Detecting Misinformation Virality on WhatsApp

Fernanda Nascimento, Melissa Sousa, Gustavo Martins, José Maria Monteiro and Javam Machado
Computer Science Department, Federal University of Ceará, Brazil

Keywords: Virality, Misinformation, WhatsApp.

Abstract: In recent years, the large-scale dissemination of misinformation through social media has become a critical issue, undermining public health, social stability, and even democracy. In many developing countries, such as Brazil, India, and Mexico, the WhatsApp messaging app is one of the primary sources of misinformation. Recently, automatic misinformation detection using machine learning methods has emerged as an active research topic. However, not all misinformation is equally significant. For instance, widely spread misinformation can have a greater impact and tends to be more persuasive to users. In this context, we address the early detection of misinformation virality by analyzing the textual content of WhatsApp messages. First, we introduced a large-scale, labeled, and publicly available dataset of WhatsApp messages, called FakeViralWhatsApp.Br, which contains 52,080 labeled messages along with their respective frequencies. Next, we conducted a series of binary classification experiments, combining three feature extraction methods, three distinct token generation strategies, two preprocessing approaches, and six classification algorithms to classify messages into viral and non-viral categories. Our best results achieved an F1-score of 0.98 for general posts and 1.0 for misinformation messages, demonstrating the feasibility of the proposed approach.

1 INTRODUCTION

In recent years, the large-scale dissemination of misinformation through social media has become a critical issue. In many developing countries, such as Brazil, India, and Mexico, one of the primary sources of misinformation is WhatsApp, a popular instant messaging application. The WhatsApp messaging app is extremely popular in Brazil, with over 165 million users out of a total population of approximately 214 million (de Sá et al., 2021). WhatsApp's popularity is primarily due to its versatility and ease of use.

In addition, WhatsApp offers two essential features facilitating content spread: public groups and channels. These functionalities are accessible through invitation links and generally focus on specific discussion topics such as politics, sports, finance, or education. WhatsApp enables users to join or share public groups and channels, allowing them to connect with hundreds of people simultaneously and quickly receive and distribute digital content. Due to their popularity, many WhatsApp groups and channels have been used to spread misinformation, particularly as part of coordinated political or ideological campaigns.

Automatic misinformation detection based on machine learning methods has been an active research topic in recent years (Cabral et al., 2021; Martins et al., 2021a; Martins et al., 2021b). However, the effectiveness of online surveillance is limited by the cost of issuing alerts for all WhatsApp messages that contain misinformation indiscriminately, especially since not all misinformation is equally significant. For example, widely spread misinformation can have a more substantial impact, as high virality reinforces perceived social norms (Kim, 2018). By accurately estimating the virality of the message in the context of misinformation detection, we can more efficiently track misinformation on the WhatsApp platform.

In this paper, we address the prediction of misinformation virality using the textual content of WhatsApp messages. Furthermore, we introduce FakeViralWhatsApp.Br, a large-scale, labeled, and publicly available dataset of WhatsApp messages, collected from public chat groups. Then, we conducted a series of experiments to evaluate six algorithms in classifying messages into viral and non-viral categories. Our best results achieved an F1-score of 0.98 for general posts and 1.0 for misinformation messages.

The remainder of this paper is organized as follows. Section 2 presents our "WhatsApp Virality"

Nascimento, F., Sousa, M., Martins, G., Monteiro, J. M., Machado and J.
Detecting Misinformation Virality on WhatsApp.
DOI: 10.5220/0013646800003967
In *Proceedings of the 14th International Conference on Data Science, Technology and Applications (DATA 2025)*, pages 685-692
ISBN: 978-989-758-758-0; ISSN: 2184-285X

definition. Section 3 discusses the main related work. Section 4 describes the methodology used in this investigation. Section 5 presents the experimental results. Conclusions and future work are presented in Section 6.

2 WHATSAPP VIRALITY DEFINITION

Messages in WhatsApp can be divided into two groups as follows:

- Viral Messages: which are sent by audiences of channels to other users or different groups.
- Ordinary or Non-Viral Messages: which are only read by the users in channels, not sent to others.

