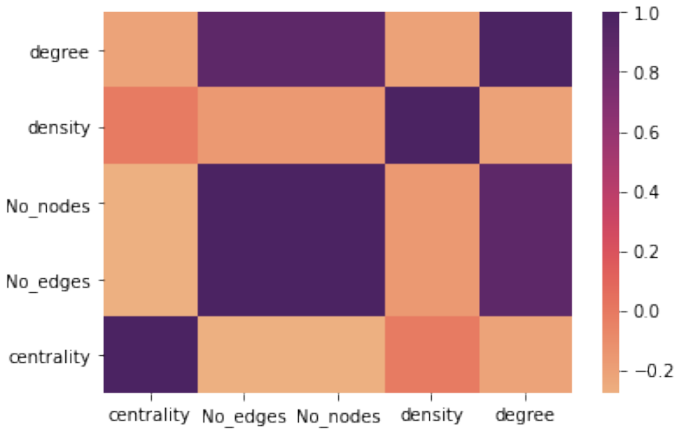

FIGURE 4.1: Correlation Check 1

Feature Selection

The selected features where then reduced to those which showed a significant degree of independence as seen in Figure 4.2.

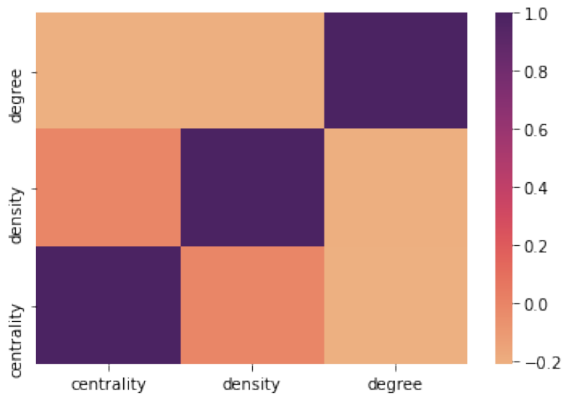

FIGURE 4.2: Correlation Check 2

Feature Statistics

The descriptive statistics of the selected features can be observed in Table 4.1. Additionally, Figures 4.3, 4.4 and 4.5 highlight the difference in the graph dispersal metrics between real and fake news.

TABLE 4.1: Descriptive Statistics of Graph Dataset

	density	degree	centrality
count	11259	11259	11259
mean	0.030916	125.387779	0.757562
std	0.045908	337.072094	0.224301
min	0.000025	1.000000	0.013986
max	0.333333	17520.000000	0.995833

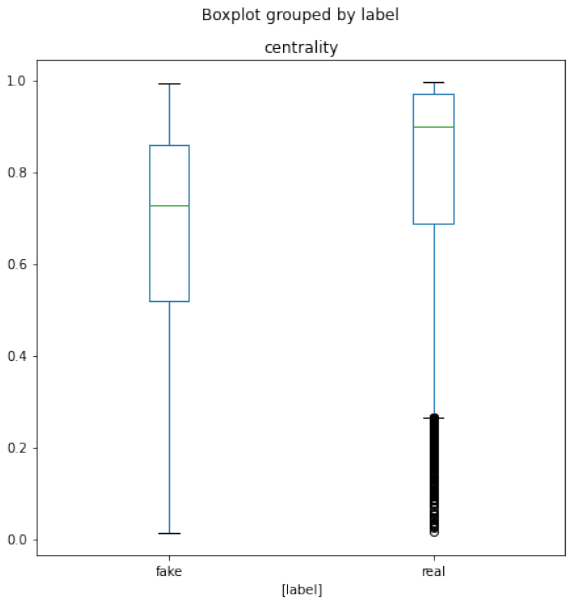

FIGURE 4.3: Centrality Boxplot

4.1.2 Data Transformation and Reduction

After extracting the graph dispersal features into a tabular data format, we joined it with the corresponding tweets which were found in the CSV files. A notable observation in the dataset was the disparity of propagation and text data. This is highlighted by the charts in Figure 4.6 and Figure 4.7 which showcase the distribution of labels in the text and propagation datasets respectively. This dataset was subsequently split into training and test sets consisting of 80% and 20% of the original dataset respectively with stratification in order to mitigate the effects of the imbalance within the dataset.

