Disinformation Campaign : Data Science Approach to Address the Real-World International Policy Challenge

Jinjae Lee
Department of Intelligence Computing
Hanyang University
Seoul, Korea
jjlee93@hanyang.ac.kr

Jiyeong Oh
Department of Intelligence Computing
Hanyang University
Seoul, Korea
ojyhi010309@hanyang.ac.kr

Michael Ing
College of Computing and Digital Media
DePaul University
Chicago, US
sing@depaul.edu

Varsha Sajja
College of Computing and Digital Media
DePaul University
Chicago, US
vsajja@depaul.edu

***Abstract*— The development of Internet technology has given us a lot of convenience. In particular, it has a positive function of easily obtaining the information that people want when they need it, but on the contrary, as more individuals and groups seek profit by exploiting these characteristics, it has caused chaos around the world in politics and economy.**

Due to the exponential increase in data, it is currently difficult for a person to directly determine the authenticity of the data. Accordingly, attempts to solve problems through data-driven approaches have increased. Currently, a lot of research to classify fake information is being conducted, and natural language processing is used as a method.

In this study, for an accurate fake news prediction model, Kaggle's fake news dataset was used, and machine learning techniques and deep learning techniques were proposed. In the machine learning technique, TF-IDF, which calculates the frequency and importance of words used in news articles, was used, and in the deep learning technique, data learning and testing were performed using word embeddings. It showed 99% performance on data we used in all machine learning (Random Forest, Logistic Regression, Stochastic Gradient Descent, SVM) and deep learning (LSTM, GRU) techniques. But in that the data of the collected news articles was insufficient, it showed a disappointing performance of 50% for completely new data.

Keywords—Fake News, Disinformation Campaign, Kaggle Fake News Prediction, Natural Language Processing, Random Forest, Logistic Regression, Stochastic Gradient Descent, SVM, LSTM, GRU

I. Introduction

Advances in technology have brought many advantages to our lives. Light bulbs increased the amount of time that humans can be active in a day, and vehicles, such as cars, trains, and airplanes, eased the restrictions on the physical distance that humans are able to act. There are many technologies that make people's lives richer, but computers and the Internet, which appeared in the 20th century, are now indispensable technologies. A typical characteristic of the Internet is the ease of access to information. Anyone can create, provide, and have access to information, regardless of distance. Due to these characteristics, the method and preference of information delivery from traditional media such as news tv and radios and newspapers to the Internet and social media have changed. The advent of social media has made information acquisition and dissemination quick, and people are easily exposed to such a large amount of information and tend to believe easily. The problem is that the authenticity of the information is unclear. Fake news created to intentionally deceive people can cause great social, political, and economic chaos. For example, fake news in the 2016 US presidential election became a big issue. Such information is easily disseminated by users through social network services, regardless of intention or not. Data has exploded due to services such as Twitter and Facebook, where posts can be shared without investigation or verification of the data, and people have reached the limit of directly controlling a huge amount of data. To promote healthy consumption of news, societies need to identify dis-informational news and many researchers are working in the field to solve a problem that can cause havoc around the world now. This paper proposes that societies could provide tools to detect and to identify fake and real news by using machine learning and deep learning to categorize news into real and fake news, using term-frequency inverse document frequency (TFIDF) and Word Embedding to extract features from text. Then offer the model in the forms of end users' tools that could foster better consumption of news by policy makers or citizens or organizations.

II. Literature Review

Disinformation detection is being done by many organizations. One such organization is https://www.politifact.com/, which exists since 2007. Politifact uses trained journalists to investigate news information that need verification. https://www.FactCheck.org is another site dedicate to politics. It fact checks politicians and their ads, interviews, debates and the statements using journalists. https://fullfact.org/ is located in England; it fact checks politicians, public institutions and journalists and viral content.

https://leadstories.com/ concentrates on virial news stories. https://mediabiasfactcheck.com/ focuses on the bias, deceptive news practices, factual accuracies and credibility of media source. https://www.snopes.com/ fact checks anything that is trending and of interest to the readers. It is the oldest and largest fact checking organization, which started in 1996. The commonality among these various fact-checking organizations is they all employ people, whether they be journalists or experts. Such people centered organizations do not scale well because of the volume and speed of information that is being churned out on social media and by news organizations and fake news sites. Their ability to fact check is inherently limit to the number of staffs and their expertise. These sites are premium mediums for fact-checking, but they do not scale because human resources are limited, especially great human resource.

How do we scale the identification of a piece of news as real or fake so that society could benefit from accurate information? How do we provide the first line of defense for a piece of news? We propose machine learning as the means to determine whether a piece of news is real or fake. Our team review the literatures on what has been done in this domain with respect to using machine learning to detect misinformation.

After 2016, there has been an uptick of machine learning research done to detect misinformation. In many articles, one persistent question comes up. What data features are needed for machine learning? A corollary question is: what embedding methods are available to turn text into numbers so that machine learning could be applied? Finally, what machine learning techniques should be applied?

What features should be included to maximize machine learning identification of fake and real news? In article the "Fake news detection: a survey of evaluation datasets" describes what features of a fake-real news dataset might be effective to detect fake news. The author cites four major components creator/spreader, target victims, news content, and social context. Social context "can be related both to the platforms used to spread the news and to the distribution pattern--community of users or broadcast pattern [13, p. 7]." From these four broad categories, D'Ulizia goes on to define eleven features that might be important to effectively detect fake and real news:

- News domain: the dataset can contain fake news items that target certain news domains, such as health, education, tourism, sport, economy, security, science, IT, and political election.

- Application purpose: datasets can be built for different aims, such as fake detection, fact-checking, veracity classification, and rumor detection. The first consists of the prediction of the chances of a particular piece of information (news article, reviews, posts, etc.) being intentionally deceptive. Fact-checking refers to the process of vetting and verifying factual statements contained in a piece of information; unlike fake detection, fact-checking works at the level of a particular statement or claim. Veracity classification is very similar to fake detection but attempts to predict the actual truth of a given piece of information. Finally, rumor detection tries to distinguish between verified and unverified information (instead of true or false) and the unverified information may turn out to be true or false, or may remain unresolved.

- Type of disinformation: fake news or misleading news may be categorized as fake reviews, fake advertisements, or fake news articles according to the types of false information they contain. Fake news articles can be further classified into (a) hoaxes, considered to be false information deliberately fabricated to masquerade as the truth (Sharma et al., 2019); (b) rumors, which refers to unsubstantiated claims that are disseminated without any evidence to support them (Sharma et al., 2019); and (c) satire, which uses humor, irony, exaggeration, ridicule, and false information to present the news.

- Language: this concerns the language of the fake news contained in the dataset, which can be written in different languages according to the sources used to retrieve them. The dataset can be categorized as multi-lingual or monolingual.

- Size: the size of the dataset is commonly determined by the number of news items that it contains. It can also be measured in kilobytes/megabytes of the overall archive.

- News content type: this concerns linguistic and syntactic features, such as headlines and body text, as well as images and videos of the news. This survey considers the following four types of news content (as they characterize the 27 datasets): headlines, body text, images, and videos.

- Rating scale: this concerns the labels that are associated with the news contained in the datasets and used to rate the truthfulness of the news. Different rating scales can be used containing a different number of rating levels. For instance, a five-point rating scale can have the following labels: true, mostly true, mixture of true and false, mostly false, and false. An example of a three-point rating scale is true, false, or unverified.

- Media platform: this concerns the digital environment where the news collected in the dataset is shared and spread to the audience. Two main types of media platforms can be used to share and transfer fake news: (i) mainstream media, meaning the traditional media, such as newspapers, TV, and radio, and (ii) online social media, such as Twitter, Facebook, Instagram, and blogs. In this paper, mainstream media include also traditional media (e.g., NBC News, Washington Post, etc.) that have extended the way they spread information from mainstream platforms also to digital platforms.

- Spontaneity: fake news in the dataset may be spontaneous if it is automatically extracted from public web sources without editing, or artificial if it is manually generated by asking someone to produce items by manipulating real ones.

- Availability: this concerns free online availability of the data contained in the dataset.

-Extraction period: this concerns the definition of a specific time frame during which the data have been collected [24, p. 9-10].

Other researchers suggest using user profiles, instead of context and content. They are concern with only the targets of fake news. In the article "The Role of User Profiles for Fake News Detection" explores the utilization of user profiles for fake news detection. The article asks three questions: who is more likely to share real/fake news, what are the characteristics of these users, and are these features useable for fake news detection. The user profile features captured are age, personality, location, political bias, profile image, etc. The top most important features in order of importance, based on Gina impurity, are register time, verified, political bias, personality, and status Count [32].

Yet, another approach that does not rely on user profiles, or context or context is fake news propagation. One article "Fake News Detection on Social Media using Geometric Deep Learning" pursues propagation as a means to detect fake news rather than the content of the news stories or social context, which includes gender, age, political affiliation, users' social structure and their reactions. Geometric deep learning refers to non-Euclidean deep learning approaches. This propagation based method offers the advantages of language independence and resistance to adversarial attacks. The authors' "underlying core algorithms are a generalization of classical convolutional neural networks to graphs, allowing the fusion of heterogeneous data such as content, user profile and activity, social graph, and news propagation [26, p. 1]."

Finally, another article mixes content, social context and Spatiotemporal Information to determine fake and real news. In the article, "FakeNewsNet: A Data Repository with News Content, Social Context, and Spatiotemporal Information for Studying Fake News on Social Media" offers a repository with eight features for fake news detection on social media. The authors compare the FakeNewsNet dataset, which they created, to BuzzFeedNews, LIAR, BS Detector, CREDBANK, BuzzFace, FaceBookHoax repositories. FakeNewsNet dataset has linguistic, visual, user, post, response, network, spatial and temporal features to address the deficiencies in other fake news datasets. But the later repositories are missing some of the eight features. BuzzFeedNews, LIAR and BS Detector have only linguistic feature; CREDBANK has linguistic, user, post, spatial and temporal features [25].

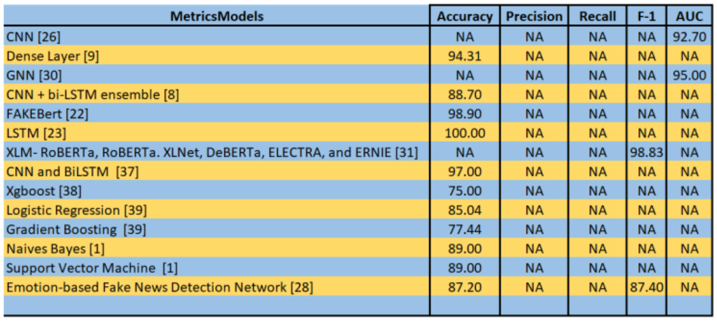
What embedding methods are available to turn text into numbers so that machine learning could be applied? There are several means to vectorize the text. Thota et al employ bag of word and TF-IDF, which are well-known natural language processing techniques, to express words as vectors [9]. Some researchers use transfer learning such a BERT with their bag of word [30]. A hybrid model is proposed with BERT-based (Bidirectional Encoder Representations from Transformers) – FakeBERT by combining various parallel blocks of the single-layer deep convolutional neural network (CNN) having different kernel sizes and filters with the BERT which applies to both the binary as well as multi-class real-world fake news dataset [22]. There are also hashing and indexing.

