Detecting Deception Using Machine Learning

Alberto Alejandro Ceballos Delgado
Cyber Forensics Intelligence Center
Department of Computer Science
Sam Houston State University
aac088@shsu.edu

William Bradly Glisson
Cyber Forensics Intelligence Center
Department of Computer Science
Sam Houston State University
glisson@shsu.edu

Narasimha Shashidhar
Department of Computer Science
Sam Houston State University
nks001@shsu.edu

J. Todd McDonald
Department of Computer Science
School of Computing
University of South Alabama
jtmcdonald@southalabama.edu

George Grispos
School of Interdisciplinary
Informatics
University of Nebraska at Omaha
ggrispos@unomaha.edu

Ryan Benton
Department of Computer Science
School of Computing
University of South Alabama
rbenton@southalabama.edu

Abstract

Today's digital society creates an environment potentially conducive to the exchange of deceptive information. The dissemination of misleading information can have severe consequences on society. This research investigates the possibility of using shared characteristics among reviews, news articles, and emails to detect deception in text-based communication using machine learning techniques. The experiment discussed in this paper examines the use of Bag of Words and Part of Speech tag features to detect deception on the aforementioned types of communication using Neural Networks, Support Vector Machine, Naïve Bayesian, Random Forest, Logistic Regression, and Decision Tree. The contribution of this paper is two-fold. First, it provides initial insight into the identification of text communication cues useful in detecting deception across different types of text-based communication. Second, it provides a foundation for future research involving the application of machine learning algorithms to detect deception on different types of text communication.

1. Introduction

The escalation of text-based communications in today's digitally dependent societies creates an atmosphere that is potentially conducive to the creation, modification, and exchange of deceptive information. Deceptive communication can be defined as communication that "tends or has power to cause someone to accept as true or valid what is false or invalid" according to Merriam-Webster [1]. These fraudulent communications constitute a security incident depending on the outcome of associated activities. Academic and industrial publications continue to indicate that security incidents plague organizations; that incident recognition is critical to response scenarios and that these issues continue to have a financial and legal impact on organizations [2-10]. Malicious forms of communication that organizations deal with range from phishing attacks, to bogus customer reviews, to fake news.

Phishing attacks send misleading, fraudulent, and malicious messages that appear to originate from a trustworthy source [11]. These types of attacks attempt to steal information and/or install malicious software on a targeted machine [11]. A report by Microsoft finds that phishing messages have increased two hundred and fifty (250) percent between January and December 2018 [12]. Furthermore, the same report found that attackers are using a variety of techniques to make their attacks increasingly polymorphic, such as changing the URL, domain, and IP address, which allows them to avoid detection software. The Microsoft report indicates that techniques such as domain spoofing, domain impersonation, user impersonation, and text lures are increasing in popularity among attackers. The report goes on to suggest that these techniques make it more challenging to detect phishing emails accurately.

A Phishlabs report demonstrates that dealing with phishing attacks is a global problem. The report states that worldwide phishing attacks grew forty point nine percent (40.9%) in 2018, with countries like Canada and Turkey seeing an increase of one hundred and seventy percent (170%) and nine hundred and five percent (905%) in phishing attacks, respectively [13].

URI: https://hdl.handle.net/10125/71478
978-0-9981331-4-0

According to the report, financial institutions are among the most popular targets as they account for almost thirty percent (30%) of all attacks in 2018. Successful attacks can prove devastating to the economy, as a report by IBM showed in 2019 when they reported that the United States lost an average of eighty point nineteen (80.19) million dollars to data breaches in 2018 [14].

Deceptive text is not only useful in phishing attacks, but it is also a viable tactic in the creation of fake customer reviews on Web sites. One article claims that out of forty-seven thousand eight hundred and forty-six (47,846) customer reviews of the first ten products listed in Amazon, two-thirds are potentially deceptive [15]. Furthermore, the authors assert that the deceptive reviews artificially inflated the positive reviews of the seller. The authors also claim that the removal of potentially fraudulent reviews negatively impacts a seller's account by dropping the seller's rating. The same article postulates that the rating inflation has created a black market, where users offer to increase a seller's reputation with positive reviews. These activities potentially damage trust in e-commerce sites like Amazon or eBay, since rating inflation may cause buyers to be unable to discern genuine buyer input from potential scammers.

In addition to phishing and fake customer reviews, deceptive communication can also impact news sources. A recent report indicates that the number of fake news reports rose by approximately three hundred and twelve point six percent (312.6%) during the last presidential election [16]. The American Society for the Advancement of Science also supports the idea that fake news is on the rise; they found that the number of fake news increased during presidential elections [17]. One of their sources [18] indicated that during the 2016 presidential election, the average American encountered between one and three fake news articles in the month before the election. Additionally, the authors of the article declare that misinformation can potentially lead to an increase in apathy, cynicism, and even encourage extremism [17].