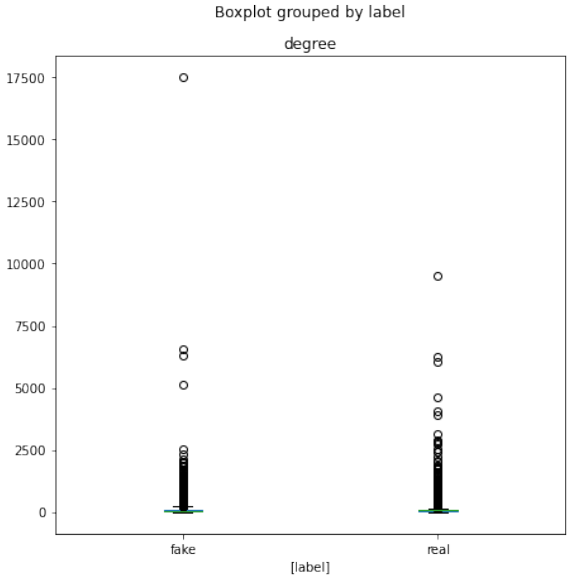

FIGURE 4.4: Degree Boxplot

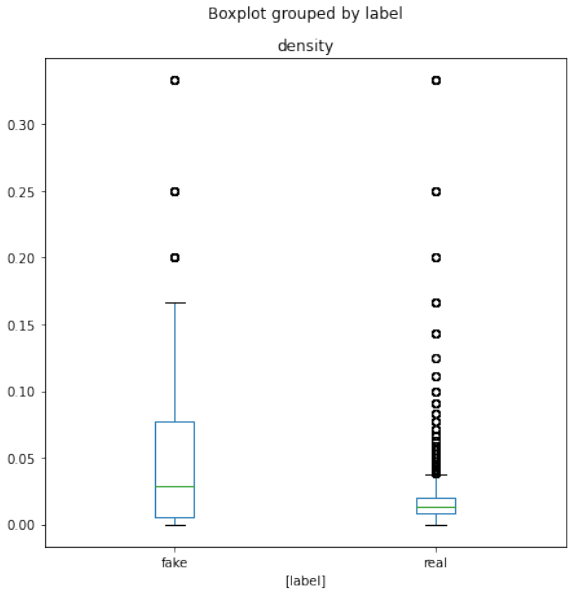

FIGURE 4.5: Density Boxplot

Final Dataset

The final dataset consisted of 11,259 instances of items with both textual and propagation data.

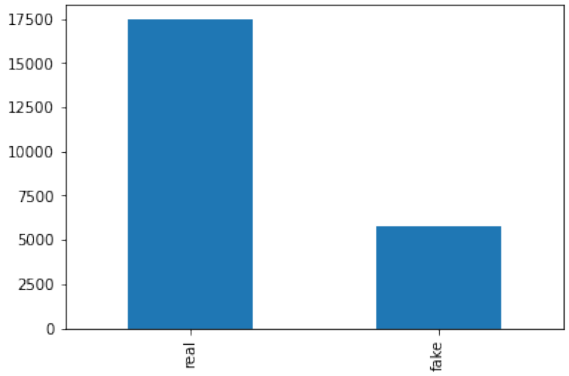

FIGURE 4.6: Label distribution in Textual data

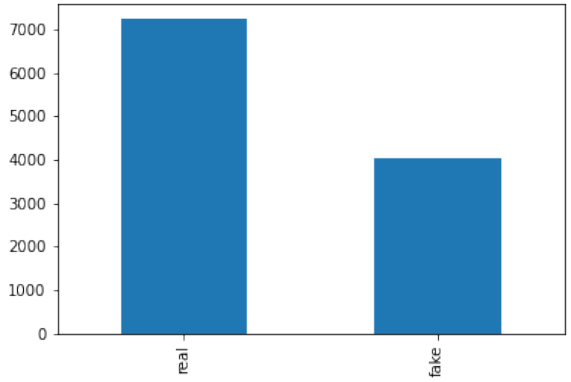

FIGURE 4.7: Label distribution in Propagation Data

Data Transformation and Reduction

In order to create suitable features for the classifiers. We applied normalization to the propagation features and also vectorized the textual data via usage of term-document matrix which was subsequently weighted using TF-IDF. We refrained from relying on stop-word removal due to the unpredictable nature of the textual data and lack of a context detection method. This resulted in a large term-document matrix of 11501 words. The increased dimensionality would be detrimental on the performance of the machine learning models except for the Naive Bayes classifier. We applied truncated singular value decomposition in order to represent the document vector in 100 dimensions for all models but the Naive Bayes. We also created a version of the dataset using only the test features to train the baseline classifiers.

