2023 18th International Workshop on Semantic and Social Media Adaptation & Personalization (SMAP) | 979-8-3503-2771-7/23/$31.00 ©2023 IEEE | DOI: 10.1109/SMAP59435.2023.10255184

A Comparative Analysis of Tweet Analysis Algorithms using Natural Language Processing and Machine Learning Models

Andreas Kanavos*, Nikos Antonopoulos†, Ioannis Karamitsos‡ and Phivos Mylonas§
*Department of Informatics, Ionian University, Corfu, Greece
akanavos@ionio.gr
†Department of Digital Media and Communication, NeMeCULAB, Ionian University, Argostoli, Greece
nikos@antonopoulos.info
‡Department of Graduate and Research
Rochester Institute of Technology, Dubai, UAE
ixkcad1@rit.edu
§Department of Informatics and Computer Engineering, University of West Attica, Athens, Greece
mylonasf@uniwa.gr

***Abstract*—Online Social Networks (OSNs) have become integral platforms for information sharing, attracting both legitimate users and spammers. Detecting and mitigating spam within these networks pose significant challenges due to the dynamic nature of content and user behavior. In this paper, we present a comprehensive comparative analysis of algorithms for tweet analysis, focusing on Natural Language Processing (NLP) and Machine Learning (ML) models. We evaluate these algorithms through sentiment analysis and multiple attribute analysis, utilizing diverse methodologies and datasets. Our study explores feed-forward neural networks, Bayesian classifiers, and transformer-based models for NLP tasks, achieving high prediction accuracy and insightful metrics such as precision, recall, and F1 score. Furthermore, we delve into multiple attribute analysis using Random Forest, Logistic Regression, and Gradient Boosting algorithms. Through a systematic exploration of various approaches, this work contributes to a deeper understanding of spam detection and sentiment analysis within the context of OSNs, paving the way for enhanced social network security and content analysis.**

***Index Terms*—Online Social Networks, Tweet Analysis, Natural Language Processing, Machine Learning, Spam Detection, Transformer Models**

I. Introduction

In recent years, the exponential growth of social media platforms has led to an overwhelming volume of user-generated content, presenting both opportunities and challenges for extracting meaningful insights. Among the most widely used platforms, Twitter serves as a microblogging platform where users share their thoughts, opinions, and information in a concise format. As a result, Twitter has become a valuable source of data for various applications, ranging from sentiment analysis and user profiling to spam detection and content recommendation [3], [13]. To unlock the potential of this data, advanced algorithms rooted in NLP and ML are essential for comprehensive tweet analysis [10].

Presently, OSNs have gained immense traction, serving as prominent platforms for idea and thought exchange [9]. Renowned OSNs include Twitter, Facebook, MySpace, and LinkedIn, among others. However, the surge in popularity of these platforms has been accompanied by a corresponding rise in malicious activities targeting them. Given the extensive user base of these networks, not all users can be authenticated as legitimate. Also, it is observed that a multitude of illegitimate or spam accounts populate each of these OSNs. The realm of sentiment analysis draws data from online social media, where users generate an ever-increasing volume of information [2], [8]. As a result, this influx of data necessitates adopting a big data approach, considering the challenges associated with efficient data storage, access, processing, and result reliability [4], [5].

This paper delves into the intricate realm of tweet analysis by conducting a thorough comparative investigation of diverse algorithms utilizing NLP and ML models. The study capitalizes on the multifaceted nature of tweets, exploring both Natural Language Analysis and Multiple Attribute Analysis. Natural Language Analysis entails the application of algorithms to understand the textual content of tweets, enabling sentiment classification, spam detection, and other forms of content understanding. On the other hand, Multiple Attribute Analysis focuses on uncovering patterns in various attributes associated with tweets, which can encompass metadata, user interactions, and contextual cues.