Due to the large volume of text communications generated by news outlets, social media, reviewers, companies, and other entities, it is impractical to detect deception on each message manually. Therefore, the development and implementation of automated algorithms and solutions are required to address this problem. Current technologies identify deception based on a single type of text communication [19, 20]. Also, for some types of communication like fake news, detection relies on manual verification [16]. The escalation of fake communications, coupled with current detection capabilities, prompts the hypothesis that fake reviews, fake news, and fake emails share common characteristics that are useful for deception detection. This hypothesis prompts the following research questions.

- Can Part of Speech (POS) tags and Bag of Words (BOW) be used to detect deception on reviews, news articles, and emails?
- Is the identification of an individual or combined feature set useful information for detecting deception in text-based communication?
- Can K Nearest Neighbors, Decision Tree, Logistic Regression, Naïve Bayesian, Neural Networks, Random Forest, and Support Vector Machine be used to detect deceptive text communications?

The contribution of this paper is two-fold. First, it provides initial insight into the identification of text communication cues that are useful in detecting deception across a variety of text communications. Second, it provides a foundation for future research involving the application of machine learning algorithms to a variety of text-based communications to detect deception.

This structure for the balance of the paper as follows: Section II presents previous research in the area of deception detection. Section III presents the research methodology. Section IV examines the results and performance of machine learning algorithms. Section V concludes the study, along with proposing future areas of research.

2. Literature Review

The escalation of text-based communications is prompting both academics and practitioners to investigate approaches for detecting deception in a variety of contexts [19-26]. These approaches target individual datasets that include emails, news articles, product reviews, and statements.

Litvinova et al [27] developed a model to detect deception on written Russian narratives. The authors utilized a text corpus Russian Deception Bank. This corpus was launched in 2014 as part of corpus called RusPersonality. This dataset contains 226 truthful and deceptive narratives on the same topic. This dataset contains information about the authors such as gender, age, and psychological test results. The authors employed a Russian language dictionary along with a Linguistic Inquiry and Word Count (LIWC) software to extract their features. The authors used standard linguistic dimensions, psychological process dimensions, punctuation parameters, the 20 most frequent function words in Russian, demonstrative pronouns and adverbs, discourse markers, intensifiers and downtowners intens, Part of Speech pronouns, perception vocabulary, and emotional words as features. The researchers utilized a Rocchio classification model. The researchers report that the accuracy of their trained model depends on the gender of the author of the text. Litvinova et al [27] reports that their model has an accuracy of 73.3% for male authors and 63.3% for female authors.

Kleinberg et al [28] used Named Entity Recognition (NER), the automatic identification and extraction of information from text, to develop a model to detect deceptive communication. Their model is based on 3 theoretical principles: truth tellers provide more detailed accounts, truth tellers have more contextual references (specific person, location, and times), and deceivers tend to withhold verifiable information. They used a dataset of hotel reviews developed by Ott et al. They used spaCy and Stanford's NER, two NER feature extraction tools, to extract features to train their model. They also extracted features using a Lexicon Word Count (LIWC) approach and a sentence specificity approach. The researchers seek to determine if truthful statements contain a higher number of named entities than false statements. Researchers report that their model outperforms the lexicon and sentence specificity approach.

An et al [29] developed a model to detect deception using personality recognition features. The researchers used the Columbia X-Cultural Deception (CXD) corpus. This corpus contains deceptive and truthful English speech from native speakers of Standard American English (SAE) and Mandarin Chinese (MC). The dataset contains approximately 125 hours of speech. The data was collected via fake job interviews in which an interviewer asked questions to the interviewee about their resume. The interviewee was instructed to lie to specific questions. The interviewees were evaluated using a NEO-FFI (Five Factor) personality inventory and divided into two groups high and low. The interviews were transcribed with using Amazon Mechanical Turk workers. The researchers extracted acoustic-prosodic low-level descriptor features, word category features from LIWC, and word scores for pleasantness, activeness, and imagery. The researchers trained a multilayer perceptron (MLP), a Long-Short-Term memory classifier, and a hybrid of the both models. The researchers report that their model improved performance as much as 6%.