Finally, what machine learning techniques should be applied? Researchers have utilized many different models for the detection of fake and real news. Abedalla used Bi-LSTM and Attention Mechanism [5]. Nasir et al. presented a Hybrid CNN-RNN model combining Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN) as a method for detecting fake news [6]. Kumar et al used CNN + bidirectional LSTM ensembled network with attention mechanism achieved the highest accuracy of 88.78% [8]. Another employed n-gram models with the best performance by using TF-IDF, logistic regression with high accuracy, Boolean label crowdsourcing techniques with high accuracy, deep neural network models which improve the performance, a combination of metadata with text improves accuracy, adversarial neural network, tractable Bayesian algorithm (Detective) minimizes the spread of false information. The parallel-capsule neural network models including the n-gram convolution layer were implemented for different lengths of the news statements [13].

A two-step method focusing on identifying fake news on social media has been proposed in this study [16]. A fake news detection model has been proposed with 23 different supervised artificial algorithms (BayesNet, JRip, OneR, Decision Stump, ZeroR, Stochastic Gradient Descent, CV Parameter Selection, Randomizable Filtered Classifier, Logistic Model Tree, Locally Weighted Learning, Classification via Clustering, Weighted Instances Handler Wrapper, Ridor, Multi-Layer Perceptron, Ordinal Learning Model, Simple Cart, Attribute Selected Classifier, J48, Sequential Minimal Optimization, Bagging, Decision Tree, IBk and Kernel Logistic Regression) after pre-processing the dataset from the unstructured to the structured dataset with text mining methods. Finally, the combined model using text mining and supervised artificial intelligence algorithms has been tested on real-world datasets and accuracy has been evaluated about the best mean values for Decision Tree, ZeroR, CV Parameter Selection and Weighted Instances Handler Wrapper algorithms. Improving/hybridizing the existing algorithms in the future can enhance the results for detecting fake news in real-world datasets.

The article "A Heuristic-driven Ensemble Framework for COVID-19 Fake News Detection" propose to use ensemble method to detect fake news related to COVID-19 tweets using a binary classification into real and fake classes. They employ different transfer models such as XLM- RoBERTa, RoBERTa. XLNet, DeBERTa, ELECTRA, and ERNIE. They manage to get F1-score of 98.31% during the competition and 98.83 post competition. They beat the leader board of 98.69% F1-score for the competition [31].

How effective are the different models we observed in the literature reviews? Below is a chart summarizing the effectiveness that we observed from the different studies:



MetricsModels	Accuracy	Precision	Recall	F-1	AUC
CNN [26]	NA	NA	NA	NA	92.70
Dense Layer [9]	94.31	NA	NA	NA	NA
GNN [30]	NA	NA	NA	NA	95.00
CNN + bi-LSTM ensemble [8]	88.70	NA	NA	NA	NA
FAKEBert [22]	98.90	NA	NA	NA	NA
LSTM [23]	100.00	NA	NA	NA	NA
XLM- RoBERTa, RoBERTa. XLNet, DeBERTa, ELECTRA, and ERNIE [31]	NA	NA	NA	98.83	NA
CNN and BiLSTM [37]	97.00	NA	NA	NA	NA
Xgboost [38]	75.00	NA	NA	NA	NA
Logistic Regression [39]	85.04	NA	NA	NA	NA
Gradient Boosting [39]	77.44	NA	NA	NA	NA
Naives Bayes [1]	89.00	NA	NA	NA	NA
Support Vector Machine [1]	89.00	NA	NA	NA	NA
Emotion-based Fake News Detection Network [28]	87.20	NA	NA	87.40	NA

The survey of the scores reveals that machine learning could be employed to detect fake and real news. Machine learning can comfortably identify fake and real news with high accuracy. The survey of the scores provides the basis for evaluating the scores of our models as well.

III. Methodology

The proposed approach can be seen in Figure 1. We collected and used the data published on Kaggle, and data preprocessing was performed. Since the ratio of the target class of the used data set does not differ significantly, so data balancing work is not required.

A. *Data Collection and Description of the Dataset*

The dataset used to detect fake news is published on the Kaggle site, and the Fake News file with 23,502 data and the True News file with 21,417 data were used. The attributes of the data set consist of four features. A brief description of the properties can be found in Table 1. Actually, we created a model using two features, title and text, and since the goal of our project was to create a model that distinguishes whether news data is fake or true, we created a new column and set 0 for fake news and 1 for true news. It was used as the target class. The other two features, news topic and date features, were removed in the pre-processing stage because we only used them to check what kind of data we used and when it was written, but were not used to build a model.

TABLE I. FEATURE OF DATASET

No.	Kaggle Fake-True News Dataset		
	Feature Name	***Attribute***	***Data Type***
1	title	Title of News	str
2	text	Body of News	str
3	subject	Subject of News	str
4	date	News release date	date

Additionally, extra data have been collected from Onion and Korean article sites for model performance tests on articles that do not belong to the dataset.

B. *Preprocessing of the Dataset*

In order to create a data-driven model, the focus should be on the preprocessing part. Just like the saying that garbage in garbage out, it means that input data is important for the model to perform accurately. It is important to understand what properties the data we will use and what problems it has. In the preprocessing part, it is common to check for duplicate data and missing values. When multiple pieces of duplicate data are learned, the importance of that data increases, so there is a high probability that it will not perform well on other data. A missing value means that there is no data for that attribute in specific row. This part is highly likely to cause errors in the learning process or adversely affect the performance of the model. Since the feature to be used as the main feature for learning in our dataset is the body of news, we removed duplicate data from the feature to have only unique values, and also removed data with missing values. The news title is also composed of strings just like the news body, and it is judged that it will affect the model performance, so the news title and the news body are integrated into one feature called document. Since our goal is to distinguish between true news and fake news based on the content of news articles, a label for authenticity of news is needed during supervised learning. The files we have are fake data and true data, so we created a target class that distinguishes fake as 0 and true as 1. The subject and date features were removed as they are not useful features in the model we want to build.

Unlike general structured data, the data we use to create a model consists of strings. A language that occurs naturally in society and that humans use for communication is called a natural language, and a language intentionally created by humans, such as a programming language used in a computer, is called an artificial language. Natural language processing means converting the natural language we use to make it easier for computers to understand, and natural language processing is necessary for computers to understand news articles and make prediction model based on them. Looking through the data, refining texts was necessary as some of the contents of the article were empty which is useless for analysis. In addition, it was also essential to change text into a form that could be recognized by computers and organize it into a form that is easy to analyze in consideration of various characteristics of language. In this process, a representative natural language processing library, NLTK, was used. Below is the list of preprocessing applied to the data.

- Cleaning : Rows with empty values in the column 'text', were removed in batches.
- Stop words removing : The stop word serves as a link between sentences and sentences. Conjunctions such as "but", "and", and "that", prepositions such as "in", "for", "of", "from", etc., articles such as "a", "an", and "the" are included in stop word. Stop word that does not have much meaning in the word itself were removed in the preprocessing step. An example of stop word removal can be seen in Figure 2.

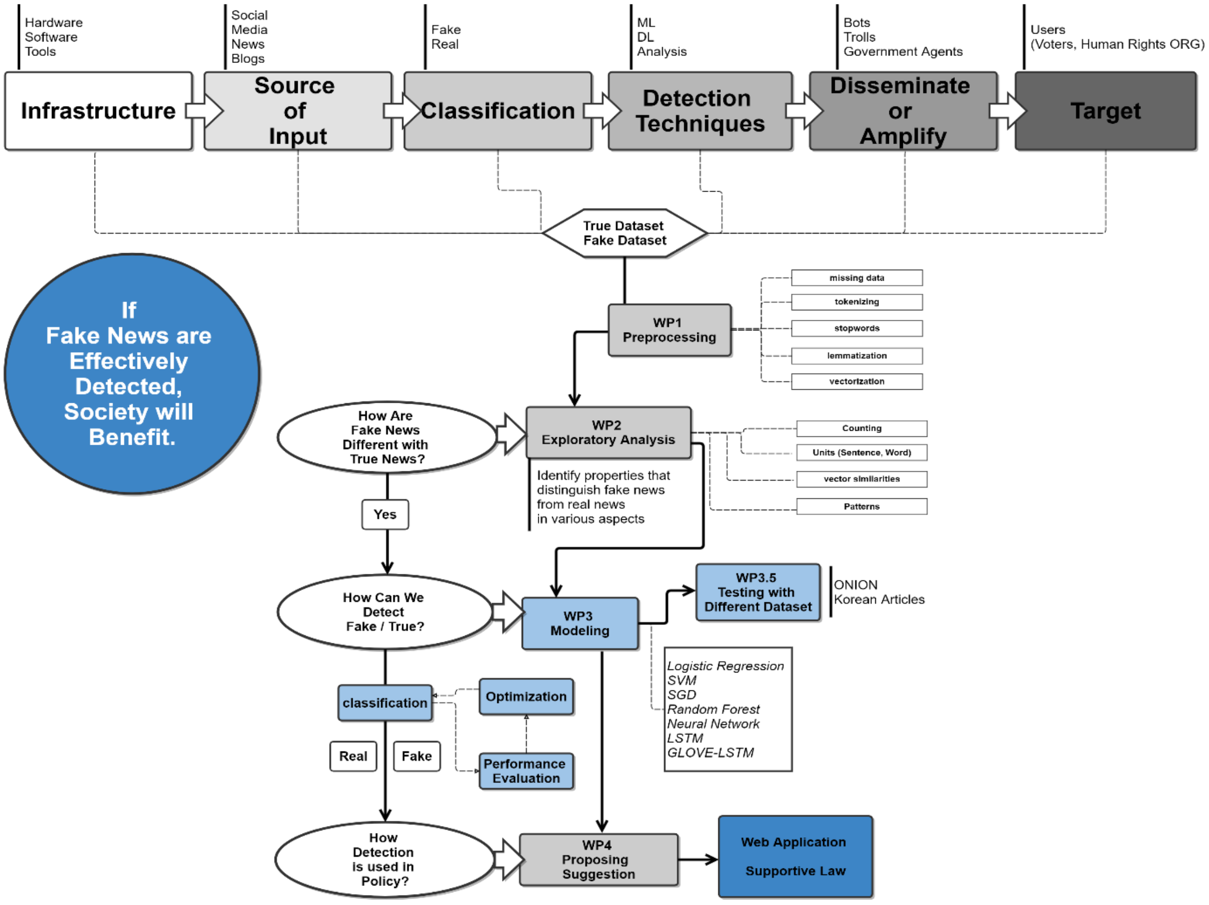
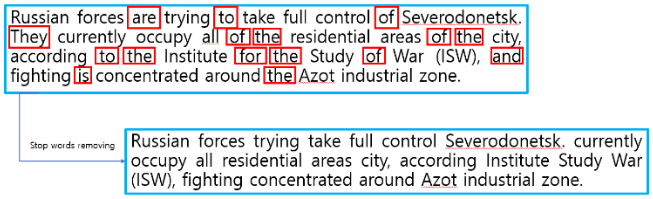


Fig. 1. Flow Chart



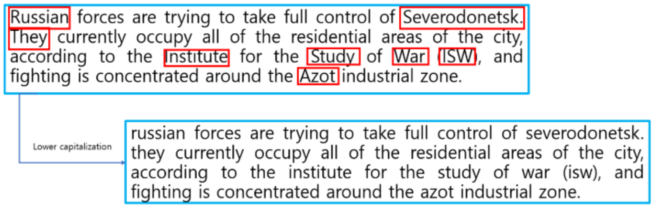
Russian forces are trying to take full control of Severodonetsk. They currently occupy all of the residential areas of the city, according to the Institute for the Study of War (ISW), and fighting is concentrated around the Azot industrial zone.