4.2 Evaluation of Classifier Performance

We utilized Bayesian hyperparameter optimization on each model in tandem with five-fold cross validation on each configuration in order to establish optimal model performance on both baseline and hybrid classification approaches. This approach works to identify the optimal parameters for the model by evaluating all attempted combinations of hyperparameter values on the model with cross-validation and returning the classifier with the best performing combinations of hyperparameters.

4.2.1 Hyperparameter Selection

The parameters selected to be tuned by the chosen models for both the text based, and hybrid classification are detailed in Tables 4.2, 4.3, 4.4, 4.5 and 4.6.

Logistic Regression

TABLE 4.2: Logistic Regression Hyperparameters

Logistic Regression Hyperparameters		Range
Solver	The optimization algorithm used.	liblinear, saga
Penalty	The regularization method used.	l1,l2
tol	The tolerance for the stopping criteria.	1e-4 — 1e-3
C	The inverse of regularization strength.	1-100

K Nearest Neighbours

TABLE 4.3: K Nearest Neighbours Hyperparameters

KNN Hyperparameters		Range
n_neighbors	The number of neighbours used.	1 – 21
weights	The weight function used.	uniform, distance
metric	The distance metric used.	euclidean, Manhattan, Minkowski

Support Vector Classifier

TABLE 4.4: Support Vector Classifier Hyperparameters

SVM Hyperparameters		Range
Gamma	The kernel coefficient used.	1e-6 — 100
Degree	The degree of the "poly" kernel.	1 – 5
kernel	The kernel type used.	linear, poly, rbf, sigmoid
C	The inverse of regularization strength.	1e-6 — 100

Naive Bayes Classifier

TABLE 4.5: Naive Bayes Classifier Hyperparameters

Naive Bayes Hyperparameters		Range
Alpha	The additive smoothing parameter.	0 – 1
fit_prior	The option to learn prior probabilities.	True, False

Random Forrest Classifier

TABLE 4.6: Random Forrest Classifier Hyperparameters

Logistic Regression Hyperparameters		Range
Bootstrap	The use of bootstrap samples.	True, False
n_estimators	The number of trees in the forest.	10,50,100,500,1000
max_features	The number of features to considered for the best split.	sqrt, log2, auto

4.2.2 Performance of Models Using Text

The performance of the baseline text based classification method using the optimized models resulted in the scores in Table 4.7 and the confusion matrices that follow when tested using the test dataset.

Logistic Regression

The hyperparameter tuning process revealed that the text based logistic regression model exhibits the highest level of accuracy using the liblinear solver, L2 regularization, an inverse regularization strength of 11 and a stopping criteria tolerance of 0.001.

TABLE 4.7: Performance of Text-Based Classifiers

	Logistic Regression	Naive Bayes	KNN	SVM	Random Forest
CV Accuracy	0.7639	0.7829	0.7546	0.7642	0.7725
Accuracy	0.7447	0.7615	0.7376	0.7411	0.7549
F1 Score	0.8155	0.8340	0.8163	0.8163	0.8231
Precision	0.7602	0.7536	0.7410	0.7494	0.7666
Recall	0.8796	0.9336	0.9087	0.8962	0.8886

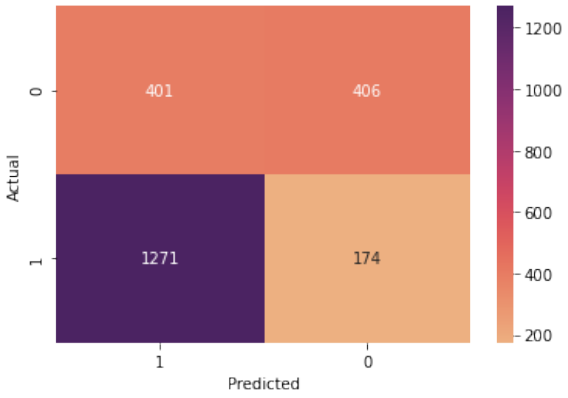

FIGURE 4.8: Text Based Logistic Regression Classifier

KNN

The tuning process revealed that the text based KNN model exhibits the highest level of accuracy using the Minkowski distance metric, 21 neighbours which are uniformly weighted.