Mendels et al [30] used the Columbia X-Cultural Deception Corpus to develop a model to detect deception using lexical and acoustic features. For acoustic features they utilized acoustic-prosodic features like pitch, intensity, spectral, cepstral, duration, voice quality, spectral harmonicity, and psychoacoustic spectral sharpness. For lexical features they utilized N-grams and embeddings using GloVe. The researchers trained two baseline classifiers: a Logistic Regression classifier trained using N-grams features, and a Random Forest classifier using acoustic-prosodic features. For deep learning models they utilized a lexical bidirectional long short-term memory (BLSTM) classifier, a Mel-Frequency cepstral coefficients (MFCC) BLSTM classifier, a Deep Neural Network (DNN) classifier using openSMILE features and a hybrid model. The researchers report that their hybrid model achieved an F1-score of 63.9% and that their Random Forest model achieved a precision of 76.11%

Litvinova et al [31] developed a dataset of Russian written texts labeled with data about their authors. The dataset contains information like gender, age, personality, neuropsychological testing data, education level, and other data about their authors. The dataset was designed for authorship profiling, deception detection, authorship attribution, and others. The dataset contains over 1850 documents from 1145 respondents. To demonstrate their dataset they performed a series of classification tasks using the corpus. They classified gender using Part of Speech tags, syntactical parameters, derivative coefficients, and number of punctuation marks. They also determined personality traits using morphological and syntactical features.

Abu-Nimeh et al. [21] analyze the effects of Bag of Words (BOW) features and metadata on detecting phishing emails. Using a dataset of nearly two thousand and nine hundred (2,900) emails, word frequency, stop word count, word count, and subject information features were used to train Support Vector Machine, Neural Network, Random Forest, Logistic Regression, and Bayesian Additive Regression Tree classifiers. The results from this research show that the evaluated Bag of Words features was able to detect ninety-five-point eleven percent (95.11%) of the phishing emails in the dataset.

To examine the effect of structural attributes and style marker features on phishing email detection, Chandrasekara et al. [22] developed a dataset consisting of four hundred (400) emails, including two hundred (200) phishing emails. This dataset was used to extract structural features, including word count, character count, word frequency distribution, and function word count, which were then used to train a Support Vector Machine (SVM) classifier. The results from this study show that the SVM classifier can accurately detect ninety-five percent (95%) of phishing emails in the evaluated dataset.

While previous research has focused on detecting deception in detection in customer reviews, news articles, and emails using features for each type, minimal research investigates the identification of features common to all three forms of communication.

3. Methodology

To investigate the hypothesis that fake reviews, news, and emails share common characteristics that are useful for deception detection, a controlled experiment, as defined by Shadish et al. [32], was divided into four stages. These stages include data collection, dataset preparation, feature extraction, and the application of machine learning algorithms. All the code used in the data preparation, feature extraction, and training and testing of the models, as well as the datasets is available at the following link: https://gitlab.com/public-data1/deception-detection/-/tree/master.

3.1. Data collection

Fake reviews, emails, and news article datasets were collected to test the new model on these types of text communications. The fake reviews dataset utilized in this experiment is from Ott et al.'s [20, 25] work. Their dataset contains eight hundred (800) labeled hotel reviews, of which four hundred (400) are truthful reviews collected from TripAdvisor, and four hundred (400) are deceitful reviews developed by Amazon Mechanical Turk workers.

The fake news dataset utilized in this experiment was developed by combining two existing datasets. The first one contains Buzzfeed and PolitiFact news articles, and the second one contains news from ABC and AMT. The Buzzfeed and PolitiFact dataset is from Shu et al.'s work [33-35]. The Buzzfeed and PolitiFact dataset includes five hundred and forty (540) truthful and five hundred and forty (540) deceitful news articles. The fake news articles are from the PolitiFact Application Programming Interface (API), which uses a team of experts to verify the claims in news articles to determine truthfulness [26]. The ABC and AMT dataset is from Perez-Rosas et al.'s work [36]. This dataset contains ninety-one (91) truthful and ninety-one (91) deceitful news articles about diverse topics.

Perez-Rosas et al. [36] developed their dataset by combining two different datasets. The first dataset consisted of truthful reviews collected from several news sources such as ABCNews, CNN, USAToday, New York Times, Fox News, Bloomberg, and others. It also consisted of deceitful reviews acquired from Amazon Mechanical Turk workers. The second dataset consists of news articles by Entertainment Weekly, People Magazine, RadarOnline, and other sites. They then verified the claims on the articles in the second dataset by using GossipCop.com, to split the articles into legitimate and deceitful categories.

The fake emails dataset utilized in this experiment from Dragomir's work [37]. It contains six thousand seven hundred and forty-two (6,742) truthful emails and five thousand one hundred and fifty-eight (5,158) deceitful emails. The deceitful emails come from phished emails corpora, and truthful emails come from the Spam Assassin project. The deceitful emails consist mostly of Nigerian prince emails attempting to persuade the reader to send them large amounts of money. The truthful emails consist of publicly released emails by Hillary Clinton.

3.2. Dataset preparation

The fake news dataset consisted of multiple text files divided into real and fake folders. These files were compiled into a single CSV file for easier analysis using a Python script.

The news articles dataset contained some missing values that were denoted using "Website is down for maintenance" or empty rows. This dataset also contained some Unicode characters that could not be processed. These invalid values were removed using a Python script.