Stop words removing

Russian forces trying take full control Severodonetsk. currently occupy all residential areas city, according Institute Study War (ISW), fighting concentrated around Azot industrial zone.

Fig. 2. Example for Stop words removing

- Lower capitalization : Since computers do not recognize upper and lower-case letters as the same, they are all unified into lower case letters. An example of lower capitalization can be seen in Figure 3.



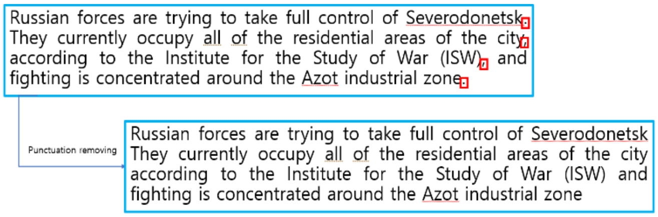
Russian forces are trying to take full control of Severodonetsk. They currently occupy all of the residential areas of the city, according to the Institute for the Study of War (ISW), and fighting is concentrated around the Azot industrial zone.

Lower capitalization

russian forces are trying to take full control of severodonetsk. they currently occupy all of the residential areas of the city, according to the institute for the study of war (isw), and fighting is concentrated around the azot industrial zone.

Fig. 3. Example of lower capitalization

- Punctuation removing : Punctuation marks such as " . ", " , ", " ! ", " ? " etc. have also been removed in the preprocessing step because they do not have much meaning by themselves, an example can be seen in Figure 4.

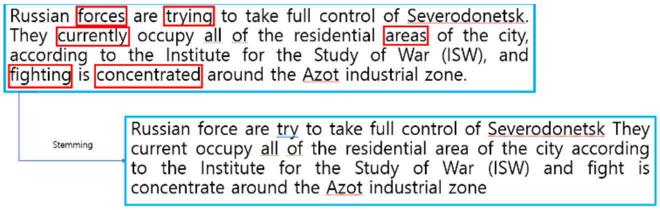


Russian forces are trying to take full control of Severodonetsk. They currently occupy all of the residential areas of the city, according to the Institute for the Study of War (ISW), and fighting is concentrated around the Azot industrial zone.

Russian forces are trying to take full control of Severodonetsk They currently occupy all of the residential areas of the city according to the Institute for the Study of War (ISW) and fighting is concentrated around the Azot industrial zone

Fig. 4. Example for Punctuation removing

- Stemming : Stemming is a technique that removes prefixes and suffixes from words, and it has the advantage of reducing the complexity of calculations in the model learning stage by matching the forms of words with similar or identical meanings to a common basic form. An example of stemming can be seen in Figure 5.

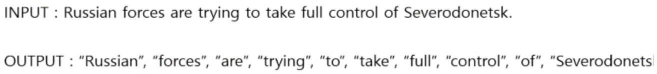


Russian forces are trying to take full control of Severodonetsk. They currently occupy all of the residential areas of the city, according to the Institute for the Study of War (ISW), and fighting is concentrated around the Azot industrial zone.

Russian force are try to take full control of Severodonetsk They current occupy all of the residential area of the city according to the Institute for the Study of War (ISW) and fight is concentrate around the Azot industrial zone

Fig. 5. Example for Stemming

- Tokenization : Tokenization refers to the operation of dividing data into units called tokens according to certain criteria. The unit of the token is different depending on the situation, but in general, the token is defined as a meaningful unit, and in our project, the standard of the token is set as a word. An example of a tokenization operation can be seen in Figure 6.



INPUT : Russian forces are trying to take full control of Severodonetsk.

OUTPUT : "Russian", "forces", "are", "trying", "to", "take", "full", "control", "of", "Severodonets

Fig. 6. Example for Tokenization

After all the preprocessing was performed, the data were purified from a total of 44,898 rows to 44267 rows.

TABLE II. PREPROCESSING RESULT

	Number of Rows
Before Processing	44898
After Processing	44267

C. Exploratory Data Analysis

The goal of this step was set to identify the distribution of data, to extract features by vectorizing data in the form of strings and to discover what differences exist in the extracted features between fake and real news. Various methods such as counting method, TF-IDF, PCA, and T-SNE had been used, and as a result it was found that there were discriminatory features between the two types of news.

- Topic Distribution : The distribution of the 'subject' feature of each data was visualized. As a result, it was determined that A significant proportion of the datasets had 'politics' as the subject. Figure 7 shows in a pie chart that indicates topics such as 'left-news', 'politics', 'politics news', and 'government news' are the main topics. This means that the model will be trained specializing in political news detection in the subsequent modeling process, while also suggesting the possibility of being vulnerable to detecting articles with topics other than politics.

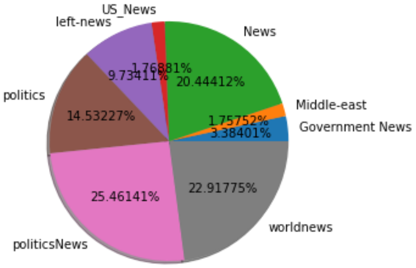


Fig. 7. Topic Distribution

- Word Length Distribution : Distribution of the number of words were also counted, for both title and content, respectively. As a result, it was confirmed that most news titles consist of less than 10 words, and news texts mostly consist of less than 1000 words.

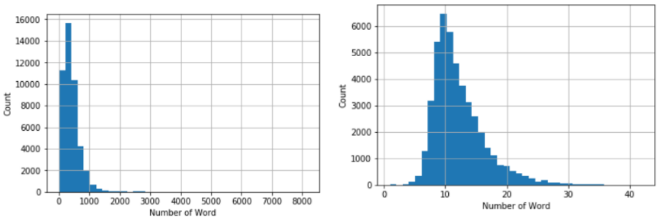


Fig. 8. Word Length Distribution

- TF-IDF Cosine Similarities : To determine whether there is a difference between 'key-words' in fake news and those in real news, TF-IDF, which calculates the importance of words in documents, was used. 500 samples were randomly selected from each kind of dataset, and the two groups of samples were combined to make a total of 1000 news dataset. Every individual document was vectorized by TF-IDF vectorizer, and the importance of words in documents was calculated for each document. In order to exclude words that appear in any document, regardless of the type of document, and words that appear meaninglessly less frequently, only TF-IDF of words that appear in at least 300 to 9500 documents out of 1000 samples was calculated. Applying cosine-similarity (0 ~ 1) to the calculated vector, the similarity among 1000 documents was compared, subsequently.

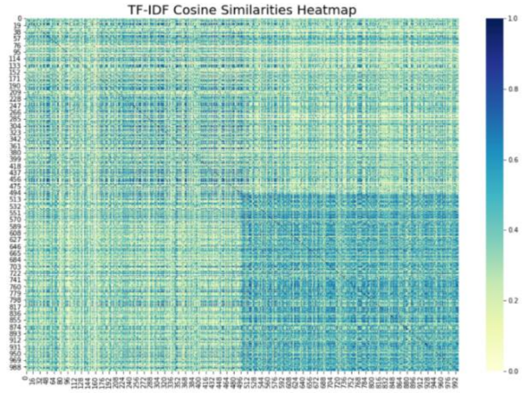


Fig. 9. TF-IDF Cosine Similarities Heatmap

In the entire sample data, document index numbers 0 to 499 were set to fake news, and the other half were set to real news. When the cosine similarity degree was displayed as a heat map, it was confirmed that the similarity between the same class was relatively higher than that between the different classes. Especially, the similarities between real news showed higher scores, compared with other cases.

TABLE III. AVERAGE TF-IDF COSINE SIMILARITY

	fake-fake	***true-fake***	***true-true***
average similarity	0.32	0.30	0.46

Consistent with the results of the heat map, it can be seen that the average of similarity between true articles (0.46) is higher than in other cases (0.32 and 0.30). These results lead to the hypothesis that the 'keywords' used in fake news and real news are different.

- TF-IDF Word Cloud : Knowing that the TF-IDF similarities of important words (key words) in fake news and true news are different, checking how the key words in each group differ to the other was implemented by Word Cloud. Randomly sampling 1000 articles from each group of dataset, top 100 words that were most frequently selected as keywords in each group, were designated as representative words. Making the frequency and letter size proportional, the representative words of each group were expressed in a word cloud.

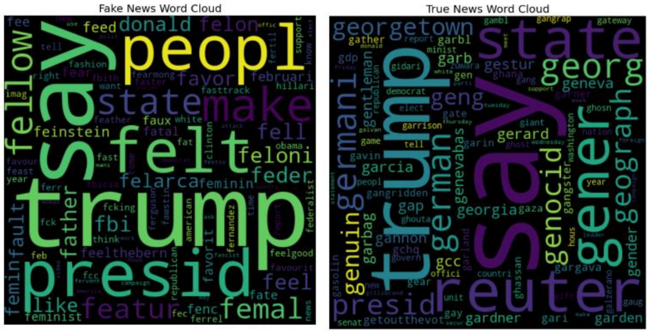


Fig. 10. Word Cloud

As a result of Word Cloud, there were some overlapping words among the keywords of fake and real news; This is inferred that most of the topics in the article are 'politics'. However, words selected as keywords only in certain groups were also found. This difference suggests the direction of the report in that it leads to the hypothesis that words that support the credibility of reliable news and words that fake news uses to incite the public, can be discovered by analyzing methods.

- Key Word Extraction with TF-IDF + T-SNE : To understand the relationship among key words that are detected to be important in each document, a tf-idf matrix of top 100 keywords is expressed in two dimensions. Top 100 words with highest tf-idf were selected from 1,000 samples, and t-SNE, which is a method of embedding similarities of data existing on high dimensions in low dimensions (two dimensions, here), was applied to decrease the dimension of the matrix from 1,000 to 2.

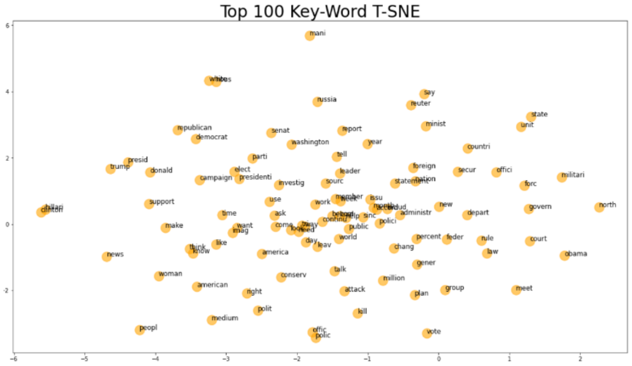


Fig. 11. Top 100 Key-Word T-SNE

Plotting 100 words on the 2-dimensional plane, it was confirmed that the distance between specific points was close, that is, the similarity between specific words was high, while the similarity between certain points was low. This leads to the inference that there is a specific relationship among words, which is meaningful in that it can be used as weight or feature integration criteria when determining words to be used for fake news detection in the report.