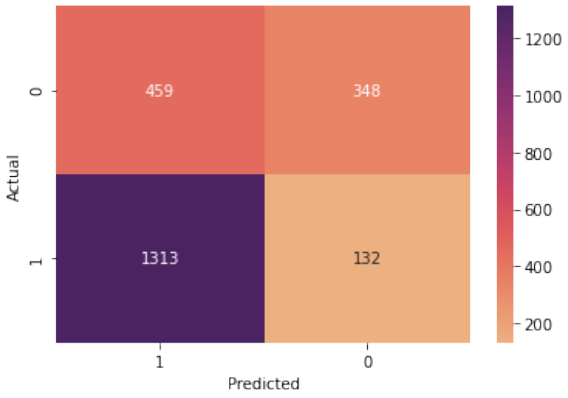

FIGURE 4.9: Text Based K Nearest Neighbours Classifier

SVM

The tuning process revealed that the text based SVM model exhibits the highest level of accuracy using the linear kernel with a kernel coefficient of 100 and an inverse regularization strength of 100 as well.

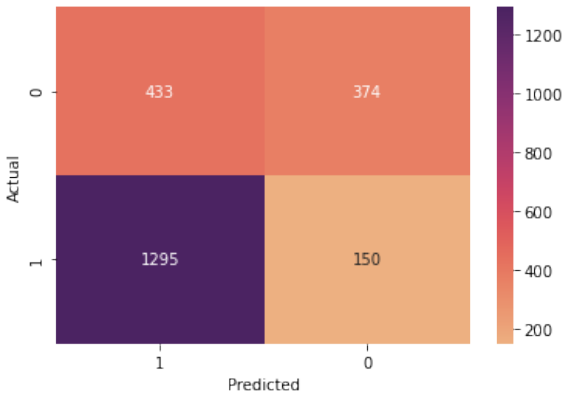

FIGURE 4.10: Text Based Support Vector Classifier

Naive Bayes

The tuning process revealed that the text based Naive Bayes model exhibits the highest level of accuracy using an additive smoothing parameter of 1 while learning from prior class probabilities.

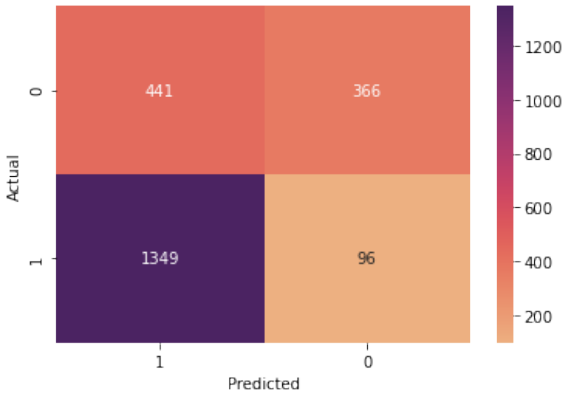

FIGURE 4.11: Text Based Naive Bayes Classifier

Random Forest

The tuning process revealed That the text based random forest model exhibits the highest level of accuracy using 1000 trees.

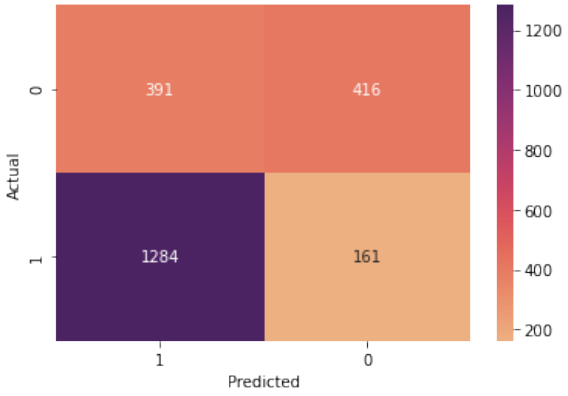

FIGURE 4.12: Text Based Random Forest Classifier

4.2.3 Performance of Models Using Combined Features

The performance of the hybrid classification method using the optimized models resulted in the scores in Table 4.8 and the confusion matrices that follow when tested using the test dataset.