The email dataset consists of a CSV file with a text message and a real field. Initial analysis of the email dataset revealed that it contained many empty rows that needed to be removed. This analysis also found multiple rows with only hexadecimal characters in the text message field, which correspond to the email footer.

The analysis also found several email addresses in the text message, which needed to be removed for further processing. A script was developed using Google Script to remove emails that contain specific keywords. This cleansing process generated the final email dataset used in the experiment.

The next step in the dataset preparation process consists of combining the email, news, and reviews dataset into one. A python script was developed for this purpose. The dataset that results from this script is used through the experiment.

3.3. Feature extraction

Table 1: Speech Tags

Definition	POS Tag
Coordinating Conjunction	CC
Cardinal Digit	CD
Determiner	DT
Existential	EX
Foreign Word	FW
Preposition/subordinating conjunction	IN
Adjective “Big”	JJ
Adjective Comparative “Bigger”	JJR
Adjective Superlative “Biggest”	JJS
List Marker	LS
Modal	MD
Noun Singular	NN
Noun Plural	NNS
Noun Proper Singular	NNP
Noun Proper Plural	NNPS
Predeterminer	PDT
Possessive Ending	POS
Personal Pronoun	PRP
Possessive Pronoun	PRP$
Adverb	RB
Adverb Superlative	RBS
Adverb Comparative	RBR
Particle	RP
To go ‘to’ the store	TO
Interjection	UH
Verb	VB
Verb Past	VBD
Verb Past Participle	VBN
Verb Singular Present	VBP
Verb 3rd person singular present	VBZ
Wh-Determiner	WDT
Wh-Pronoun	WP
Possessive Wh-Pronoun	WP$
Wh-Adverb	WRB

The first step in feature extraction is the creation of a dictionary to identify typographical errors. A script was develop using Python’s Natural Language Token Kit (NLTK) [38, 39] to load a corpus containing a repository of English words. The script also uses Python’s NLTK library to load a list of known English stopwords. The script iterates over the combined dataset, stems each word it encounters, removes stop words, and stores the filtered words into a file. The resulting file is used a dictionary in this experiment.

The next step involves extracting features from the text message data. Extracted features are classified into two categories that include a Single Feature (SF) group and a Bag of Words (BOW) group.

The Single Feature group consists of counting the number of occurrences of each Part of Speech (POS) tag listed in Table 1, number of words, number of characters, typographical errors, number of sentences, the occurrence of each letter, and number of special characters in a message. The Bag of Words group consists of counting each word in each message. The Part of Speech tags listed come from Python’s Natural Language Token Kit documentation [38]. This feature extraction process is accomplished using a Python script and it generates a final file used as the final dataset throughout this experiment.

Initial analysis of the dataset generated on the previous step is performed to remove low variance features to reduce the feature space. Figure 1 shows a histogram displaying the number of occurrences of each Part of Speech tag feature. Part of Speech tag features whose frequency was more than eighty percent (80%) for a single value were determined to have low variance and were removed.

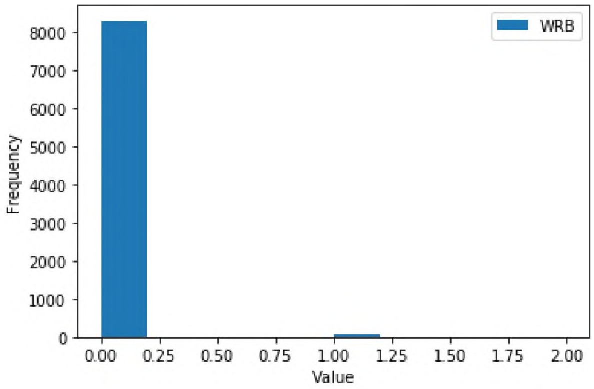

Figure 1: Feature Histogram

The dataset produced in the previous step was split into three (3) groups to evaluate the effects of each feature set on the model’s accuracy. The first dataset contains the Single Feature group only, the second dataset contains Bag of Words features only, and the third dataset includes both features.

3.4. Machine learning application

The machine learning algorithms selected were K Nearest Neighbors (KNN), Decision Tree (DT), Logistic Regression (LR), Naïve Bayesian (NB), Neural Networks (NN), Random Forest (RF), and Support Vector Machine (SVM). These algorithms were selected due to their accuracy in detecting deception based on literature [19-26]. Each model was tested for the most optimal hyperparameter and developed using Python's SciKit Learn library version 3.4 [40, 41].

The training process started by splitting the three datasets into two subgroups that consist of testing and training. Eighty percent (80%) of the data was allocated for training, and twenty percent (20%) was allocated for testing per the small amount of data available [42]. Then, the data was used to develop the models with the initial hyperparameters provided by Python's SciKit Learn library.

The accuracy of each model with their hyperparameters was recorded. To evaluate the model's accuracy, the hyperparameters were varied; the models were trained on the training dataset, tested on the testing dataset, and their accuracy was recorded for all combinations of hyperparameters. This process was used to determine the hyperparameters associated with the highest accuracy, which were used to train and test the models.