In this work, we propose an embracing definition for the concept of "Virality on WhatsApp", which is formulated next:

Definition 2.1 (Viral Message)**.** A published message k, in a set of n WhatsApp messages, is called viral if the associated post counter is greater than t% of the frequency of the other posts.

3 RELATED WORKS

In recent years, there has been increasing interest in detecting viral messages across various social networks, including Twitter and Telegram. However, few studies have explicitly focused on the WhatsApp platform.

The problem of predicting viral messages on Telegram was investigated in (Dargahi Nobari et al., 2021). In this paper, the authors proposed statistical and word embedding-based approaches to detect viral messages on Telegram. Their experiments demonstrated that the word embedding approach significantly outperformed other baseline models, achieving an F1-score of 0.88.

The Twitter network has been investigated in many works. In (Esteban-Bravo et al., 2024), a forecasting methodology was presented to predict the virality level of fake news using only its content. Specifically, machine learning models were used to classify fake news into four levels of virality. Using a large dataset of messages posted on Twitter, the Random Forest algorithm achieved the best performance, reporting an F1-score of 0.99.

In (Elmas et al., 2023), the authors leveraged Twitter's "Viral Tweets" dataset to assess existing metrics for labeling tweets as viral and proposed a new one. They also introduced a transformer-based model for early detection of viral tweets, achieving an F1-score of 0.79. In (Arunkumar and Jadhav, 2024), machine learning algorithms were used to predict whether a tweet would go viral. Additionally, the authors identified and validated the key factors contributing to a tweet's success.

In (Zhang and Gao, 2024), the authors proposed a novel approach to predict viral rumors and vulnerable users using a unified graph neural network model. They pre-train network-based user embeddings and leverage a cross-attention mechanism between users and posts, besides a community-enhanced vulnerability propagation (CVP) method to improve user and propagation graph representations. Furthermore, they employ two multi-task training strategies to mitigate negative transfer effects among tasks in different settings, enhancing the overall performance of the proposed approach. The results showed that the proposed approach outperforms baselines in all three tasks: rumor detection, virality prediction, and user vulnerability scoring. The proposed method reduces mean squared error (MSE) by 23.9% for virality prediction, reporting an MSE of 0.197 and 0.603 for Twitter and WEIBO datasets, respectively.

In (Rameez et al., 2022), the authors introduced ViralBERT, a model designed to predict tweet virality using both content and user features. Their results showed that ViralBERT outperformed baseline models, reporting an F-Measure of 0.523 and an Accuracy of 0.494.

In (Liu et al., 2024), the authors examined two structural features of WhatsApp to analyze information spread: breadth and depth. Breadth measures the maximum number of groups to which a message is simultaneously sent, while depth measures how often a message is forwarded. By analyzing different dissemination patterns on WhatsApp, the study provided insights into message propagation dynamics on private messaging platforms.

In (Maarouf et al., 2024), the authors collected 25,219 cascades with 65,946 retweets from X (formerly known as Twitter) and classified them as hateful vs. normal. Then, using a linear regression, they estimated differences in the spread of hateful vs. normal content based on author and content variables. The results pointed to important aspects that explain differences in the spreading of hateful vs. normal content.

4 THE PROPOSED DATASET

To develop an automatic detector of misinformation virality in the context of WhatsApp, it is essential to use a large-scale dataset composed of messages in Brazilian Portuguese (PT-BR) that have circulated on this platform. However, no existing corpus with these characteristics has been identified. To bridge this gap, we built a dataset, named FakeViralWhatsApp.Br, consisting of messages collected from public WhatsApp groups and channels. For this purpose, we followed the guidelines proposed by (Rubin et al., 2015) for constructing a corpus designed for classification tasks.

4.1 Data Collection

The WhatsApp messages collection was conducted using the platform described in (de Sá et al., 2023) between September 1, 2023, and November 19, 2023. On WhatsApp, a total of 269,473 messages were collected from 179 public groups. Manually labeling such a large volume of messages is unfeasible. Therefore, a strategy to reduce the number of messages to be annotated is necessary.