- TF-IDF Word Cloud : Based on the TF-IDF of each document, PCA and TSNE were used together to examine how fake news and reliable news are distributed and how they differ. 500 samples were randomly taken for each data group to create 1000 datasets, and after creating a TF-IDF matrix of keywords for each document, the matrix was first reduced to 50 dimensions with PCA. Then, using T-SNE, each document was expressed in 1,000 dots in two dimensions.

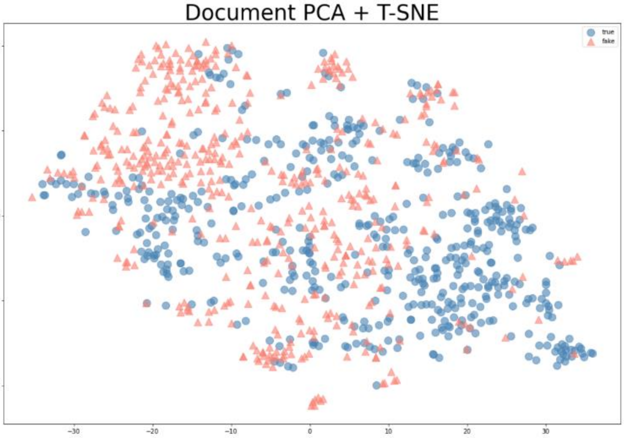
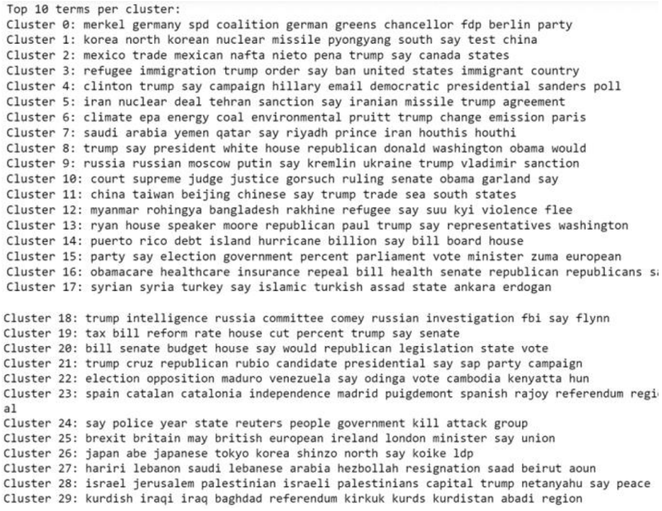


Fig. 12. PCA + T-SNE

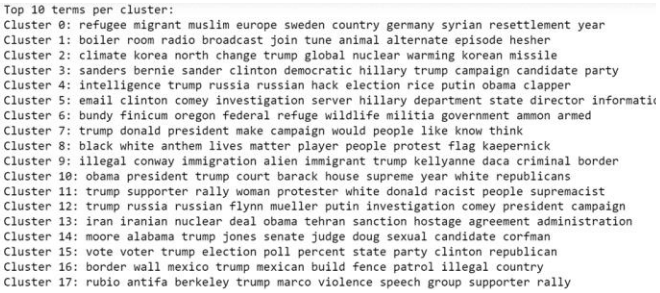
The distribution of documents reduced to two dimensions identified where the density of fake news is high and where that of real news is high. This supports the hypothesis that the key words of the document, the importance score of the word, and dimensions can be used as a powerful tool to distinguish fake news from true news.

- Clustering with K-means Using 30 Clusters using TF-IDF : Examining the top 10 words in each cluster for both True and Fake News suggest that True News words appear closer associated with each other than the words for Fake News.

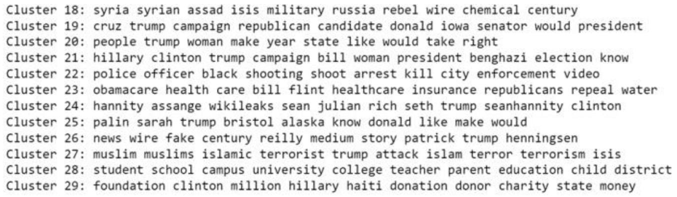


```
Top 10 terms per cluster:
Cluster 0: merkel germany spd coalition german greens chancellor fdp berlin party
Cluster 1: korea north korean nuclear missile pyongyang south say test china
Cluster 2: mexico trade mexican nafta nieto pena trump say canada states
Cluster 3: refugee immigration trump order say ban united states immigrant country
Cluster 4: clinton trump say campaign hillary email democratic presidential sanders poll
Cluster 5: iran nuclear deal tehran sanction say iranian missile trump agreement
Cluster 6: climate epa energy coal environmental pruitt trump change emission paris
Cluster 7: saudi arabia yemen qatar say riyadh prince iran houthis houthi
Cluster 8: trump say president white house republican donald washington obama would
Cluster 9: russia russian moscow putin say kremlin ukraine trump vladimir sanction
Cluster 10: court supreme judge justice gorsuch ruling senate obama garland say
Cluster 11: china taiwan beijing chinese say trump trade sea south states
Cluster 12: myanmar rohingya bangladesh rakhine refugee say suu kyi violence flee
Cluster 13: ryan house speaker moore republican paul trump say representatives washington
Cluster 14: puerto rico debt island hurricane billion say bill board house
Cluster 15: party say election government percent parliament vote minister zuma european
Cluster 16: obamacare healthcare insurance repeal bill health senate republican republicans s
Cluster 17: syrian syria turkey say islamic turkish assad state ankara erdogan

Cluster 18: trump intelligence russia committee comey russian investigation fbi say flynn
Cluster 19: tax bill reform rate house cut percent trump say senate
Cluster 20: bill senate budget house say would republican legislation state vote
Cluster 21: trump cruz republican rubio candidate presidential say sap party campaign
Cluster 22: election opposition maduro venezuela say odinga vote cambodia kenyatta hun
Cluster 23: spain catalan catalonia independence madrid puigdemont spanish rajoy referendum regi
al
Cluster 24: say police year state reuters people government kill attack group
Cluster 25: brexit britain may british european ireland london minister say union
Cluster 26: japan abe japanese tokyo korea shinzo north say koike ldp
Cluster 27: hariri lebanon saudi lebanese arabia hezbollah resignation saad beirut aoun
Cluster 28: israel jerusalem palestinian israeli palestinians capital trump netanyahu say peace
Cluster 29: kurdish iraqi iraq baghdad referendum kirkuk kurds kurdistan abadi region
```



```
Top 10 terms per cluster:
Cluster 0: refugee migrant muslim europe sweden country germany syrian resettlement year
Cluster 1: boiler room radio broadcast join tune animal alternate episode hesher
Cluster 2: climate korea north change trump global nuclear warming korean missile
Cluster 3: sanders bernie sander clinton democratic hillary trump campaign candidate party
Cluster 4: intelligence trump russia russian hack election rice putin obama clapper
Cluster 5: email clinton comey investigation server hillary department state director informati
Cluster 6: bundy finicum oregon federal refuge wildlife militia government ammon armed
Cluster 7: trump donald president make campaign would people like know think
Cluster 8: black white anthem lives matter player people protest flag kaepernick
Cluster 9: illegal conway immigration alien immigrant trump kellyanne daca criminal border
Cluster 10: obama president trump court barack house supreme year white republicans
Cluster 11: trump supporter rally woman protester white donald racist people supremacist
Cluster 12: trump russia russian flynn mueller putin investigation comey president campaign
Cluster 13: iran iranian nuclear deal obama tehran sanction hostage agreement administration
Cluster 14: moore alabama trump jones senate judge doug sexual candidate corfman
Cluster 15: vote voter trump election poll percent state party clinton republican
Cluster 16: border wall mexico trump mexican build fence patrol illegal country
Cluster 17: rubio antifa berkeley trump marco violence speech group supporter rally
```



```
Cluster 18: syria syrian assad isis military russia rebel wire chemical century
Cluster 19: cruz trump campaign republican candidate donald iowa senator would president
Cluster 20: people trump woman make year state like would take right
Cluster 21: hillary clinton trump campaign bill woman president benghazi election know
Cluster 22: police officer black shooting shoot arrest kill city enforcement video
Cluster 23: obamacare health care bill flint healthcare insurance republicans repeal water
Cluster 24: hannity assange wikileaks sean julian rich seth trump seanhannity clinton
Cluster 25: palin sarah trump bristol alaska know donald like make would
Cluster 26: news wire fake century reilly medium story patrick trump henningsen
Cluster 27: muslim muslims islamic terrorist trump attack islam terror terrorism isis
Cluster 28: student school campus university college teacher parent education child district
Cluster 29: foundation clinton million hillary haiti donation donor charity state money
```

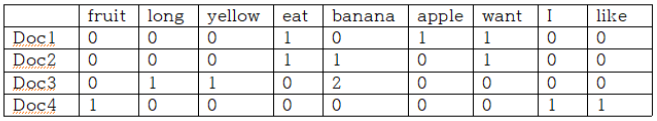
Fig. 13. K-Means Clustering

This observation is further enhanced when the Silhouette Coefficients were calculated; Silhouette coefficient shows how internally consistent the 30 clusters are; their values are 0.025 and 0.013, respectively, for True and Fake News. This leads the question, 'Might this suggest that real news texts are more coherent while fake news are less coherent? '. Different clustering sizes would need to be tried to find out if this is in fact true since the current coefficients are low.

D. *Word Representation*

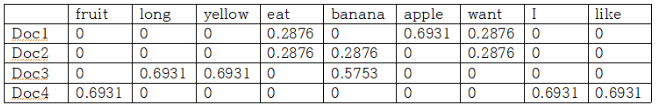
As mentioned above, it is difficult for a computer to understand text data, so we need to represent the data as numeric values so that it is easy for the computer to understand. There are many ways to represent text in natural language processing, but only two techniques were covered in this paper because only the word representation methods, TF-IDF and Glove, were tested in this project.

- TF-IDF : Term Frequency-Inverse Document Frequency (TF-IDF) refers to a statistical value indicating how important a word is in a specific document when there is a document group consisting of several documents. It can be mainly used for the task of finding the similarity of documents, the task of determining the importance of the search results in the search system, and the task of finding the importance of a specific word in the document. The characteristic of TF-IDF is that a word that appears frequently in all documents is judged to have low importance, and a word that appears frequently only in a specific document is judged to have high importance. That is, when the TF-IDF value is low, the importance of the word is low, and when the TF-IDF value is high, the importance of the word is high. For example, stop words such as "the" or "a" appear frequently in all documents, so the TF-IDF value of the stop word is lower than the TF-IDF value of other words. First table below is an example table showing the number of occurrences of each word for 4 documents. Second table is an example table in which TF-IDF values indicating importance according to the frequency of occurrence of words in a document are calculated.



	fruit	long	yellow	eat	banana	apple	want	I	like
Doc1	0	0	0	1	0	1	1	0	0
Doc2	0	0	0	1	1	0	1	0	0
Doc3	0	1	1	0	2	0	0	0	0
Doc4	1	0	0	0	0	0	0	1	1

Fig. 14. Number of occurrences of each word in a document



	fruit	long	yellow	eat	banana	apple	want	I	like
Doc1	0	0	0	0.2876	0	0.6931	0.2876	0	0
Doc2	0	0	0	0.2876	0.2876	0	0.2876	0	0
Doc3	0	0.6931	0.6931	0	0.5753	0	0	0	0
Doc4	0.6931	0	0	0	0	0	0	0.6931	0.6931

Fig. 15. TF-IDF value

- Glove : Glove is a word embedding methodology developed at Stanford University in 2014. Count-based methodologies such as TF-IDF consider the overall statistical information of the corpus, but have poor performance for inferring the meaning of words. And methodologies such as Word2Vec, which excel at word-to-word inference based on prediction, do not reflect the overall statistical information of the corpus because the embedding vector considers surrounding words only within the window size. Glove uses both methodologies to overcome the limitations of each method. The core idea of Glove is to make the dot product of the embedded central word and surrounding word vectors equal to the probability of co-occurrence in the entire corpus.