The remainder of this paper is organized as follows: Section II presents an overview of related work in the field, whereas Section III provides insights into the fundamental concepts, and methods employed in our investigation. Section IV presents the research findings, accompanied by details about the dataset utilized in our experiments. Finally, Section V concludes the paper, summarizing key insights, and outlines potential avenues for future research.

II. Related Work

The cornerstone of this analysis lies in the systematic methodology employed to evaluate the efficacy of different algorithms. To assess Natural Language Analysis, the study embarks on a journey encompassing feed-forward neural networks, Bayesian classifiers, and transformer-based models. These algorithms illuminate the power of NLP in discerning sentiments and differentiating between spam and legitimate content [14]. The intricate interplay between text-based neural networks and advanced transformer models underscores the intricate art of tweet interpretation.

A substantial body of research has investigated spam detection within OSNs, addressing challenges and proposing diverse methods [17]. In [1], the combination of social honeypots and machine learning techniques emerges as an effective strategy for identifying spam. These "honeypots" entail the creation of synthetic profiles to attract spammers' attention.

Exploring the domain of pharmaceutical spam detection on Twitter, [12] employs text mining and decision tree (J48) and Naive-Bayes algorithms. A training set of 65 pharmaceutical-related words facilitates accurate classification of incoming spam. Real-time online spam filtering, discussed in [15], uses machine learning algorithms, notably Support Vector Machine (SVM) and Decision Tree. Messages are scrutinized and discarded preemptively if flagged as spam.

Context-aware spam originating from shared information on social networks is studied in [16], with a focus on defense strategies against context-aware e-mail attacks on platforms like Facebook. To safeguard social networks against content polluters, [19] employs honeypots and machine learning algorithms, collecting substantial spam data and showcasing the potential for protection.

An investigation into detecting spam bots, especially on platforms like Twitter, is detailed in [11]. Behavior patterns of suspicious accounts are closely examined using various classification methods, including Naive-Bayes, determined as the most effective choice through comparative analysis of labeled accounts.

III. Methodology

The goal of this thesis is to detect spam content in Twitter users' posts using ML models. Initially, the data collection phase is implemented and in following, two categories of analysis are considered:

A. Data Collection and Pre-processing

To create the training and test datasets for natural language analysis, data is downloaded with use of Twitter API. The downloaded tweets meet specific criteria, including containing the hashtag #COVID19, being in English, and not being retweets. The collection of COVID-19 related tweets ensures the relevance of the data to current events.

After data collection, the trivial pre-processing steps are applied to the text of the tweets, like the removal of various links (URLs), stopwords, numbers, special characters (#, @, ...), and unnecessary spaces, as well as the conversion of all characters to lowercase to avoid duplicate words during analysis and stemming, lemmatization.

B. 1st Training Dataset: Natural Language Analysis

In this category, the analysis is based solely on the text of the tweets to classify them as spam or non-spam.

The training dataset for natural language analysis consists of 5,572 data entities, with 747 belonging to the spam class and 4,825 to the non-spam class. This results in a class distribution of 13.4% spam and 86.6% non-spam. The length of the tweets varies in both classes.

Below, Figure 1 presents how the number of words and characters varies, in the largest percentage of the data, per class.

C. 2nd Training Dataset: Multiple Attribute Analysis

In this category, tweets are classified based on multiple characteristics extracted from the data, all of which are numeric with integer values.

For multiple attribute analysis, features are extracted from the .json object of each tweet to create the training and test datasets. The data is already in numerical form, so no feature extraction is needed. However, some data values may be undefined, empty, or infinite, which are replaced during pre-processing.

Some models work better with quantized data, where each attribute's value is assigned to an interval based on equal distribution among available categories. The quantized data is used to create the 3rd training dataset.