4.2 Data Anonymization

To ensure user privacy, personal data such as names and phone numbers were anonymized. Additionally, we applied a hash function to generate a unique and anonymous identifier for each user based on their phone number. Furthermore, a hash function was also used to create a unique and anonymous identifier for each group, derived from its name. Since these groups are publicly accessible, our approach does not violate WhatsApp's privacy policy[1] nor the General Data Protection Law (LGPD).

4.3 Data Labeling

Data labeling, the process of annotating data to train supervised machine learning models, is a complex task (Haber et al., 2023). Its main challenges include label quality and consistency, cost aspects, etc. In this context, we needed to determine whether a given WhatsApp message was viral or not. For this, we used Definition 2.1, where a published message k, in a set of n messages, is called viral if the associated post counter is greater than t% of the frequency of the other posts. In this work, we used $t = 90\%$, that is, a message k will be labeled as "viral" if its post counter is greater than 90% of the frequency of the other posts. Thus, using $t = 90\%$, of the total of 164,662 messages, 16,466 posts were labeled as "viral" and 148,196 were labeled as "non-viral". It is important to note that, after this step, we obtained a highly unbalanced dataset. Then, we applied an undersampling strategy to address this issue by randomly selecting a subset of non-viral messages, resulting in 16,466 posts. After this procedure, the resulting dataset comprised 16,466 viral and 16,466 non-viral messages, totaling 52,080 posts. We will refer to this subset as the "Viral Dataset". The definition of virality based on message frequency is supported by well-established research in the field of misinformation on messaging platforms (Resende et al., 2019). In future work, we intend to study this threshold sensitivity.

Next, we assessed whether a given message included any form of misinformation. However, manually labeling 52,080 messages is unfeasible. Therefore, a strategy to reduce the number of messages to be annotated is necessary. We applied the following strategy to reduce the number of messages to be labeled. From the subset of 16,466 viral messages, a message was randomly selected and subsequently labeled. We repeated this procedure until we obtained 1,000 viral messages containing misinformation. Then, we repeated this strategy for the subset of 16,466 non-viral messages, obtaining 1,000 non-viral messages containing misinformation. This process results in a balanced subset, called "Misinformation Dataset", containing 2,000 misinformation posts.

It is important to emphasize that the labeling process was entirely manual to ensure a high-quality textual corpus. Three annotators conducted the labeling process, and disagreements were resolved through a collective review to ensure consistency and reliability. The FakeViralWhatsApp.Br dataset can be accessed freely from a public repository [2].

5 EXPERIMENTAL EVALUATION

5.1 Baseline Evaluation

To provide a baseline for the problem of predicting viral messages on the WhatsApp platform, a series of experiments was conducted using the FakeViralWhatsApp.Br dataset.

[1] https://www.whatsapp.com/legal/privacy-policy

[2] https://github.com/jmmfilho/misinformation_virality

5.1.1 Features and Classification Algorithms

As previously mentioned, three distinct feature extraction methods were evaluated: Bag of Words (BoW), Term Frequency–Inverse Document Frequency (TF-IDF), and Word2vec. These methods were chosen due to their simplicity, speed, and widespread use in text classification tasks. Pre-trained embedding vectors were not used due to the high occurrence of misspelled words, emoticons, and neologisms in the corpus. The text was converted to lowercase before applying the BoW, TF-IDF, and Word2Vec methods. It is important to note that emojis are highly prevalent in the dataset and play a significant role in the language used in instant messaging applications. For this reason, they were retained in the preprocessing step. However, since emoji combinations can generate different types of tokens, a whitespace separation strategy was applied, ensuring that each emoji is treated as an individual token. Additionally, URL normalization was performed, where only the domain name was preserved. Due to the lexical diversity of the corpus, the resulting feature vectors are sparse and exhibit high dimensionality.