E. *Machine Learning*

Machine learning is the field of study and exploration of computer algorithms that automatically improve the performance of tasks through experience. Algorithms in machine learning can build mathematical models based on sample data known as "training data" to make predictions or decisions on new data without explicitly programming them. There are several techniques in machine learning, and the techniques we used to detect fake news are Logistic Regression, Random Forest, Stochastic Gradient Descent, Support Vector Machine and Multi-Layer Perceptron.

- Logistic Regression(LR) : Regression analysis is one of the statistical techniques for modeling the relationship between variables, and is used to predict future events between a dependent variable (target) and one or more independent variables (features). In general regression, it is used when the dependent variable is a continuous value (length, weight, etc.), but logistic regression is used when the dependent variable is discrete such as 0 or 1, true or false.

- Random Forest(RF) : Random Forest is one of the ensemble models, in which multiple decision trees are formed, new data is passed through each tree, and the tree with most votes is selected as the final classification result by voting with the classification results of each tree. Some trees generated by Random Forest may be overfitted, but since there are many other trees, overfitting in some trees does not have a significant effect. The advantage is that the model is simple, overfitting doesn't occur well, and it generalizes well to new data. However, since it uses multiple decision trees, it consumes a lot of memory and has disadvantages that it does not work well for high-dimensional data or sparse data.

- Stochastic Gradient Descent(SGD) : The basic principle of a machine learning algorithm is to find the parameter value at the point where the cost function for the predicted value is the minimum. When the training data set is put in and the model is trained, the model finds the optimal parameter based on the given value, and the method is to find the point where the cost function becomes the minimum value. Gradient descent is the method used to find the point at which the cost function has a minimum value. Gradient descent uses the differential coefficient to find the point where the slope of the cost function is zero. When training a machine learning algorithm, if it is difficult to use all the training data, it is called stochastic gradient descent to randomly select some samples from the training data set and proceed with gradient descent.

- Support Vector Machine(SVM) : The Support Vector Machine is a model that defines decision boundaries, that is, baselines for classification. When a new unclassified point appears, it is possible to perform the classification task by determining which side of the boundary it belongs to. As the dimension increases, the model becomes more complex, and the decision boundary also changes to a higher-dimensional form defined as a hyperplane rather than a line or plane form. The boundary that maximizes the margin, which means the distance between the decision boundary and the support vector, is defined as the optimal boundary that can best classify the two groups. SVM can be used for both classification and prediction problems, and it has high prediction accuracy among machine learning models. In addition, it works well for both low-dimensional and high-dimensional data, but it is difficult to interpret and understand the model, and it has the disadvantages of slow speed and large amount of allocated memory when building a model for large data.

- Multi-Layer Perceptron(MLP) : Multi-Layer Perceptron is a concept that can be said to be the beginning of a Neural Network, and it is a concept created with inspiration from the neurons that make up the human brain. Perceptron receives multiple inputs and link to one output, and the input is multiplied by weight and added with bias, passes through the activation function, and comes out as a probabilistic output (Figure 16). Since the single-layer perceptron has a limitation that non-linear classification is impossible, MLP overcomes the drawback by stacking multiple layers (Figure 17).

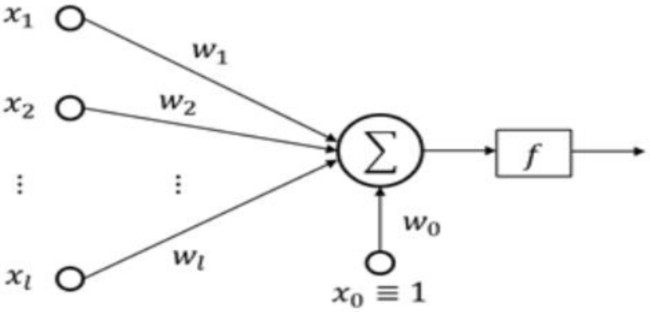


Fig. 16. Perceptron

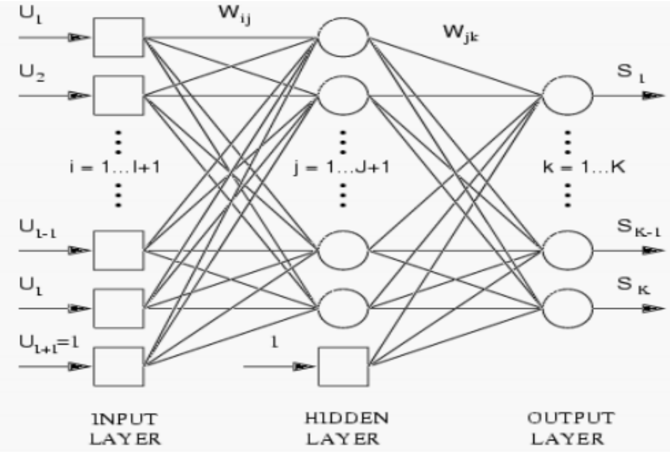


Fig. 17. Multi-Layer Neural Network

F. Deep Learning

In the field of machine learning, various predictive models exist, and they play an organic role. Among them, a model created by simulating a human neural network is called an artificial neural network, and the artificial neural network has a high degree of freedom in implementation, so that various types of structures can be built. A neural network made by stacking several layers corresponding to the depth of a neural network is called a deep neural network, and the method of learning is called deep learning. There are various types of models in deep learning models, but among them, models designed to process sequence-type input data are specialized for natural language processing, and there are RNN, LSTM, and GRU models.

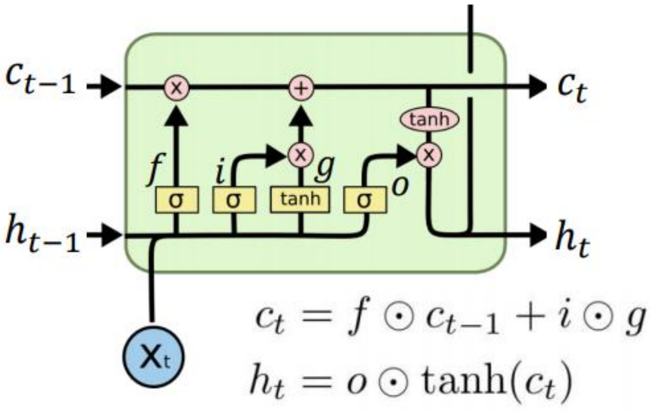

c_{t-1}
c_t
h_{t-1}
h_t
f
i
g
o
σ
σ
tanh
σ
tanh
X_t
$c_t = f \odot c_{t-1} + i \odot g$
$h_t = o \odot \tanh(c_t)$

Fig. 1. LSTM

- Long Short Term Memory(LSTM) : Language models used for natural language processing and language translation are mainly created to solve the problem of predicting the next word. In general, the probability of the next occurrence of a word is calculated from a given word sequence, or a middle empty word is predicted from both given words. Recurrent Neural Network (RNN) is a model in which the result value from the current neuron returns to the future node and affects it, and is used for sequenced events. The disadvantage of RNN is that the vanishing gradient problem occurs by using sigmoid or tanh as the activation function. LSTM tried to solve the vanishing gradient problem by adding the cell state to the RNN, forgetting unnecessary content and updating important information.
- Gated Recurrent Units(GRU) : GRU is a simplified version of LSTM and integrates the forget gate and input gate of the LSTM as an update gate to determine how much to update or how much to discard information in the previous state. The reset gate controls how much you want to remember the previous state. Compared to LSTM, the performance is similar, but the training time is faster.

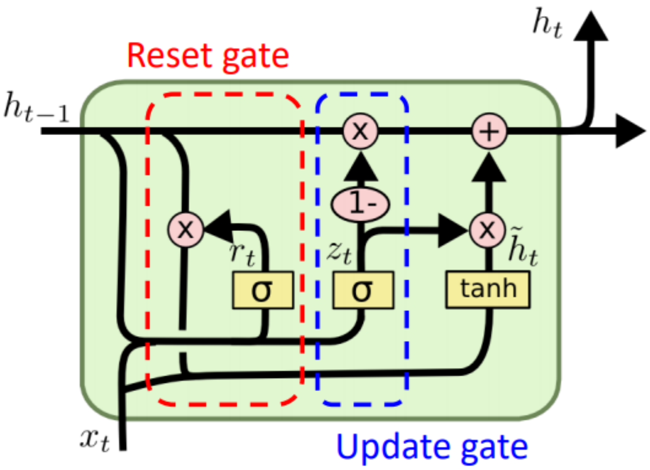
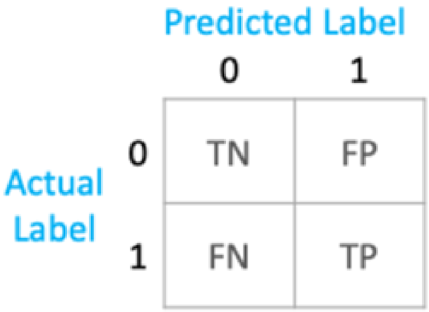


Fig. 2. GRU

G. Performance Measurement

Accuracy, precision, recall, sensitivity, F1 scores, ROC curves, and AUCs are commonly used as indicators for evaluating the performance of a classification model. Figure 19 is a confusion matrix used to compare and confirm the data prediction results and actual results. When the predicted value is positive, if the actual value is positive, it is expressed as True-Positive (TP). If the predicted value is positive but the actual value is negative, it is expressed as False-Positive (FP). When the predicted value is negative, if the actual value is positive, it is expressed as False-Negative (FN), and if the prediction is negative and the actual value is negative, it is expressed as True-Negative (TN).