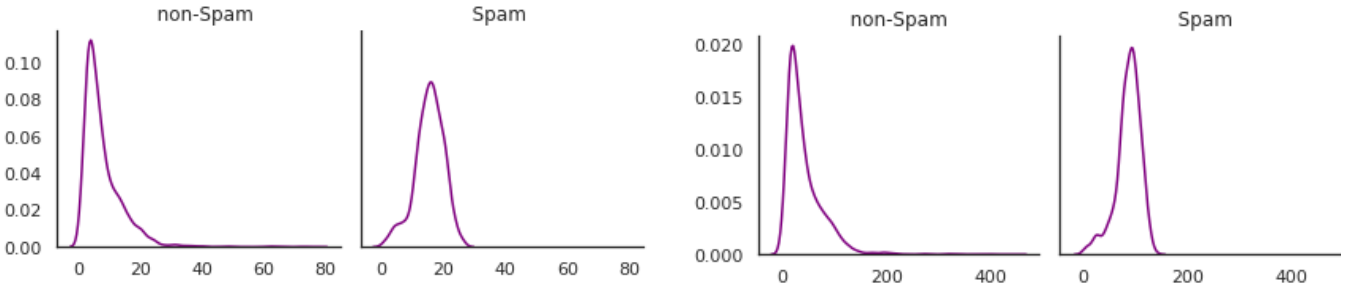
The training dataset for multiple attribute analysis contains 10,000 input entities, evenly distributed between the spam and non-spam classes. The six characteristics considered are:

- The number of hashtags contained in the tweet as presented in Figure 2.
- The number of user references to the tweet as presented in Figure 3.
- The number of numeric digits in the tweet as presented in Figure 4.
- The number of users following the account that posted the tweet as presented in Figure 5.
- The number of users this account follows as presented in Figure 6.
- The number of public lists this account is a member of as presented in Figure 7.

IV. Experimental Evaluation

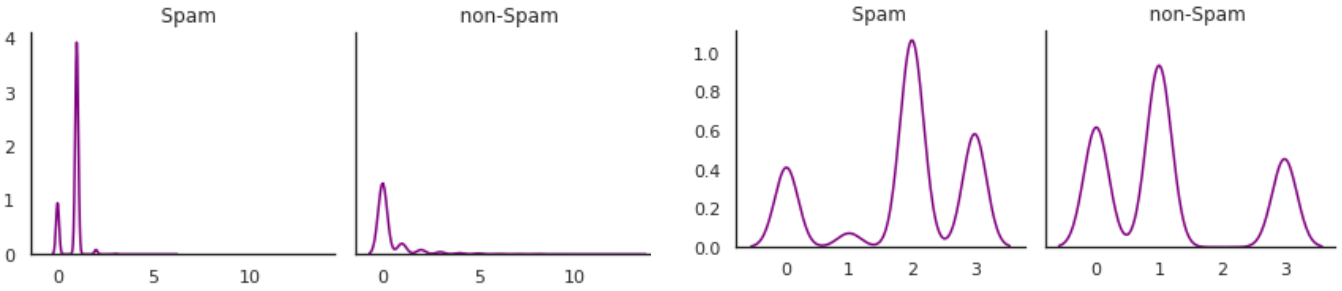
A. Natural Language Analysis

The Natural Language Analysis class of implementations involves applying various algorithms to analyze the text from each tweet. The algorithms used and their corresponding results are described below.