Three different tokenization strategies were evaluated: unigrams, bigrams, and trigrams. While this approach results in high-dimensional vectors, it is expected to reveal distinct patterns in the messages, as bigrams and trigrams can capture more context-related information. Additionally, two approaches were considered to assess the impact of preprocessing techniques: i) no preprocessing, and ii) using stopwords removal and lemmatization. These techniques aim to reduce noise, enabling a more precise representation of the relevant features present in the messages.

Thus, 14 different execution scenarios were created by combining three feature extraction methods (BoW, TF-IDF, and Word2Vec), three tokenization strategies (unigrams, bigrams, and trigrams), and two preprocessing approaches (with and without preprocessing). For each of these scenarios, we evaluated six classical classification algorithms: Logistic Regression - LR, Regularized Logistic Regression - RLG, Decision Tree - DT, Regularized Decision Tree - RDT, Gradient Boosting - GB, and Regularized Gradient Boosting - RGB. The classification algorithms were implemented using the scikit-learn Python library (Pedregosa et al., 2011). All data and code used in the experiments are available in our online repository[3].

5.1.2 Performance Metrics

As previously mentioned, the investigated problem is a binary classification task, where "viral" represents the positive class (also our class of interest), and "non-viral" represents the negative class. To evaluate the performance of each model, the following metrics were used: False Positive Rate (FPR), Precision (PRE), Recall (REC), and F1-score (F1).

After performing k-fold cross-validation ($k = 5$), we selected the best classifier and the most effective features. Next, we retrained the model using a randomly selected training set, which corresponds to 80% of the total available data. Subsequently, we evaluated the model's performance using the remaining 20% of the data, which was initially set aside to form the test set.

5.1.3 Experimental Setups

In this paper, we conducted three distinct experiments. The first one aimed to assess the feasibility of predicting whether general WhatsApp messages (i.e., messages that may or may not contain misinformation) would go viral or not. For this purpose, we used the subset referred to as "Viral Dataset", which comprises 26,040 viral messages and 26,040 non-viral messages, totaling 52,080 posts. The second experiment is similar to the previous one. However, it was designed to assess whether incorporating a feature that estimates the probability of a message containing misinformation enhances the prediction of general WhatsApp messages' virality. The third experiment evaluated the feasibility of predicting the virality of WhatsApp misinformation messages, that is, messages labeled as misinformation. In this experiment, we used the subset referred to as the "Misinformation Dataset" which consists of 2,000 messages involving some form of misinformation, with 1,000 classified as viral and 1,000 as non-viral.

6 RESULTS

This section presents and discusses the results obtained from three distinct experiments: predicting the virality of general WhatsApp messages; predicting the virality of general WhatsApp messages using the misinformation probability feature; and predicting the virality of WhatsApp misinformation messages. The best results for each experiment are shown in Table 1.

[3]https://github.com/jmmfilho/misinformation_virality

Table 1: Best Result for Each Experiment.

Experiment	Precision	Recall	F1-score
General Messages (Word2Vec + Regularized Gradient Boosting)	0.974	1.000	0.987
General Messages Using the Misinformation Probability Feature (TF-IDF Bigram + Logistic Regression)	0.987	0.999	0.993
Misinformation Messages (BoW Trigram + Logistic Regression)	**1.000**	**1.000**	**1.000**

6.1 Predicting Virality of General WhatsApp Messages

This experiment aimed to assess the feasibility of predicting whether general WhatsApp messages (i.e., messages that may or may not contain misinformation) would go viral or not, a binary classification problem. Six classical classification algorithms were evaluated in 14 different scenarios.

Logistic Regression algorithm achieved the best result using Bow and Trigram in both scenarios, with and without preprocessing, achieving F1-score values of 0.89 and 0.96, respectively. The Regularized Logistic Regression algorithm achieved the best result using Bow and Trigram in both scenarios, with and without preprocessing, achieving F1-score values of 0.89 and 0.95, respectively. Thus, the L1 regularization method did not provide performance gains for the Logistic Regression algorithm.

Decision Tree algorithm achieved the best result using Word2Vec in both scenarios, with and without preprocessing, achieving F1-score values of 0.87 and 0.83, respectively. The Regularized Decision Tree algorithm achieved the best result using Word2Vec in both scenarios, with and without preprocessing, achieving F1-score values of 0.74 and 0.82, respectively. So, the regularization method did not provide performance gains for the Decision Tree algorithm.