		Predicted Label	
		0	1
Actual Label	0	TN	FP
	1	FN	TP

Fig. 3. Confusion Matrix

The accuracy is used to calculate the performance of the model, the formula is:

$$\boldsymbol{Accuracy = \frac{TP+TN}{TP+FN+FP+TN}} \tag{1}$$

Precision is the ratio of the samples (TP) that actually belong to the positive class among the samples (TP+FP) predicted to be a positive class, and a larger value means better performance. The formula is :

$$\boldsymbol{Precision = \frac{TP}{TP+FP}} \tag{2}$$

Recall is the ratio of the number of samples predicted to be positive class (TP) among samples (TP+FN) belonging to the actual positive class. Also, a larger value means better performance. The formula is :

$$\boldsymbol{Recall = \frac{TP}{TP+FN}} \tag{3}$$

The F1 score is an index that can explain how effective the model is by utilizing the precision and recall used to measure the model's performance. For both precision and recall, the higher the value, the better the performance. However, since precision and recall are complementary evaluation indicators, if one of them is forcibly increased, the other may fall. The F1 score was created for the purpose of using the two indicators harmoniously considering the trade-off relationship.

$$\boldsymbol{F1\ score = 2 * \frac{Precision*Recall}{Precision+Recall}} \tag{4}$$

The ROC curve and the AUC score based thereon are important indicators to measure the predictive performance of binary classification and are curves that show how the True Positive Rate (TPR) changes when the False Positive Rate (FPR) changes. TPR stands for recall and is called sensitivity. If FPR is taken as the x-axis and TPR as the y-axis, the change in TPR according to the change in FPR appears on the curve. And as an index corresponding to this sensitivity, there is a specificity called True Negative Rate (TNR). TNR = TN/(FP+TN), and the FPR seen earlier is expressed as FP/(FP+TN), that is, 1-TNR. The ROC curve calculates the change value of TPR while changing FPR from 0 to 1. To make FPR 0, set the threshold to 1, and if we set the threshold to 0, FPR becomes 1. The ROC curve is to obtain the FPR by changing the threshold value from 1 to 0, and to obtain the TPR value according to the change in the FPR value. The Area Under Curve (AUC) value is the area under the ROC curve in Figure 20. Generally, the closer to 1, the better.

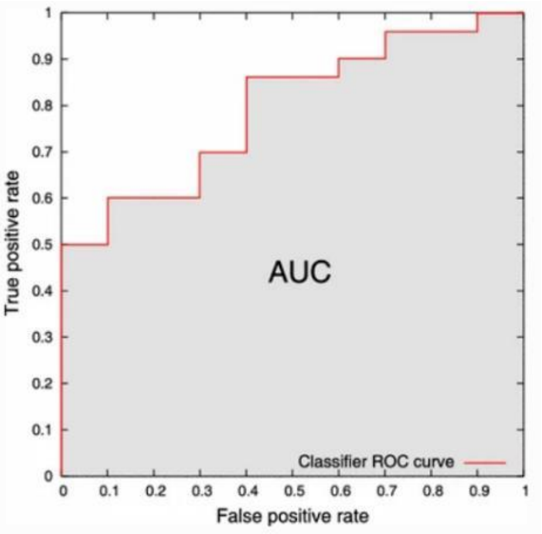


Fig. 4. AUC & ROC Curve

IV. Proposed Solution

We apply natural language processing (NLP) to extract features for machine learning. NLP applies to our pre-processing steps. The essential steps for NLP pre-processing consist of work word segmentation, tokenization, word stopping, word stemming, term frequency weighting, term frequency, and inverse document frequency weighting. We clean, analyze and lemmatize the data so that machine learning tasks could be performed. We employ NLP to remove punctuations, non-printable characters, number and special symbol characters that are non-English. We utilize tokenization, word stopping and lemmatized, term frequency inverse document frequency weighting (TFIDF). We use lemmatization over word stemming because lemmatization is generally more powerful. And the end product of our NLP processing is the TFIDF. It is the TFIDF that goes into our machine learning tasks.

Our machine learning and deep learning component consists of seven models; All models, where possible, are measured on accuracy, precision, recall, F-1, and AUC. The models are the crux of our fake and real news detection effort. We initially ran these models in their default settings from the Scikit-Learn library, but because the vanilla settings produce outstanding results we did not adjust these models in our project. Our seven models, starting from the base model, are described in the following section.

Logistic Regression (LR), our base model, is parametric classification model. LR has fixed inputs and output categorical and binary predictions. The inputs are independent variables and the output is a binary dependent variable. Instead of fitting to a straight line, like linear regression, LR fits to sigmoid, an S curve line to the data.

We employ the default LR from Scikit-Learn package. The default has the below tuning parameters:

class sklearn.linear_model.**LogisticRegression**(*penalty='l2'*, *, *dual=False*, *tol=0.0001*, *C=1.0*, *fit_intercept=True*, *intercept_scaling=1*, *class_weight=None*, *random_state=None*, *s* *olver='lbfgs', max_iter=100, multi_class='auto', verbose=0, warm_start=False, n_jobs=None, l1_ratio=None*)[1]

Multilayer Perceptron (NN) is the entry point into deep learning. NN begins with the input data passed through to the input layer having weights linked to neurons in the hidden layers. NN may have any number of hidden layers, which depend on the complexity of data. At each hidden layer the weights are connected to the next layer. The previous layer's outputs serve as the input to the next layer and so on. The final output layer is where the prediction is produced. NN's training on input-output pair set and learn about the correlations between the input-output set. The training comprises of modified the weights and biases to achieve the minimum error. The backward pass, backpropagation, makes changes to the weights and biases with respective to the error, which could be measured using various means.

We use the default NN from Scikit-Learn library. The default has the below tuning parameters:

class sklearn.neural_network.**MLPClassifier**(*hidden_layer_sizes=(100,), activation='relu', *, solver='adam', alpha=0.0001, batch_size='auto', learning_rate='constant', learning_rate_init=0.001, power_t=0.5, max_iter=200, shuffle=True, random_state=None, tol=0.0001, verbose=False, warm_start=False, momentum=0.9, nesterovs_momentum=True, early_stopping=False, validation_fraction=0.1, beta_1=0.9, beta_2=0.999, epsilon=1e-08, n_iter_no_change=10, max_fun=15000*) [2]

The random forest classifier (rfc) is a cluster of Decision Trees; it is considered one of the best algorithms for classification tasks. The premise of rfc is that a set of weak learners can produce a strong learner. Rfc gives weights to observations in each tree and utilizes a subset of predictors from the full set of predictors to build each tree; rfc does random samplings, which have the same size as the original dataset, with replacements, which means it has duplicates, for each decision tree. Using only a subset of predictors decorrelated the trees in the forest. Finally, rfc average the predictions of each tree to ascertain the final model.

We use the default rfc from Scikit-Learn library. The default has the below tuning parameters:

class sklearn.ensemble.**RandomForestClassifier**(*n_estimators=100, *, criterion='gini', max_depth=None, min_samples_split=2, min_samples_leaf=1, min_weight_fraction_leaf=0.0, max_features='sqrt', max_leaf_nodes=None, min_impurity_decrease=0.0, bootstrap=True, oob_score=False, n_jobs=None, random_state=None, verbose=0, warm_start=False, class_weight=None, ccp_alpha=0.0, max_samples=None*) [3]

Stochastic Gradient Descent Classifier (SGD) describes a linear classifier that employs stochastic gradient descent, an optimizer, which uses a single value of x to determine where the function y is the minimum.

We use the default SGD from Scikit-Learn library. The default has the below tuning parameters, which means it is the same a LinearSVC:

class sklearn.linear_model.**SGDClassifier**(*loss='hinge', *, penalty='l2', alpha=0.0001, l1_ratio=0.15, fit_intercept=True, max_iter=1000, tol=0.001, shuffle=True, verbose=0, epsilon=0.1, n_jobs=None, random_state=None, learning_rate='optimal', eta0=0.0, power_t=0.5, early_stopping=False, validation_fraction=0.1, n_iter_no_change=5, class_weight=None, warm_start=False, average=False*) [4]

LinearSVC (SVC) is special case of the Support Vector Machine (SVM) but SVC uses a linear kernel while SVM uses radial kernel. SVC also converges faster when there's a large dataset. SVC is based on statistical learning framework rather than probabilistic like LR. The goal of SVC learning process is to minimize the error rate. A kernel function maps data from input space to feature space to create linear decision-making functions to data in the leading space. The learning process of SVC is to find the line with the maximum margin that separates the data into two classifications and solve the problem of overfitting.

We use the default SVC from Scikit-Learn library. The default has the below tuning parameters:

class sklearn.svm.**LinearSVC**(*penalty='l2', loss='squared_hinge', *, dual=True, tol=0.0001, C=1.0, multi_class='ovr', fit_intercept=True, intercept_scaling=1, class_weight=None, verbose=0, random_state=None, max_iter=1000*) [5]

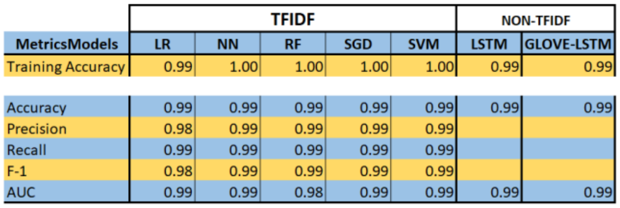
In the modeling stage, the machine learning model was trained using TFIDF, and the deep learning model was divided into two versions: a version that only performed basic tokenization and a version that applied data learned with gloves. And it involves testing and holdouts and splitting into prototypes (70,) and running the model described above. We initially only split the data into train and test with a 75% and 25% split. After producing outstanding high scores, we thought our models were wrong. We discussed what could go wrong and we took these additional verification steps.

The verifications include many different processes. We check for duplicates; there is only 6 duplicate records, which should not impact since we have over 40,000 records. We verify the code for errors such as using training dataset for testing and vice versa and for preprocessing errors. One of our team members verified the pre-processing code and reran them with the conclusion that there were no significant errors. Our dataset was balance between fake and real classes are about balance 47.7% real and 52.3% fake news.

Another verification is checking for data leakage. We thought it might be data leakage. This is true the first we ran the models because we turned all the documents into TFIDF and then split it into two sets of train and test. This had caused the data leakage because we were vectorizing, the tfidf, the entire dataset. Therefore, the training run already knew the features that will be present in the testing set. After recognizing our mistake, we divided our data into three parts before doing the TFIDF step and we saved the CountVectorizer and TfidfTransformer off to be used to perform TFDIF step for the test and holdout sets. We also used data segregation before tokenization and training for deep learning models. We employ the holdout set to do some sanity check on our models because holdout set is one of the ways to deal with data leakage.

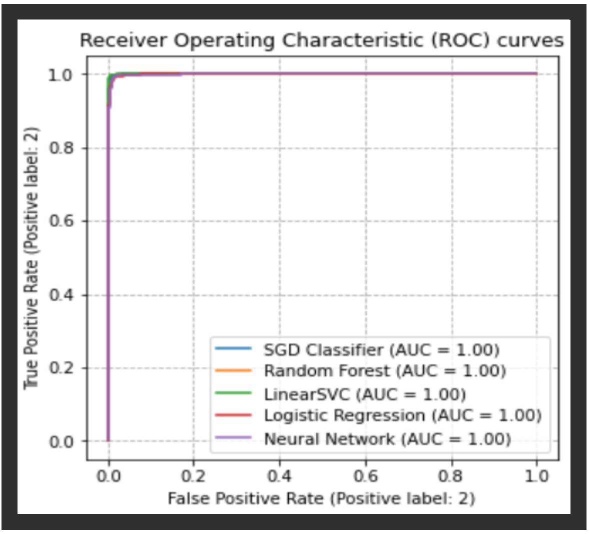
After making corrections and redoing the training and testing, our models' performance results are in the below chart.

The both training and the evaluation perform outstanding well at 100% or 99%. The base model LR performs at 99% during both training and testing.