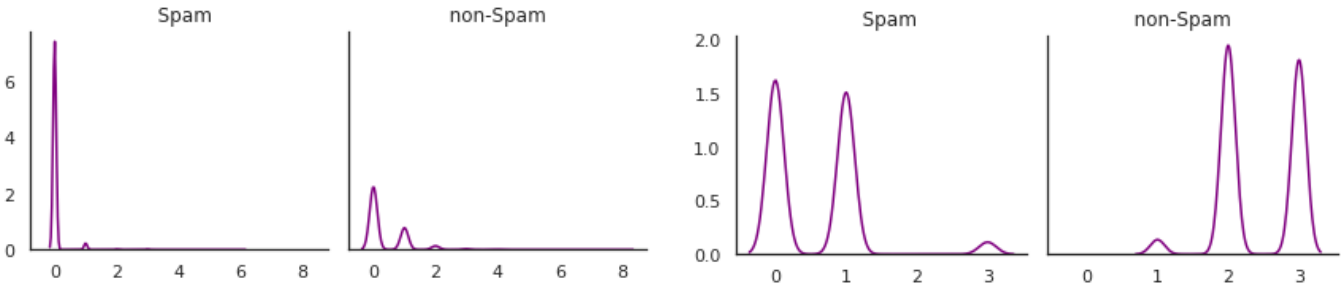
(a) Words (b) Characters

Fig. 1. Number of Words and Characters



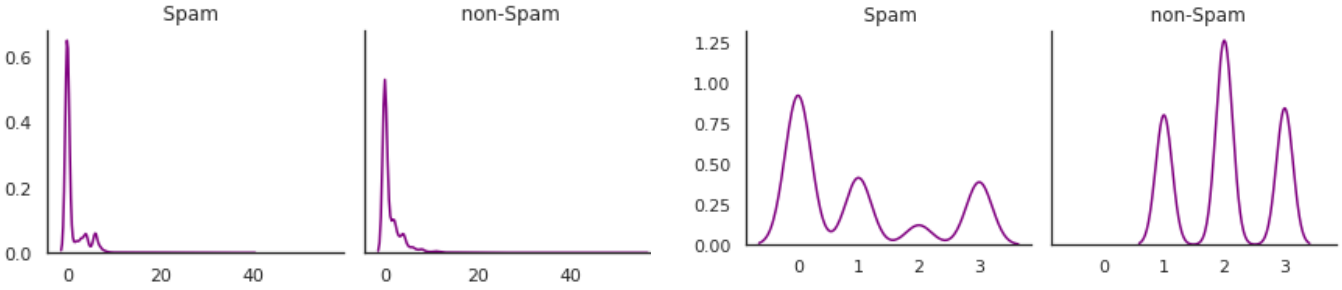
(a) 2nd Training Dataset (b) 3rd Training Dataset

Fig. 2. The Number of Hashtags contained in the Tweet



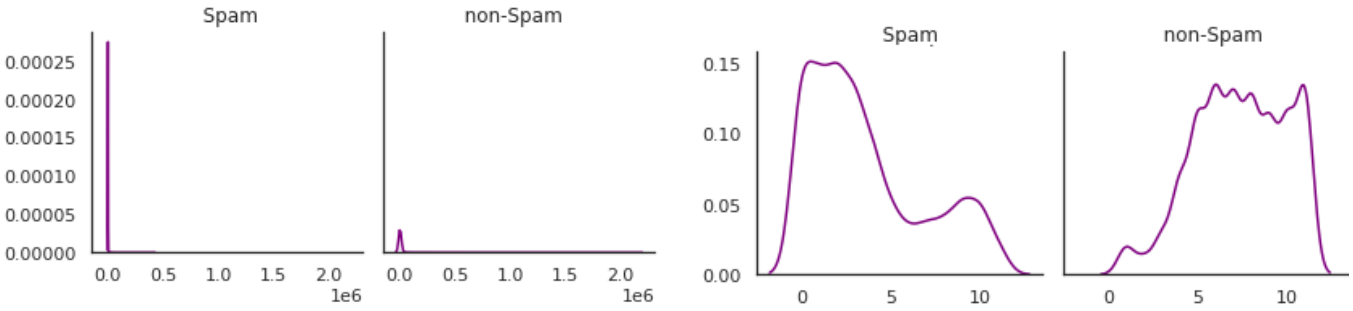
(a) 2nd Training Dataset (b) 3rd Training Dataset