TABLE 4.8: Performance of Hybrid Classifiers

	Logistic Regression	Naive Bayes	KNN	SVM	Random Forest
CV Accuracy	0.8178	0.8036	0.8037	0.8379	0.8475
Accuracy	0.7993	0.7851	0.7882	0.8193	0.8393
F1 Score	0.8515	0.8373	0.8458	0.8656	0.8782
Precision	0.8105	0.8143	0.7937	0.8277	0.8546
Recall	0.8969	0.8616	0.9052	0.9073	0.9031

Logistic Regression

The tuning process revealed that this model exhibits the highest level of accuracy using the liblinear solver, L1 regularization, an inverse regularization strength of 68 and a stopping criteria tolerance of 0.001.

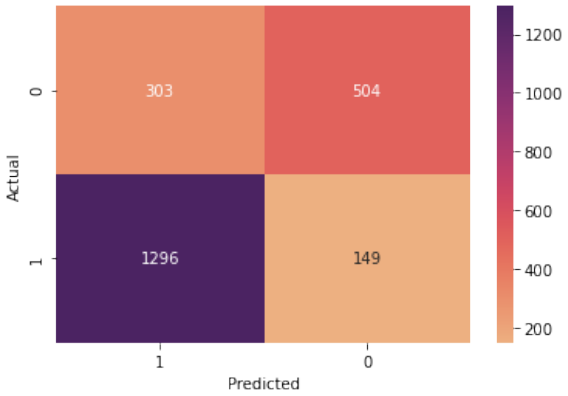

FIGURE 4.13: Logistic Regression Classifier

KNN

The tuning process revealed that this model exhibits the highest level of accuracy using the Minkowski distance metric, 8 neighbours which are weighted by the inverse of their distance.

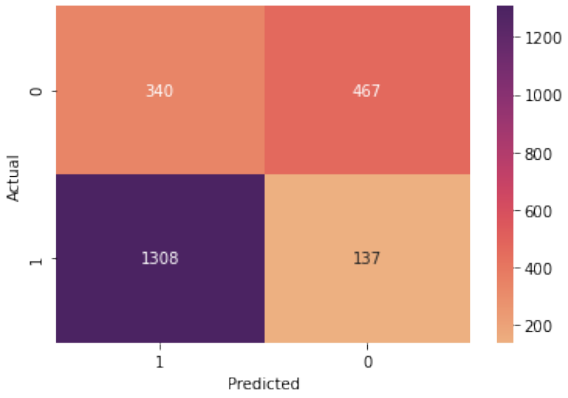

FIGURE 4.14: K Nearest Neighbours Classifier

SVM

The tuning process revealed that this model exhibits the highest level of accuracy using the RBF kernel with a kernel coefficient of 0.061 and an inverse regularization strength of 100.

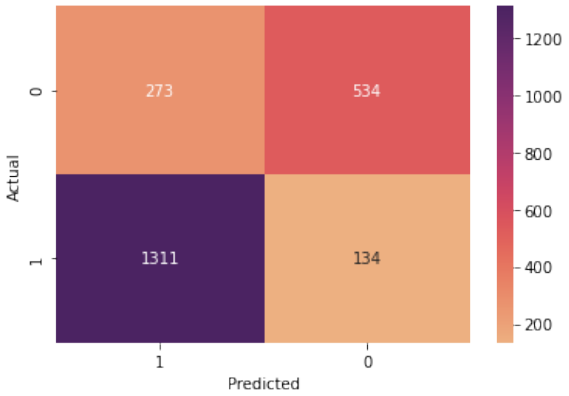

FIGURE 4.15: Support Vector Classifier

Naive Bayes

The tuning process revealed that this model exhibits the highest level of accuracy using an additive smoothing parameter of 1 while ignoring prior class probabilities.

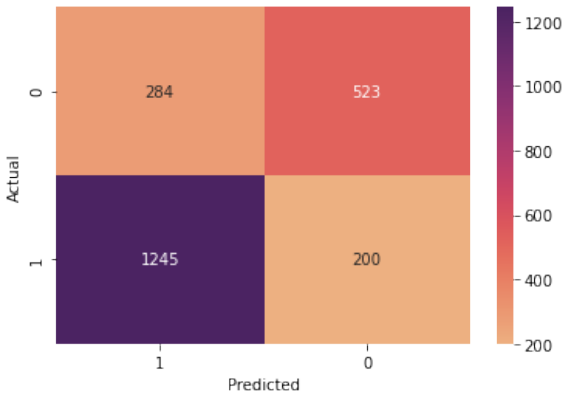

FIGURE 4.16: Naive Bayes Classifier

Random Forest

The tuning process revealed that this model exhibits the highest level of accuracy using 1000 trees as well.