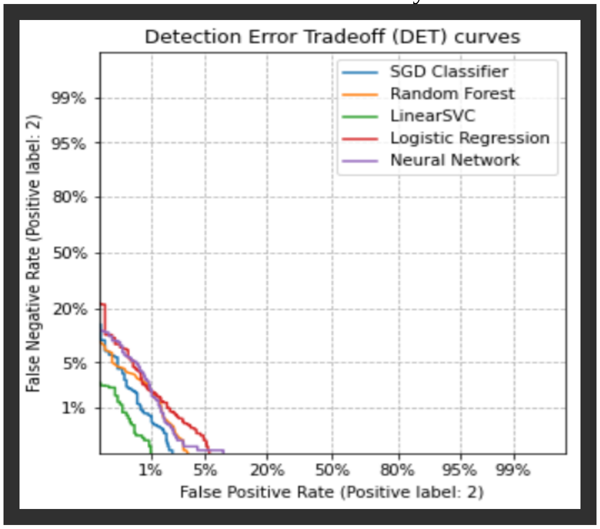
	TFIDF					NON-TFIDF	
MetricsModels	LR	NN	RF	SGD	SVM	LSTM	GLOVE-LSTM
Training Accuracy	0.99	1.00	1.00	1.00	1.00	0.99	0.99
Accuracy	0.99	0.99	0.99	0.99	0.99	0.99	0.99
Precision	0.98	0.99	0.99	0.99	0.99		
Recall	0.99	0.99	0.99	0.99	0.99		
F-1	0.98	0.99	0.99	0.99	0.99		
AUC	0.99	0.99	0.98	0.99	0.99	0.99	0.99

How can we tell which of the models performed the best since the evening our base model LR does equally well? We review the AUC plot and it was not able to distinguish the best performing model. They all appear to be one and we cannot tell which one does better based on the top left corner, although the linear model appears to lead the pack. But the indicator is so slight that we need to look at another tool.

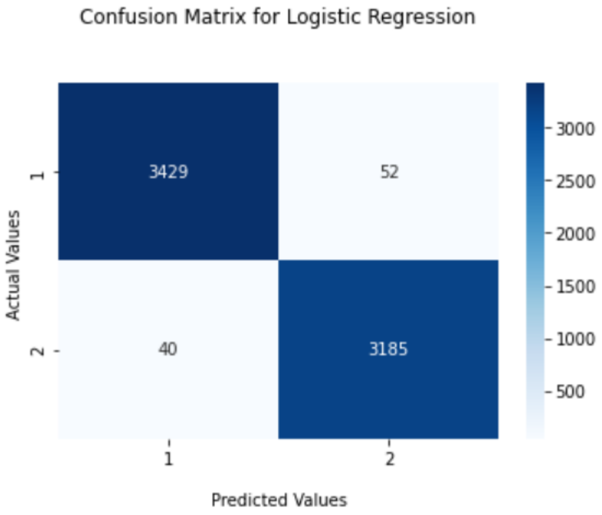


Both the AUC and the Detection Error Trade Off plot (DET) only two of the four rates based on the confusion the matrix. The four rates the true positive rate (TPR), false positive rate (FPR), false negative rate (FNR) and true negative rate (TNR). They both only plot the FPR on the x-axis and FNR on the y-axis. “The false positive rate is calculated as the ratio between the number of negative events wrongly categorized as positive and the total number of actual negative events. The false positive rate usually refers to the expectancy of the false positive ratio, while a false negative is the opposite error, where the test result incorrectly indicates the absence of a condition when it is actually present. [6]” The Detection Error Trade Off plot clearly shows that the LinearSVC performs the best with error varying between 1% for False Positive Rate and 2.5% for False Negative Rate. The SGD model, although the default settings resembles the LinearSVC, but because the SGDClassifier has other parameters that are different from LinearSVC the performance varies between 2.5% for False Positive Rate and 12% for False Negative Rate.

What are the best and worst performing model? The LR turns out to be the worst performing model. It varies between 6% for False Positive Rate and 21% for False Negative Rate. These numbers seem to imply that, therefore, the accuracy or AUC cannot be 99%--it was an implication raised in our final presentation. What do these numbers really mean?



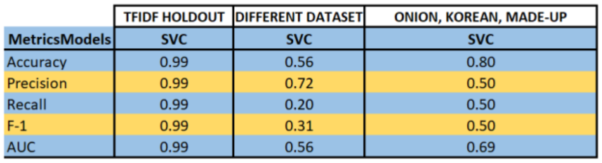
We review the confusion matrix result for LR. The numbers for the LR from the confusion matrix are below:



The calculated FPR and FNR based on the numbers from the confusion matrix produce FPR of 1% and FNR 1.6%. Therefore, FPR is not 6% as suggested by Detection Error Trade Off plot. Likewise, the FNR is not 21% either. What do these numbers mean? The best research I can find is the Detection Error Trade Off is scaled non-linearly to **accentuate** the differences in the top left corner of the AUC so that it would be easier to identify the better performing models. Nielson states, “The x- and y-axes are scaled non-linearly by their standard normal deviates (or just by logarithmic transformation), yielding tradeoff curves that are more linear than ROC curves, and use most of the image area to highlight the differences of importance in the critical operating region [7].” The DET plot is like doing log transformation on a data but to actually understand the result of using the scaled data in the model in its original scale context we need to reverse the process. In fact, the DET documentation says DET is scaled by scipy.stats.norm. Therefore, FPR or FNR rate of 6% or 21% in DET plot is scaled and it needs to be reversed to come back to original error percent. We also manually calculate the LR accuracy, which is (TP+TN)/ (TP+TN+FP+FN), which comes to 98.6%.

Therefore, based on the DET analysis, the best performing model is the linear SVC.

After testing the model and selection the best model, we also use the holdout set to do a sanity check of the linear SVC model. In addition, we also apply data from different dataset to our best performing model. The holdout set performance results are below:



	TFIDF HOLDOUT	DIFFERENT DATASET	ONION, KOREAN, MADE-UP
MetricsModels	SVC	SVC	SVC
Accuracy	0.99	0.56	0.80
Precision	0.99	0.72	0.50
Recall	0.99	0.20	0.50
F-1	0.99	0.31	0.50
AUC	0.99	0.56	0.69

The table shows the holdout set performs as well as our test set. But when the model is applied to entirely new and different dataset with about 8000 samples with only a statement for each sample rather than the entire article like our dataset, the score drops to 56 for both accuracy and AUC. But when we applied the model to another set of data that have the entire article from the Onion, Korean site and totally fabricated by us, the accuracy and AUC go up to 80 and 69, respectively.

The difference in the performance results between our dataset and new datasets may be attributed to various reasons. First, our data has politically news as its majority. Second, our data covers only one year of news from December 2016 to December 2017. Third, full-article dataset is hard to come be and the Different Dataset text length is only one sentence long, while our data is full articles or series of tweets that make up the document. In the Onion, Korean and Made-Up we show that full article document improves the linear SVC score by 13 and 24 points for AUC and accuracy, respectively, even though they were recently written articles.

Therefore, our linear SVC is able to predict fake and real news very well at 99% accuracy and AUC with a 95 percent confidence interval between 99 and 100. The model takes about 400 milliseconds to run. Its ability to detect fake and real news and quick run time makes linear SVC a very good model to use in an online application where fake and real news could be verified.

V. Proposed Solution

Based on our model result, substantive solutions has been proposed in which our model can contribute to healthy news consumption. One of the possible ways to apply our model is to deploy it in a web application.

However, at the same time, the implementation of the application suggested, is also expected to cause a number of problems; for example, when the right to use a fake news search model is given to the country, the possibility that the government can abuse it, or manipulate and censor articles under the leadership of the government. In other words, it leads to another question, "What happens when the home government is the purveyor of disinformation?". As there is an additional need for supportive solutions that can deal with its side effects also, laws to support identification of the news are also suggested.

A. *Web Application*

Proposed solution to prevent diffusion of fake news, is to automatically classify the truth of news that appears on search engines such as Google, Yahoo, and Naver. After that, mark reliable news or information as government-certified on the article of those search engine sites like putting a verification mark on a celebrity's account on Instagram or Facebook.

Furthermore, an online application can be created for users or policy makers who want to use our model to check the authenticity of the news. Our prototype is a news detection system that people can use freely; If user paste the news articles text into the search box, it offers a classification result of news with a confidence interval.

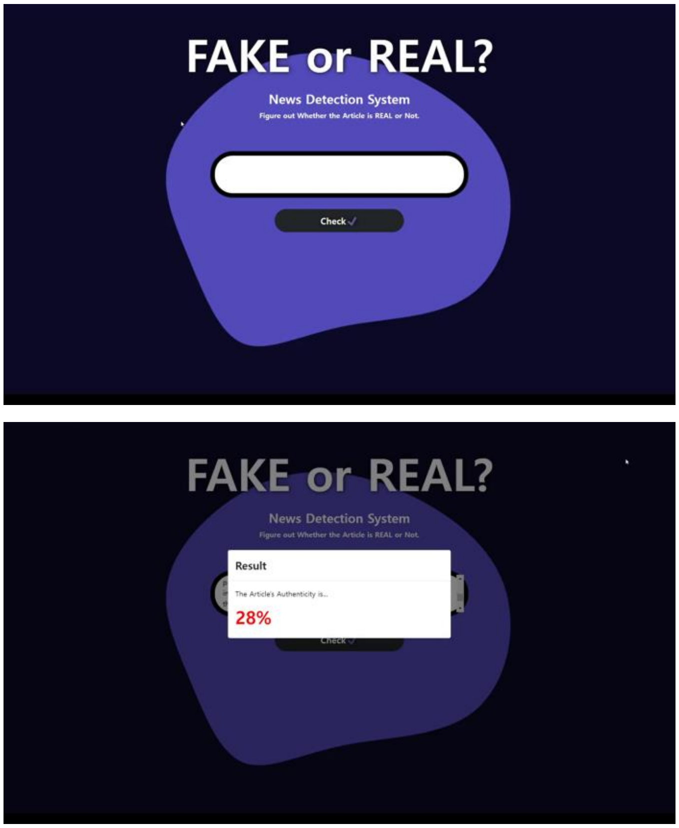

FAKE or REAL?
News Detection System
Figure out Whether the Article is REAL or Not.
Check ✓
FAKE or REAL?
News Detection System
Figure out Whether the Article is REAL or Not.
Result
The Article's Authenticity is...
28%
Check ✓

Fig. 5. Web Application

B. *Laws to Support Identification*

For appropriate operation of the prototypes presented above, three major types of legislation will be needed. Below is a list of the types of legislation deemed necessary.

Law Required for Publishers: Online publishers should be licensed by national licensing agency like the Federal Aviation Administration or Federal Communication Commission which aviation and communication with the United States. Also, the law requires publishers to provide link to News Verification Web Application so that readers can easily verify what they are reading. It can be as specific as stating all internet publishers who make more than 100,000 worth of money or more than 1000 hits per hour on their page or have more than 1000 followers to provide at least one link to such applications to verify their published articles in a prominent location on the page.

Laws Allowing Storage of Articles: It allows storage of news articles for ONLY disinformation detection applications. Its purpose is to help preventing redundant fetches for the same articles and allows detection companies to crawl and store news articles for only the explicit purpose of disinformation detection. Laws allowing disinformation detecting applications to by-pass copyright laws to allow the news articles to be stored and used for only disinformation detection, can be another option.

Laws Required for Social Media Companies: It requires social media companies to prominently display a seal in the articles fetched that are of foreign origin. This provides transparency and authentication of the news and tag news propaganda by foreign companies or government.

In addition, to prevent the model being abused under the leadership of the government, a law containing the duty for mandatory disclosure of the model operation process, should also be prepared.