4. Results and Analysis

For K Nearest Neighbors, the model was trained for the first one thousand (1000) possible values of K. Figure 2 shows a plot K value vs. accuracy, and inspecting this graph reveals that the value of K that produced the highest accuracy was sixty (60).

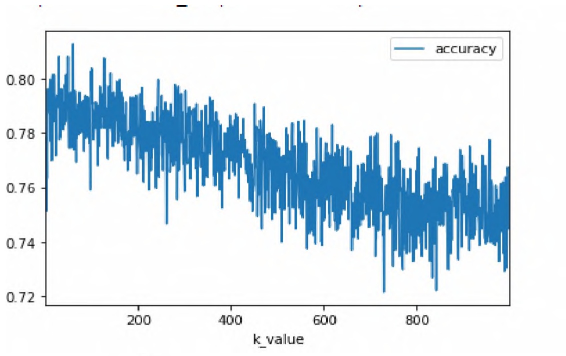

Figure 2: K value vs. accuracy

For Support Vector Machine, the model was trained with a Linear Regression kernel (SVMLR) and a Radial Basis Function kernel (SVMRBF) for each dataset. The accuracies for each model were recorded. Then the model's accuracy was compared, and the model with the kernel that performed better, on average, was selected. Figure 3 shows the accuracy of each model trained with the different datasets and shows that the model with the SVMRBF performed better than the SVMLR.

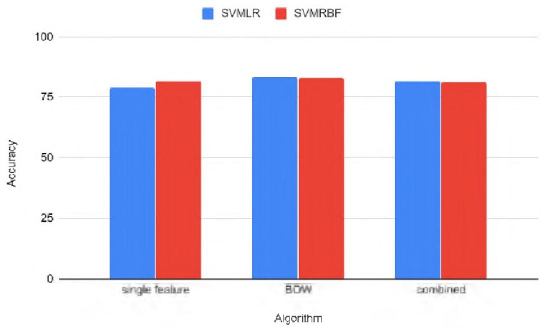

Figure 3: SVM Kernel Accuracy

The remaining models were trained with the default values provided by Python's Scientific Kit (SciKit) Learn library. For decision tree, the criterion used for feature selection is Gini impurity, the splitter used is best, the tree is expanded until all leaves are pure or contain less than 2 samples. For logistic regression it uses a ovr loss function, it ran for a maximum of 100 iterations, it uses an lbfgs solver, it uses an inverse of regularization strength of 1, and an l2 penalty. For naïve bayessian it uses a gaussian naïve bayes classifier. For neural network it uses a multilayered perceptron classifier with 100 neurons per layer, with a relu activation function, with an adam optimizer, with a constant learning rate, with 200 maximum iterations. For random forest it uses 100 trees in the forest, it uses a gini criterion for quality of a split, it expands all leaves until they are all pure or contain less than 2 samples.

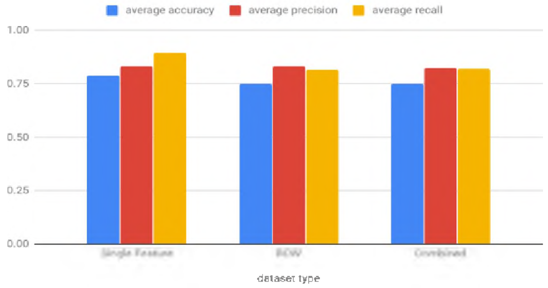

Figure 4: Average model accuracy, precision, and recall per dataset

The model's accuracy, precision, and recall were averaged for each dataset to study its effects. Figure 4 presents the average accuracy, recall and precision of the models for each dataset. Models trained with single features dataset have an average accuracy of 78.88%, an average precision of 83.05%, and an average recall of 89.30%. Models trained with Bag of Word features have an average accuracy of 74.97%, an average precision of 82.98%, and an average recall of 81.64%. Finally models trained with both featureset combined have an average accuracy of 75.01%, an average precision of 82.49%, and an average recall of 81.95%. This graph suggests that single features provide more useful information than Bag of Words feature or both combined, as suggested by Ott et al. [9] research.