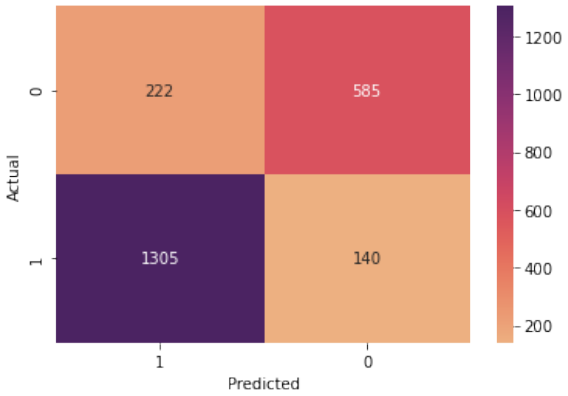

FIGURE 4.17: Random Forest Classifier

4.3 Comparison and Hypothesis Testing of Classifiers

From the highlighted results above, it can be observed that the hybrid classification Pipeline performs better both in training and on the test data set. A notable observation is the superior recall exhibited by all models compared to precision. This implies that the classifiers will generally have a lower level of false negatives compared to false positives making the pipeline more suited to a fake news detection task instead as opposed to use as a fake news filter. An important observation is the reduced impact of the propagation features on the Naive Bayes classifier. This can be attributed to the reliance of that model on the unreduced bag of words representation which occupies substantially more vector space thereby reducing the impact of Propagation features.

In order to establish that the observed difference between the accuracies of the hybrid and text-based classifiers is not the result of statistical noise within the test dataset. The goal of a null hypothesis statistical test is to reject or accept a null hypothesis that postulates that there are no differences between the distributions of a contrasted pair of data samples. This decision is based on the P-value provided by the chosen statistical test. The P-value has a threshold of 0.05 which acts as a decision boundary(i.e. the P-value of the test should be less than the threshold in order to reject the null hypothesis). The predictions of both pipelines serve as a paired observation which can be tested using the paired T-test (Japkowicz & Shah, 2011). The paired T-test is a parametric test that can be used to establish if the difference between two means is meaningful. It is adapted from the more general Welch's T-test. The P-values for each pair of classifiers as highlighted in Table 4.9 are all less than the threshold of 0.05 thereby confirming the proposed accuracy trade-off.

4.3.1 Paired T Test

TABLE 4.9: Comparison and Paired T-test of Classifiers.

Classifier	Accuracy		F1-Score		P-Value
	Hybrid	Text	Hybrid	Text	
Logistic Regression	0.7993	0.7447	0.8515	0.8155	9.59e-06
Naive Bayes	0.7851	0.7615	0.8373	0.8340	3.03e-62
KNN	0.7882	0.7376	0.8458	0.8163	2.43e-11
SVM	0.8193	0.7411	0.8656	0.8163	7.95e-15
Random Forest	0.8393	0.7549	0.8782	0.8231	6.37e-14

Chapter 5

Conclusion and Future Work

5.1 Key Insights

In this study, We were able to illustrate the impact of social media propagation features on fake news classification using classical natural language processing and machine learning techniques. We were also able to establish that the increased accuracy is independent of the machine learning technique utilized thereby confirming the robustness and flexibility of the proposed approach. It should be noted that the impact of this approach is subject to the proportion of graphical features used.

5.2 Reflection

We were able to establish a baseline performance level for a combined fake news classification system which utilizes the bag of words representation and rudimentary graph dispersal metrics in combination with classical machine learning algorithms while also highlighting notable performance trade-offs. This thesis also serves as an exploration of fake news classification using less computationally expensive methods in comparison a vast number of previously undertaken approaches. However, a notable deficiency in the undertaken approach is the lack of extensive exploration of text and graph features using more specialized methods. The robustness of the study is also limited by the use of a single pre-existing dataset.

5.3 Recommendations for Future Work

Further confirmation of the validity of the proposed approach can be explored via investigating its performance with regard to early detection of fake news as explored by (Liu et al., 2018). The adequacy of the data representation method utilized in this study can also be further validated by utilization in tandem with neural and deep learning methods. The findings presented in this study can also be enhanced by an extensive study of the impacts specific propagation metrics on the classification accuracy in order to identify the most relevant features.