Fig. 3. The Number of User References to the Tweet



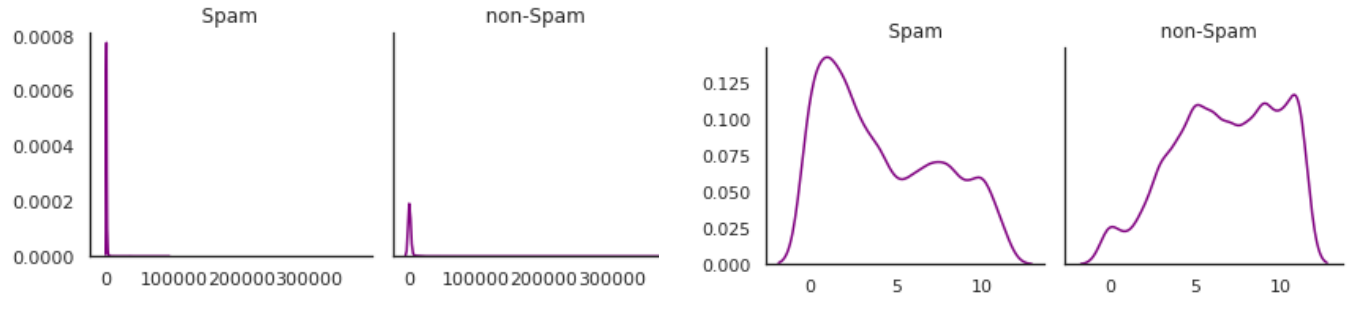
(a) 2nd Training Dataset (b) 3rd Training Dataset

Fig. 4. The Number of Numeric Digits in the Tweet



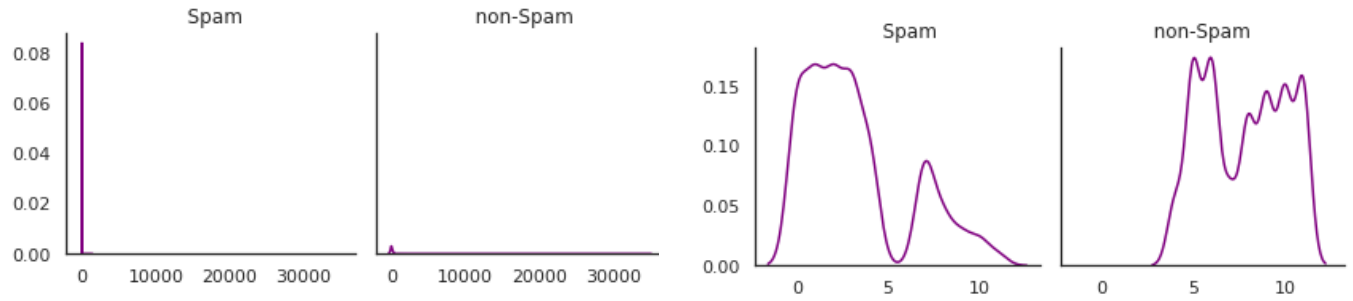
(a) 2nd Training Dataset (b) 3rd Training Dataset

Fig. 5. The Number of Users Following the Account that posted the Tweet



(a) 2nd Training Dataset (b) 3rd Training Dataset

Fig. 6. The Number of Users this Account Follows



(a) 2nd Training Dataset (b) 3rd Training Dataset

Fig. 7. The Number of Public Lists this Account is a Member of

1) Feed-forward Neural Networks: In this implementation, we use a feed-forward neural network model built with the help of the TensorFlow library and the Keras API. The goal is to make predictions based solely on the text from each tweet, utilizing NLP techniques. The data from the first set will be used as the training data.

Before training the neural network, the training data is pre-processed to remove unwanted elements. However, neural networks work with numeric data, not chunks of text. Therefore, the text data is converted into numeric form and a dictionary is created with all the words from the tweets.

The neural network implementation starts with an embedding layer that efficiently manages words using word embeddings, where similar words have similar encodings in an n-dimensional space. The input of this layer is the sentences, and its output is a matrix of size 100×16, where 100 is the number of vectors, and 16 is the dimension of each vector. Each word in the original sentence is represented as a vector of 16 dimensions. The vectors are initially assigned arbitrary values, which are updated during the training process. After training, similar data will have similar encodings.