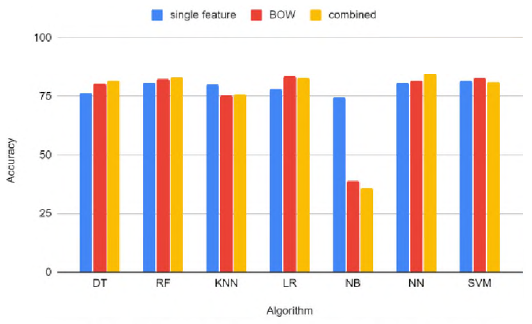

Figure 5: Model's accuracy per dataset

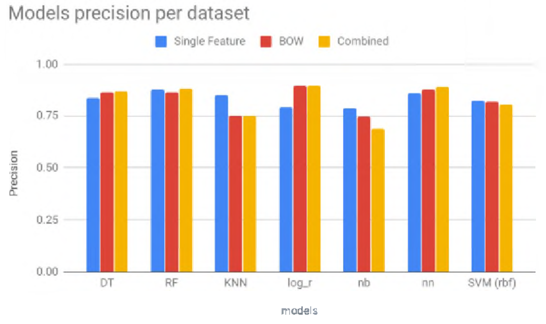

Figure 6: Model's precision per dataset

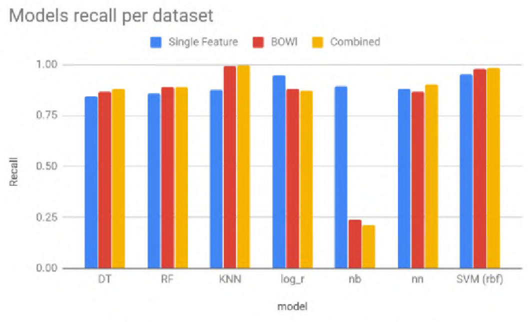

Figure 7: Model's recall per dataset

Figure 5, 6, and 7 shows the model's accuracy, precision, and recall by each dataset. The Neural Network model performed on average better than all other models with 82.35% accuracy, 87.42% precision, and 88.28% recall. Random Forest closely followed with 81.97% accuracy, 87.29% precision, and 87.90% recall. Support Vector Machine has an average accuracy of 81.87%, an average precision of 81.55%, and an average recall of 97.13%.This suggests that algorithms that can process a large number of features perform better at this task than other algorithms.

The Naïve Bayesian model performed the worst with Bag of Words with an average accuracy of 49.73%, an average precision of 73.96%, and an average recall of 44.91%. The combined dataset accuracy is 36.86%, which is possibly due to the large number of features to process or potential noise features.

5. Limitations

One of the limitations of this research is the lack of a large and reliable corpus of deceptive communication. It is difficult to develop a labelled dataset of certain types of communication like false reviews because they might suffer from biases. For example, the deceptive hotel review dataset was developed using Amazon Mechanical Turk workers [36]. However, some researchers [19] have argued that the reviews developed by the AMT workers may not properly emulate real fake reviews because they were paid to develop them. Furthermore, other researchers [19] have utilized datasets prelabeled by existing software like Yelp's review filtering software. The accuracy of models trained on this dataset rely on the assumption that the review filtering software algorithm is reliable [19]. However, since the dataset was labelled using the original review filtering software, any biases with the original filtering software will transfer to models trained on that dataset. This research utilized Perez-Rosas et al [36] false review dataset and thus the models trained on that dataset might not accurately reflect false reviews.

Another limitation is the size of the datasets used. The length of reviews in general tend to be smaller than news articles. Furthermore, the length of deceptive emails in general are larger than the real emails since the dataset used for false emails includes a large number of short emails that are responses to previous emails like "FYI" or "Okay". While the larger emails constitude mostly phishing emails or scams like the Nigerian prince scam. Therefore, the model's accuracy might be influenced by the length of each communication. Where longer text messages might have a higher likelihood to be identified as false.

A third limitation is the lack of variety on the datasets. All of the datasets come from English written documents. Therefore, the models might be biased and not perform well on documents written in other languages.

6. Conclusions and future work

The research presented in this paper aimed to investigate the use of shared characteristics between news articles, product reviews, and emails to detect deception. To accomplish this goal, a dataset of reviews, news articles, and emails was collected from different sources. The collected datasets were cleaned and merged to create a large dataset of text-based messages. Part of Speech (POS) and Bag of Word (BOW) features were extracted from this newly created dataset using Python's scripts and libraries. These features were divided into groups: one with only Part of Speech tag features, one with only Bag of Word features, and one with both combined. These features were used to classify and train different machine learning models. Each model had its training and testing accuracy recorded and analyzed.

The results from this research suggest that Part of Speech (POS) tags and Bag of Words (BOW) can be used to detect deception across different types of text communication. The average accuracy of the machine learning models trained with these features suggests that the models can detect deception on different text-based communication.

The identification of individual and combined features provide useful information for deception detection according to the results from this research. The average accuracy of models trained with a single feature was higher than those trained with combined features. Furthermore, using group analysis the average accuracy of models trained with Part of Speech tags features is greater than those with Bag of Words features, which suggests that Part of Speech tags provide more useful information than Bag of Words features.

The results from this research suggest that Neural Networks, K-Nearest Neighbors, Support Vector Machine, Decision Tree, Logistic Regression, Naïve Bayesian, and Random Forest can be used to detect deceptive communication across different types of text-based messages.