VI. Conclusion

The development of Internet technology has brought many conveniences to humans. Among them, the characteristic that desired information can be easily shared and used irrespective of time and place is a great advantage. However, certain individuals or groups are exploiting these characteristics to gain benefits, and the spread of false information is exacerbating global chaos beyond the scope of individuals or groups. For example, in the 2016 US presidential election, disinformation that Hillary Clinton sold weapons to IS terrorist groups or that Pope Francis supported Trump was widespread. In the 2022 South Korean presidential election, false information about the candidates of each party was spread, and in the Russian-Ukrainian war, false information was spread that Russia did not attack Ukraine. And many people invested with false information about Bitcoin and suffered huge losses. Disinformation disrupts and harms many people across politics, society and economy, and a global effort is needed to correct it. Our team used a data-driven approach to classify false information, and we utilized the news and SNS data sets of the Kaggle site to carry out the project. In order to use the unstructured data composed of text data as an input value of the classification model, unlike the structured data composed of numerical data, various pre-processing processes mentioned above were performed. For the five machine learning models, TF-IDF, which indicates the importance of words in a document, was used. TF-IDF could not be applied to LSTM and GRU models because location information of words and order information of sentences were not reflected in the document. For the LSTM and GRU models, the general tokenization method and the pre-trained data for the word glove were used. The performance of machine learning models and deep learning models showed 99% accuracy for our data. We questioned the high level of accuracy and decided that additional verification of the data was necessary. We separated the data and tested again, and the same accuracy was still output for our data set. However, because the project period was short and the data collected by our team was collected for a short period of time, not the data collected over a long period, only the authenticity of a specific issue in that year could be accurately predicted and other data, the performance was poor. However, if we collect a variety of data over a long period of time, and can invest more time in model optimization, we think we can achieve a sufficient performance improvement. And if the web application we proposed in this project is applied to a well-developed model, We expect that government agencies can easily mitigate the spread of fake news and determine the authenticity of articles, thereby reducing citizens and the country from falling into confusion.

References

[1] Mahir, E. M., Akhter, S., & Huq, M. R. (2019, June). Detecting fake news using machine learning and deep learning algorithms. In *2019 7th International Conference on Smart Computing & Communications (ICSCC)* (pp. 1-5). IEEE.

[2] Buntain, C., & Golbeck, J. (2017, November). Automatically identifying fake news in popular twitter threads. In *2017 IEEE International Conference on Smart Cloud (SmartCloud)* (pp. 208-215). IEEE.

[3] Wani, A., Joshi, I., Khandve, S., Wagh, V., & Joshi, R. (2021, February). Evaluating deep learning approaches for covid19 fake news detection. In *International Workshop on Combating On line Ho st ile Posts in Regional Languages dur ing Emerge ncy Si tuation* (pp. 153-163). Springer, Cham.

[4] Shim, J. S., Lee, Y., & Ahn, H. (2021). A link2vec-based fake news detection model using web search results. *Expert Systems with Applications, 184*, 115491.

[5] Abedalla, A., Al-Sadi, A., & Abdullah, M. (2019, October). A closer look at fake news detection: A deep learning perspective. In *Proceedings of the 2019 3rd International Conference on Advances in Artificial Intelligence* (pp. 24-28).

[6] Nasir, J. A., Khan, O. S., & Varlamis, I. (2021). Fake news detection: A hybrid CNN-RNN based deep learning approach. *International Journal of Information Management Data Insights*, *1*(1), 100007.

[7] Girgis, S., Amer, E., & Gadallah, M. (2018, December). Deep learning algorithms for detecting fake news in online text. In *2018 13th International Conference on Computer Engineering and Systems (ICCES)* (pp. 93-97). IEEE.

[8] Kumar, S., Asthana, R., Upadhyay, S., Upreti, N., & Akbar, M. (2020). Fake news detection using deep learning models: A novel approach. *Transactions on Emerging Telecommunications Technologies*, *31*(2), e3767.

[9] Thota, A., Tilak, P., Ahluwalia, S., & Lohia, N. (2018). Fake news detection: a deep learning approach. *SMU Data Science Review, 1*(3), 10.

[10] Reis, J. C., Correia, A., Murai, F., Veloso, A., & Benevenuto, F. (2019). Supervised learning for fake news detection. *IEEE Intelligent Systems, 34*(2), 76-81.

[11] Andrew Dawson, and Martin Innes (2019). "How Russia's Internet Research Agency built its disinformation campaign." *The Political Quarterly*.

[12] John D. Gallacher, Vlad Barash, Philip N. Howard, and John Kelly (2017, October). "Junk News on Military Affairs and National Security: Social Media Disinformation Campaigns Against US Military Personnel and Veterans." *COMPROP DATA MEMO*.

[13] Mohammad Hadi Goldani, Saeedeh Momtazi and Reza Safabakhsh (2021, March). "Detecting fake news with capsule neural networks." *Applied Soft Computing, Volume 101(106991), Elsevier*.

[14] Michał Choraś, Konstantinos Demestichas, Agata Giełczyk, Álvaro Herrero, Paweł Ksieniewicz, Konstantina Remoundou, Daniel Urda and Michał Woźniak (2021, March). "Advanced Machine Learning techniques for fake news (online disinformation) detection: A systematic mapping study." *Applied Soft Computing, Volume 101(107050), Elsevier*.

[15] Katerina Sedova, Christine McNeill, Aurora Johnson, Aditi Joshi, and Ido Wulkan (2021, December). "AI and the future of disinformation campaigns." *Center for Security and Emerging Technology*.

[16] Feyza Altunbey Ozbay, and Bilal Alatas (2020, February). “Fake news detection within online social media using supervised artificial intelligence algorithms.” *Physica A: Statistical Mechanics and its Applications, Volume 540(123174), Elsevier*.

[17] de Oliveira, Nicollas R., Pedro S. Pisa, Martin A. Lopez, Dianne S.V. de Medeiros, and Diogo M.F. Mattos. (2021, January). "Identifying Fake News on Social Networks Based on Natural Language Processing: Trends and Challenges." *Information 12, no. 1: 38*.

[18] Luis Vargas, Patrick Emami, and Patrick Traynor. (2020, November). “On the Detection of Disinformation Campaign Activity with Network Analysis.” *Cloud Computing Security Workshop (CCSW’20), Virtual Event, USA. ACM, New York, NY, USA*.

[19] Mehwish Nasim, Andrew Nguyen, Nick Lothian, Robert Cope, and Lewis Mitchell. (2018, April). “Real-time Detection of Content Polluters in Partially Observable Twitter Networks.” *WWW ’18 Companion: The 2018 Web Conference Companion, April 23–27, 2018, Lyon, France. ACM, New York, NY, USA.*

[20] Ujjwal Singh, Nishit Raghuvansi and Hind Dev (2021, April). “Detection of Fake News Using Machine Learning.” *Turkish Journal of Computing and Mathematics Education, Volume 12(6)*.

[21] Albahar, M., (2021, February) “A hybrid model for fake news detection: Leveraging news content and user comments in fake news.” *IET Inf. Secur. 2021;15:169–177*.

[22] Rohit Kumar Kaliyar, Anurag Goswami, and Pratik Narang (2021, January). “FakeBERT: Fake news detection in social media with a BERT-based deep learning approach.” *Multimedia Tools and Applications 80:11765–11788*.

[23] Meesad, P. (2021). Thai Fake News Detection Based on Information Retrieval, Natural Language Processing and Machine Learning. *SN Computer Science (2021) 2:425*.

[24] D’Ulizia, A., Caschera, M.C., Ferri, F. & Grifoni, F. (2021). Fake news detection: a survey of evaluation datasets. *Pee J Computer Science*, 7:e518 DOI 10.7717/peerj-cs.518.

[25] Shu, Kai and Mahudeswaran, Deepak and Wang, Suhang and Lee, Dongwon and Liu, Huan (2020, Nov 3). FakeNewsNet: A Data Repository with News Content, Social Context, and Spatiotemporal Information for Studying Fake News on Social Media. *arXiv:1809.01286*.

[26] Monti, F., Frasca, F., Eynard, D., Mannion, D., & Bronstein, M. M. (2019, Feb 10). Fake News Detection on Social Media using Geometric Deep Learning. *arXiv:1902.06673v1 [cs.SI]*.

[27] Stahl, K. (2018, May 15). Fake news detection in social media. *Journals*. Retrieved March 21, 2022.

[28] Guo, C., Cao, J., Zhang, X., Shu, K. & Yu, M.(2019, March). Exploiting Emotions for Fake News Detection on Social Media. Retrieved March 21, 2022.

[29] Lu, Y., & Le, C. (2020, Apr 24). GCAN: Graph-aware Co-Attention Networks for Explainable Fake News Detection on Social Media. *arXiv: 2004.11648v1 [cs.CL]*.

[30] Hamid, A., Sheikh, N., Said, N., Ahmad, K., Gul, A., Hassan, L. & Al-Fuqaha, A. (2020, Nov 30). Fake News Detection in Social Media using Graph Neural Networks and NLP Techniques: A COVID-19 Use-case. *arXiv: 2012.07517v1 [cs.CL]*.

[31] Das, S.D., Basak, A., & Dutta, S. (2021, Jan 21). A Heuristic-driven Ensemble Framework for COVID-19 Fake News Detection. *arXiv: 2101.03545v1 [cs.CL]*.

[32] Shu, K., Zhou, A., Wang, S., Zafarani, R. & Lui, H. (2019). The Role of User Profiles for Fake News Detection. *IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*.

[33] Khan, J. Y., Khondaker, M. T. I., Afroz, S., Uddin, G., & Iqbal, A. (2021). A benchmark study of machine learning models for online fake news detection. *Machine Learning with Applications*, *4*, 100032.

[34] Ahmad, I., Yousaf, M., Yousaf, S., & Ahmad, M. O. (2020). Fake news detection using machine learning ensemble methods. *Complexity*, *2020*.

[35] Murayama, T., Wakamiya, S., Aramaki, E., & Kobayashi, R. (2021). Modeling the spread of fake news on Twitter. *Plos one*, *16*(4), e0250419.

[36] Lakshmanarao, A., Swathi, Y., & Kiran, T. S. R. (2019). An effecient fake news detection system using machine learning. *International Journal of Innovative Technology and Exploring Engineering*, *8*(10), 3125-3129.

[37] Bangyal, W. H., Qasim, R., Ahmad, Z., Dar, H., Rukhsar, L., Aman, Z., & Ahmad, J. (2021). Detection of Fake News Text Classification on COVID-19 Using Deep Learning Approaches. *Computational and Mathematical Methods in Medicine*, *2021*.

[38] Khanam, Z., Alwasel, B. N., Sirafi, H., & Rashid, M. (2021, March). Fake news detection using machine learning approaches. In *IOP Conference Series: Materials Science and Engineering* (Vol. 1099, No. 1, p. 012040). IOP Publishing.

[39] Kulkarni, P., Karwande, S., Keskar, R., Kale, P., & Iyer, S. (2021). Fake News Detection using Machine Learning. In *ITM Web of Conferences* (Vol. 40, p. 03003). EDP Sciences.

[40] Englmeier, K. (2021). The role of text mining in mitigating the threats from fake news and misinformation in times of corona. *Procedia Computer Science*, *181*, 149-156.