The neural network consists of the following layers:

- Embedding layer: Transforms input sentences into a matrix of word embeddings.
- Flatten layer: Converts the two-dimensional data to one-dimensional for the fully connected part of the network.
- Output layer: Consists of a single node representing the network's prediction for the input tweet.

The model is trained for 15 epochs, during which the error on the training and validation data is reduced, and the prediction accuracy on both datasets is increased. The model achieves the following prediction accuracy:

- 99.85% on the training data
- 98.57% on the validation data

To avoid overfitting, the number of epochs is then reduced to 7. This model achieves the following prediction accuracy:

- 99.51% on the training data
- 98.30% on the validation data

2) Bayesian Classifier: In this implementation, we use a Bayesian classifier. The training data used is from the first set, and it has undergone the same pre-processing steps.

The algorithm is trained with the training data, which consists of a document-term matrix (DTM) with dimensions $5,572 \times 7,599$. 5,572 corresponds to the size of the training dataset, and 7,599 corresponds to the size of the dictionary containing all the different words.

The Bayesian classifier is chosen, assuming that the data follows a polynomial distribution. The model achieves the following prediction accuracy:

- 96.95% on the training data
- 97.0% on the validation data

To evaluate the model robustly, the training data is divided into 5 subsets, each containing 2,000 data points. The algorithm is trained and tested five times, each time with a different subset as the validation data and the rest as the training data.

From these cases, the following metrics for the validation data are obtained:

TABLE I
BAYESIAN CLASSIFIER FOR VALIDATION DATA (5 CASES)

Instance	Training Precision	Validation Precision	Recall	F1 Score
1st Case	87.0%	87.5%	84.3%	87.1%
2nd Case	87.2%	86.1%	83.1%	85.6%
3rd Case	86.6%	87.9%	85.4%	86.6%
4th Case	86.9%	86.8%	83.3%	85.0%
5th Case	86.4%	86.4%	85.0%	85.6%

Overall, the Bayesian classifier demonstrates good performance in classifying tweets as spam or non-spam.

3) Transformer-based Model: Another approach to Natural Language Analysis is using transformer-based models, such as BERT (Bidirectional Encoder Representations from Transformers). In this implementation, we utilize the Hugging Face's 'transformers' library to employ the pre-trained BERT model.

The BERT model is first fine-tuned on the training data using the AdamW optimizer and the Binary Cross-Entropy Loss (BCELoss). The model is trained for 3 epochs with a learning rate of 2e-5 and achieves the following prediction accuracy:

- 98.63% on the training data
- 98.58% on the validation data

Overall, for feed-forward neural networks and transformers, the metrics for the validation data are derived in Table II.

TABLE II
FEED-FORWARD NEURAL NETWORKS AND TRANSFORMERS

Model	Accuracy	Precision	Recall	F1 Score
Feed-forward Neural Networks	98.3%	97.0%	89.7%	93.2%
Transformers	98.58%	97.2%	90.9%	93.9%

B. Multiple Attribute Analysis

The Multiple Attribute Analysis class of implementations involves utilizing various ML algorithms to analyze the tweets' attributes. To assess the algorithm's performance, we employ a 5-fold cross-validation approach on the training data set. The training data set consists of 10,000 labeled samples, and we divide it into 5 subsets, each containing 2,000 data points.

Specifically, cross-validation is a resampling technique used to evaluate ML models on limited data samples. By dividing the data into multiple subsets and iteratively using them for training and validation, we can obtain a more robust evaluation of the algorithm's performance. In our 5-fold cross-validation, we split the data into 5 sets: 4 sets each containing 2,000 samples are used for training, and the remaining set with 2,000 samples is used for validation in each iteration. We repeat this process 5 times, with each of the 5 subsets serving as the validation set exactly once.

1) Random Forest: Random Forest is an ensemble learning method that constructs multiple decision trees during training and outputs the mode of the classes for classification tasks. The data from the second set is used for training the Random Forest model. Before training, the data is pre-processed to handle missing values and ensure numerical compatibility. The data is split into training and validation sets in an 80:20 ratio.