Reviews, news articles, and emails share common characteristics that can be used to detect deception according to the results from this research. Bag of Words and Part of Speech tags features were extracted from each type of text-based communication and used to train different machine learning models. The models were able to accurately detect deception on the aforementioned types of text-based communication with an average accuracy of seventy percent.

Future work focuses on the development of a large publicly available dataset of verified deceitful and truthful reviews, news articles, and emails. This dataset could be developed in cooperation with news verification organizations, popular email providers, and local review organizations like Yelp. The methodology to determine the truthfulness of an article should be transparent and public for reliability. This dataset would allow further research on different machine learning technologies such as deep neural networks and Word2Vector for automatic deception detection. Future research also investigates the impact of different languages and cultural interpretations on deception detection algorithms. Furthermore, future work should evaluate the performance of the models discussed in this research paper on single datasets not just combined datasets.

10. References

[1] Merriam-Webster. *Deceptive adjective*. https://www.merriam-webster.com/dictionary/deceptive. Date of Last Access: 10/4/2020.

[2] Glisson, W.B., L.M. Glisson, and R. Welland. *Web Development Evolution: The Business Perspective on Security*. in *Thirty-Fifth Annual Western Decision Sciences Institute*. 2006. Hawaii: Western Decision Sciences Institute.

[3] Glisson, W.B., L.M. Glisson, and R. Welland. *Secure Web Application Development and Global Regulation*. in *The Second International Conference on Availability, Reliability and Security (ARES)* 2007. Vienna, Austria: IEEE.

[4] Glisson, W.B. and T. Storer. *Investigating Information Security Risks of Mobile Device Use Within Organizations* in *Americas Conference on Information Systems (AMCIS)*. 2013.

[5] Glisson, W.B. and R. Welland, *Web Engineering Security (WES) Methodology*. Communications of the Association for Information Systems, 2014. **34**: p. Article 71.

[6] Grispos, G. and K. Bastola. *Cyber Autopsies: The Integration of Digital Forensics into Medical Contexts*. in *2020 IEEE 33rd International Symposium on Computer-Based Medical Systems (CBMS)*. 2020. IEEE.

[7] Grispos, G., W.B. Glisson, D. Bourrie, T. Storer, and S. Miller. *Security Incident Recognition and Reporting (SIRR): An Industrial Perspective*. in *Twenty-third*

Americas Conference on Information Systems. 2017. Boston: Americas Conference on Information Systems.

[8] Grispos, G., W.B. Glisson, and T. Storer. *Rethinking Security Incident Response: The Integration of Agile Principles*. in *Americas Conference on Information Systems (AMCIS)*. 2014. Savannah, GA.

[9] Grispos, G., W.B. Glisson, and T. Storer. *Security Incident Response Criteria: A Practitioner's Perspective*. in *Americas Conference on Information Systems (AMCIS)*. 2015. Puerto Rico.

[10] Grispos, G., W.B. Glisson, and T. Storer. *How Good is Your Data? Investigating the Quality of Data Generated During Security Incident Response Investigations*. in *Proceedings of the 52nd Hawaii International Conference on System Sciences*. 2019. Hawaii.

[11] CISCO. *What Is Phishing?* https://www.cisco.com/c/en/us/products/security/email-security/what-is-phishing.html. Date of Last Access:

[12] Agrawal, A., D. Fantham, D. Ghosh, D. Kelley, E. Florio, E. Avena, A. Douglas, F.T. Seng, J. Trull, J. Borenstein, K. Selvaraj, K. Kaplinska, K. Laidler, M. Duncan, M. Simos, P. Henry, P. Pandey, R. Pliskin, R. McGee, S. Kathuria, S. Wacker, T. Ganacharya, V. Grebennikov, and Y. Zohar, *Microsoft security intelligence report*, in *Microsoft security intelligence report*. 2019. p. 20-24.

[13] *2019 Phishing Trends and Intelligence Report*. 2019, Phishlabs.

[14] *2019 Cost of a Data Breach*. 2019, IBM Security.

[15] Dwoskin, E. and C. Timberg, *How merchants use Facebook to flood Amazon with fake reviews*, in *The Washington Post*. 2018.

[16] *Cybersecurity Report*. 2019, DFNDR Lab.

[17] Lazer, D., M. Baum, Y. Benkler, A. Berinsky, K. Greenhill, F. Menczer, M. Metzger, B. Nyhan, G. Pennycook, and D. Rothschild, *The science of fake news*. Science, 2018. **359**(6380): p. 1094-1096.

[18] Allcott, H.G., Matthew, *Social Media and Fake News in the 2016 Election*. Journal of Economic Perspectives, 2017. **31**: p. 211-236.

[19] Mukherjee, A., V. Venkataraman, B. Liu, and N. Glance, *Fake review detection: Classification and analysis of real world and pseudoreviews*. UIC-CS-03-2013, Technical Report, 2013.