Gradient Boosting algorithm achieved the best result using Word2Vec in both scenarios, with and without preprocessing, achieving F1-score values of 0.94 and 0.98, respectively. The Regularized Gradient Boosting achieved the best result using Word2Vec in both scenarios, with and without preprocessing, achieving F1-score values of 0.94 and 0.98, respectively. Then, the use of the L1 regularization method provided a slight improvement in the performance in the scenario using text preprocessing. Besides, it can be concluded that Regularized Gradient Boosting emerged as the most effective method for detecting viral messages among those evaluated. However, in general, regularization did not lead to performance improvements.

Figure 1 presents the learning curve for Regularized Gradient Boosting in the context of the first experiment. Figure 2, in turn, displays the corresponding confusion matrix for the same model, evaluated using the General WhatsApp Messages dataset.

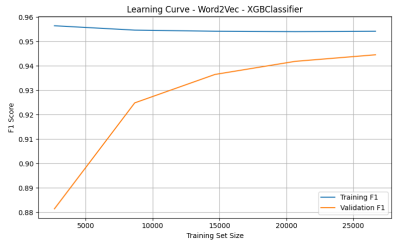

Figure 1: Learning Curve for Regularized Gradient Boosting on General WhatsApp Messages.

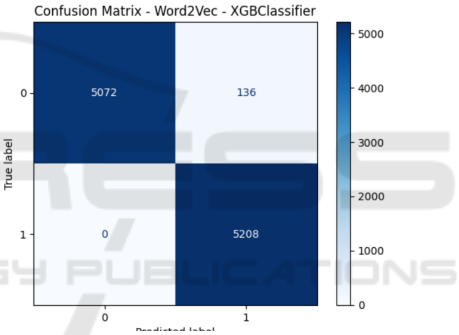

Figure 2: Confusion Matrix of Regularized Gradient Boosting on General WhatsApp Messages.

6.2 Predicting Virality of General WhatsApp Messages Using Misinformation Probability

This experiment was designed to examine whether incorporating a feature that estimates the likelihood of a message containing misinformation improves the prediction of the virality of general WhatsApp messages. To this end, we utilized the subset referred to as the "Viral Dataset", augmenting it with a novel feature representing the estimated probability that a message contains misinformation. This feature was derived using the misinformation classifier proposed in (Cabral et al., 2021).

The Logistic Regression algorithm, without text preprocessing and employing TF-IDF representations with bigrams, achieved an F1-score of 0.993. This performance exceeds the F1-score of 0.96 obtained

when the misinformation probability feature was not included. Similarly, the Regularized Logistic Regression model, under the same conditions, no text preprocessing, and using TF-IDF with bigrams, also attained an F1-score of 0.989, surpassing the F1-score of 0.95 observed when the misinformation probability feature was not included.

The Decision Tree algorithm, when combined with text preprocessing and the use of Word2Vec embeddings, achieved an F1-score of 0.957. This performance surpasses the F1-score of 0.864 obtained by the same algorithm when the misinformation probability feature was not included in the model. The Regularized Decision Tree algorithm, without text preprocessing and using Word2Vec embeddings, achieved an F1-score of 0.836. This result outperforms the F1-score of 0.758 obtained by the same model when the misinformation probability feature was not used.

The Gradient Boosting algorithm, without text preprocessing and employing TF-IDF representations with unigrams, achieved an F1-score of 0.993. This performance exceeds the F1-score of 0.985 obtained when the misinformation probability feature was not incorporated into the model. The Regularized Gradient Boosting algorithm, without text preprocessing and employing TF-IDF representations with bigrams, achieved an F1-score of 0.991. This score surpasses the F1-score of 0.985 obtained by the same model when the misinformation probability feature was not included.