2) Logistic Regression: Logistic Regression is another ML algorithm used for multiple attribute analysis. The same pre-processed data from the Random Forest implementation is used for training. The data is split into training and validation sets with an 80:20 ratio.

3) Gradient Boosting: Gradient Boosting is another ensemble learning technique that builds multiple weak learners (typically decision trees) sequentially. Each new learner corrects the errors of its predecessor, leading to improved overall performance. The data is split into training and validation sets with an 80:20 ratio.

The prediction accuracies of these classifiers on the validation data are tabulated in Table III, illustrating their performance across various attributes.

TABLE III
RANDOM FOREST, LOGISTIC REGRESSION AND GRADIENT BOOSTING CLASSIFIERS FOR VALIDATION DATA

Attributes	Random Forest	Logistic Regression	Gradient Boosting
1st Attribute	92.47%	89.34%	91.58%
2nd Attribute	83.12%	78.95%	80.23%
3rd Attribute	95.36%	91.02%	93.47%
4th Attribute	87.21%	82.17%	85.09%
5th Attribute	96.84%	93.76%	95.25%

From these results, it's evident that Random Forest performs well for attribute 5 but relatively worse for attribute 2. Similar trends are observed for Logistic Regression and Gradient Boosting, showcasing strengths for attribute 5 while displaying comparatively lower accuracy for attribute 2.

The results of this section underline the differential performance of these ML algorithms when analyzing tweet attributes. Attribute-specific strengths and limitations offer valuable insights, contributing to a comprehensive understanding of attribute analysis applications such as sentiment analysis, spam detection, and user profiling.

C. Discussion

For Natural Language Analysis, the feed-forward neural network achieved high accuracy on both the training and validation data. However, to avoid overfitting, it was essential to limit the number of epochs during training. The transformer-based BERT model also performed well, capturing contextual information and achieving high accuracy, precision, recall, and F1 score on the validation data.

In the Multiple Attribute Analysis, Random Forest, Logistic Regression, and Gradient Boosting were utilized to analyze the attributes of the tweets. Attribute 5 consistently had the highest prediction accuracy across all three algorithms. On the other hand, attribute 2 showed relatively lower accuracy compared to the other attributes.

Overall, the results demonstrate the effectiveness of various algorithms in analyzing tweets from different perspectives. The findings provide valuable insights for understanding tweet content and attributes in real-world applications, including sentiment analysis, spam detection, and user profiling.

V. Conclusions and Future Work

In conclusion, this paper aims to shed light on the ever-evolving landscape of tweet analysis algorithms by conducting a rigorous comparative exploration. The diverse range of algorithms employed underscores their potential in extracting meaningful insights from the complex world of tweets. By bridging the gap between NLP and ML, this study contributes to the broader discourse surrounding social media analysis and provides valuable insights for researchers, practitioners, and decision-makers seeking to harness the power of tweet data.

In light of the insights gained from this comprehensive analysis of tweet analysis algorithms using both NLP and ML models, several promising directions for future research emerge. Firstly, while the current study focused on specific attributes and sentiment analysis, further exploration could encompass a broader range of attributes, such as user behavior, content virality, and context-based analysis [6], [20]. Additionally, investigating the integration of ensemble methods, where multiple algorithms are combined to enhance predictive accuracy, could lead to more robust models. Finally, the study predominantly employed well-established algorithms; exploring emerging techniques and models, like deep learning architectures or attention mechanisms, could potentially yield improved performance [7], [18].

Acknowledgement

This research was funded by the European Union and Greece (Partnership Agreement for the Development Framework 2014-2020) under the Regional Operational Programme Ionian Islands 2014-2020, project title: "Indirect costs for project "TRaditional corfU Music PresErvation through digiTal innovation"", project number: 5030952.

References

[1] F. Benevenuto, G. Magno, T. Rodrigues, and V. Almeida. Detecting spammers on twitter. In *7th Annual Collaboration, Electronic messaging, Anti Abuse and Spam Conference (CEAS)*, volume 6, page 12, 2010.