[20] Ott, M., Y. Choi, C. Cardie, and J. Hancock, *Finding deceptive opinion spam by any stretch of the imagination*. Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies, 2011. **1**: p. 309-319.

[21] Abu-Nimeh, S., D. Nappa, X. Wang, and S. Nair, *A comparison of machine learning techniques for phishing detection*. Proceedings of the anti-phishing working groups 2nd annual eCrime researchers summit, 2007: p. 60-69.

[22] Chandrasekaran, M., K. Narayanan, and S. Upadhyaya, *Phishing email detection based on structural properties*. NYS cyber security conference, 2006. **3**.

[23] Feng, S., R. Banerjee, and Y. Choi, *Syntatic stylometry for deception detection*. Proceedings of the 50th annual meeting of the association for computational linguistics: Short papers, 2012. **2**: p. 171-175.

[24] Fuller, C., D. Biros, and D. Delen, *An investigation of data and text mining methods for real world deception detection*. Expert systems with applications, 2011. **38**(7): p. 8392-8398.

[25] Ott, M., C. Cardie, and J. Hancock, *Negative deceptive opinion spam*. Proceedings of the 2013 conference of the north american chapter of the association for computational linguistics: human language technologies, 2013: p. 397-501.

[26] Wang, W., *"Liar, liar pants on fire": A new benchmark dataset for fake news detection*. arXiv preprint arXiv:1705.00648, 2017.

[27] Litvinova, O., P. Seredin, T. Litvinova, and J. Lyell. *Deception detection in russian texts*. in *Proceedings of the Student Research Workshop at the 15th Conference of the European Chapter of the Association for Computational Linguistics*. 2017.

[28] Kleinberg, B., M. Mozes, A. Arntz, and B. Verschuere, *Using named entities for computer-automated verbal deception detection*. Journal of forensic sciences, 2018. **63**(3): p. 714-723.

[29] An, G., S.I. Levitan, J. Hirschberg, and R. Levitan. *Deep Personality Recognition for Deception Detection*. in *INTERSPEECH*. 2018.

[30] Mendels, G., S.I. Levitan, K.-Z. Lee, and J. Hirschberg. *Hybrid Acoustic-Lexical Deep Learning Approach for Deception Detection*. in *INTERSPEECH*. 2017.

[31] Litvinova, T., O. Litvinlova, O. Zagorovskaya, P. Seredin, A. Sboev, and O. Romanchenko. *" Ruspersonality": A Russian corpus for authorship profiling and deception detection*. in *2016 International FRUCT Conference on Intelligence, Social Media and Web (ISMW FRUCT)*. 2016. IEEE.

[32] Shadish, W.R., T.D. Cook, and D.T. Campbell, *Experimental and quasi-experimental designs for generalized causal inference/William R. Shedish, Thomas D. Cook, Donald T. Campbell*. 2002: Boston: Houghton Mifflin.

[33] Shu, K., D. Mahudeswaran, S. Wang, D. Lee, and H. Liu, *FakeNewsNet: A Data Repository with News Content, Social Context, and Dynamic Information for Studying Fake News on Social Media.* arXiv preprint arXiv:1809.01286, 2018.

[34] Shu, K., A. Sliva, S. Wang, J. Tang, and H. Liu, *Fake News Detection on Social Media: A Data Mining Perspective.* ACM SIGKDD Explorations Newsletter, 2017. **19**(1): p. 22-36.

[35] Shu, K., S. Wang, and H. Liu, *Exploiting Tri-Relationship for Fake News Detection.* arXiv preprint arXiv:1712.07709, 2017.

[36] Perez-Rosas, V., B. Kleinberg, A. Lefevre, and R. Mihalcea, *Automatic detection of fake news.* arXiv preprint arXiv:1708.07104, 2017.

[37] Dragomir, R. *ACL Data and Code Repository*. 2018.

[38] Bird, S., E. Klein, and E. Loper, *Natural language processing with Python: analyzing text with the natural language toolkit*. 2009: O'Reilly Media, Inc.

[39] NLTK, *NLTK*.

[40] Buitinck, L., G. Louppe, M. Blondel, F. Pedregosa, A. Mueller, O. Grisel, V. Niculae, P. Prettenhofer, A. Gramfort, and J. Grobler, *API design for machine learning software: experiences from the scikit-learn project.* arXiv preprint arXiv:1309.0238, 2013.

[41] Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, and V. Dubourg, *Scikit-learn: Machine learning in Python.* the Journal of machine Learning research, 2011. **12**: p. 2825-2830.

[42] Tarang, S. *About Train, Validation, Test Sets in Machine Learning.* 2019. https://towardsdatascience.com/train-validation-and-test-sets-72cb40cba9e7. Date of Last Access: August 20, 2019.