References

Allcott, H., & Gentzkow, M. (2017). Social media and fake news in the 2016 election. *Journal of Economic Perspectives*. doi: 10.1257/jep.31.2.211

aurelien geron. (2017). Hands on machine learning with scikit learn and tensorflow concepts tools and techniques to build intelligent systems. *No journal available*. doi: undefined

Chandra, S., Mishra, P., Yannakoudakis, H., Shutova, E., Nimishakavi, M., & Saeidi, M. (2020). Graph-based modeling of online communities for fake news detection. *arxiv computation and language*. doi: null

Esmaeilzadeh, S., Peh, G. X., & Xu, A. (2019). Neural abstractive text summarization and fake news detection. *arXiv: Computation and Language*. doi: null

jacob devlin, mingwei chang, kenton lee, & kristina toutanova. (2018). Bert pre training of deep bidirectional transformers for language understanding. *arxiv computation and language*. doi: undefined

Japkowicz, N., & Shah, M. (2011). Evaluating learning algorithms: A classification perspective. *No journal available*. doi: null

Liu, Y., Liu, Y., Wu, Y.-F. B., & Wu, B. (2018). Early detection of fake news on social media through propagation path classification with recurrent and convolutional networks. *null*. doi: null

Manning, C. D., Raghavan, P., & Schütze, H. (2008). Introduction to information retrieval: Scoring, term weighting, and the vector space model. *No journal available*. doi: 10.1017/cbo9780511809071.007

Monti, F., Frasca, F., Eynard, D., Mannion, D., & Bronstein, M. M. (2019). Fake news detection on social media using geometric deep learning. *arXiv: Social and Information Networks*. doi: null

Ramos, J. (1999). *Using tf-idf to determine word relevance in document queries.*

Ruchansky, N., Seo, S., & Liu, Y. (2017). Csi: A hybrid deep model for fake news detection.

arxiv learning. doi: 10.1145/3132847.3132877

Shu, K., Mahudeswaran, D., Wang, S., Lee, D., Lee, D., & Liu, H. (2018). Fakenewsnet: A data repository with news content, social context and spatialtemporal information for studying fake news on social media. *arXiv: Social and Information Networks*. doi: null

Shu, K., Mahudeswaran, D., Wang, S., & Liu, H. (2019). Hierarchical propagation networks for fake news detection: Investigation and exploitation. *arXiv: Social and Information Networks*. doi: null

Snoek, J., Larochelle, H., & Adams, R. P. (2012). Practical bayesian optimization of machine learning algorithms. *arXiv: Machine Learning*. doi: null

Tacchini, E., Ballarin, G., Vedova, M. L. D., Moret, S., de Alfaro, L., & de Alfaro, L. (2017). Some like it hoax: Automated fake news detection in social networks. *null*. doi: null

Vicario, M. D., Bessi, A., Zollo, F., Petroni, F., Scala, A., Caldarelli, G., ... Quattrociocchi, W. (2016). The spreading of misinformation online. *Proceedings of the National Academy of Sciences of the United States of America*. doi: 10.1073/pnas.1517441113

Yang, K. C., Yang, K.-C., Niven, T., & Kao, H.-Y. (2019). Fake news detection as natural language inference. *arXiv: Computation and Language*. doi: null

Zhang, J., & Luo, Y. (2017). Degree centrality, betweenness centrality, and closeness centrality in social network. *No journal available*. doi: 10.2991/msam-17.2017.68

Zhang, X., Cao, J., Cao, J., Li, X., Sheng, Q., Zhong, L., ... Shu, K. (2020). Mining dual emotion for fake news detection. *arXiv: Computation and Language*. doi: 10.1145/3442381.3450004

Zhao, Z., Zhao, J., Sano, Y., Levy, O., Takayasu, H., Takayasu, M., ... Havlin, S. (2018). Fake news propagates differently from real news even at early stages of spreading. *EPJ Data Science*. doi: 10.1140/epjds/s13688-020-00224-z

Zhou, X., & Zafarani, R. (2018). Fake news: A survey of research, detection methods, and opportunities. *arXiv: Computation and Language*. doi: null

Appendix A

Supplementary Materials I

1. FakeNews Jupyter Notebook
2. Graph — Folder of JSON files containing the graph data used in this study
3. Text — Folder of CSV files containing the textual data used in this study
4. requirements.txt — List of python packages used in this study