[2] N. C. Dang, M. N. Moreno-García, and F. D. la Prieta. Sentiment analysis based on deep learning: A comparative study. *Electronics*, 9(3):483, 2020.

[3] G. Drakopoulos, A. Kanavos, P. Mylonas, and S. Sioutas. Discovering sentiment potential in twitter conversations with hilbert-huang spectrum. *Evolving Systems*, 12(1):3–17, 2021.

[4] E. Kafeza, A. Kanavos, C. Makris, G. Pispirigos, and P. Vikatos. T-PCCE: twitter personality based communicative communities extraction system for big data. *IEEE Transactions on Knowledge and Data Engineering*, 32(8):1625–1638, 2020.

[5] E. Kafeza, A. Kanavos, C. Makris, and P. Vikatos. T-PICE: twitter personality based influential communities extraction system. In *IEEE International Congress on Big Data*, pages 212–219, 2014.

[6] A. Kanavos, I. Karamitsos, and A. Mohasseb. Exploring clustering techniques for analyzing user engagement patterns in twitter data. *Computers*, 12(6):124, 2023.

[7] A. Kanavos, F. Kounelis, L. Iliadis, and C. Makris. Deep learning models for forecasting aviation demand time series. *Neural Computing and Applications*, 33(23):16329–16343, 2021.

[8] A. Kanavos, I. Perikos, I. Hatzilygeroudis, and A. K. Tsakalidis. Emotional community detection in social networks. *Computers & Electrical Engineering*, 65:449–460, 2018.

[9] K. Kandasamy and P. Koroth. An integrated approach to spam classification on twitter using url analysis, natural language processing and machine learning techniques. In *IEEE Students' Conference on Electrical, Electronics and Computer Science*, pages 1–5, 2014.

[10] K. Larson and R. T. Watson. The impact of natural language processing-based textual analysis of social media interactions on decision making. In *21st European Conference on Information Systems*, page 70, 2013.

[11] K. Lee, J. Caverlee, and S. Webb. Uncovering social spammers: Social honeypots + machine learning. In *33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 435–442, 2010.

[12] K. Lee, B. D. Eoff, and J. Caverlee. Seven months with the devils: A long-term study of content polluters on twitter. In *5th International Conference on Weblogs and Social Media*. The AAAI Press, 2011.

[13] B. Liu and L. Zhang. A survey of opinion mining and sentiment analysis. In *Mining Text Data*, pages 415–463. Springer, 2012.

[14] A. Mohasseb, B. Aziz, and A. Kanavos. SMS spam identification and risk assessment evaluations. In *16th International Conference on Web Information Systems and Technologies (WEBIST)*, pages 417–424, 2020.

[15] G. Stringhini, C. Kruegel, and G. Vigna. Detecting spammers on social networks. In *26th Annual Computer Security Applications Conference (ACSAC)*, pages 1–9. ACM, 2010.

[16] K. Thomas, C. Grier, J. Ma, V. Paxson, and D. Song. Design and evaluation of a real-time URL spam filtering service. In *32nd IEEE Symposium on Security and Privacy (S&P)*, pages 447–462, 2011.

[17] M. Verma, Divya, and S. Sofat. Techniques to detect spammers in twitter: A survey. *International Journal of Computer Applications*, 85(10), 2014.

[18] S. Vernikou, A. Lyras, and A. Kanavos. Multiclass sentiment analysis on covid-19-related tweets using deep learning models. *Neural Computing and Applications*, 34(22):19615–19627, 2022.

[19] A. H. Wang. Detecting spam bots in online social networking sites: A machine learning approach. In *24th Annual IFIP Conference on Data and Applications Security and Privacy*, pages 335–342, 2010.

[20] V. Zamparas, A. Kanavos, and C. Makris. Real time analytics for measuring user influence on twitter. In *27th International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 591–597, 2015.