It is important to emphasize that, across all six classification algorithms evaluated, the inclusion of a feature representing the probability that a message contains misinformation consistently led to improved performance. These results suggest that the misinformation probability feature plays a meaningful role in explaining the virality of WhatsApp messages. Furthermore, the use of regularization did not lead to performance improvements.

Figure 3 presents the learning curve for Logistic Regression in the context of the second experiment. Figure 4, in turn, displays the corresponding confusion matrix for the same model. Figure 3 shows that the model generalizes well, achieving high F1-score values with relatively few training examples. By Figure 4, it is noteworthy that the number of false positives and false negatives is extremely low and well balanced between the two classes.

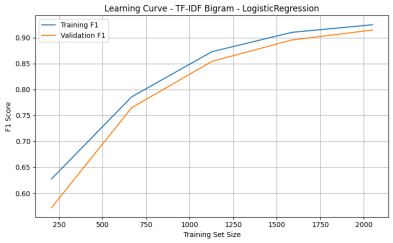

Figure 3: Learning Curve for Logistic Regression Using Misinformation Probability on General Messages.

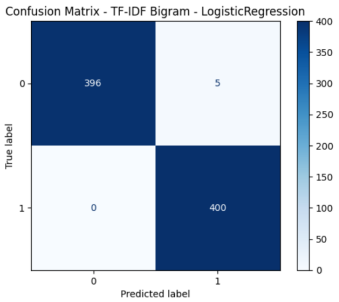

Figure 4: Confusion Matrix for Logistic Regression Using Misinformation Probability on General Messages.

6.3 Predicting Virality of WhatsApp Misinformation Messages

This experiment aimed to assess the feasibility of predicting whether WhatsApp messages containing misinformation would go viral or not, a binary classification problem. Six classical classification algorithms were evaluated in 14 different scenarios.

Logistic Regression achieved the best result using BoW and Bigram without preprocessing and BoW and Bigram with preprocessing, achieving F1-score values of 0.9995 and 1.0, respectively. The Regularized Logistic Regression algorithm performed best using BoW and Bigram without preprocessing and BoW and Bigram with preprocessing, yielding F1-scores of 0.9995 and 1.0, respectively. As evidenced by these results, the application of L1 regularization did not lead to performance improvements for the Logistic Regression model.

The Decision Tree achieved its highest performance using Word2Vec embeddings in both scenarios, with and without text preprocessing, attaining F1-scores of 0.969 and 0.974, respectively. The Regularized Decision Tree algorithm performed best using Word2Vec in both scenarios, with and without text preprocessing, yielding F1-scores of 0.939 and 0.886, respectively. These results indicate that the application of regularization did not contribute to performance gains for the Decision Tree model.

The Gradient Boosting algorithm achieved its best performance using BoW and Unigram/Bigram/Trigram without text preprocessing, reaching an F1-score of 0.9995. Similarly, the Regularized Gradient Boosting model also attained an F1-score of 0.9995 under the same conditions. These results suggest that the application of regularization did not yield performance improvements for the Gradient Boosting algorithm.

It is important to highlight that, across all six classifiers evaluated, the experiments conducted using the dataset composed exclusively of messages containing some form of misinformation (called "Misinformation Dataset") yielded higher performance metric values compared to those conducted with the "Viral Dataset", which includes both misinformation and non-misinformation messages. These findings suggest that virality prediction performs better when applied to texts previously identified as containing misinformation.

Figure 5 presents the learning curve for Logistic Regression in the context of the third experiment. Figure 6, in turn, displays the corresponding confusion matrix for the same model, evaluated using the Misinformation WhatsApp Messages. Figure 6 reveals that the model produced neither false positives nor false negatives, indicating perfect classification performance for the evaluated dataset.

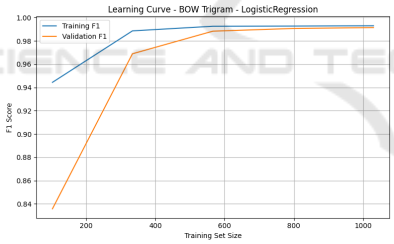

Figure 5: Learning Curve of Logistic Regression on WhatsApp Misinformation Messages.

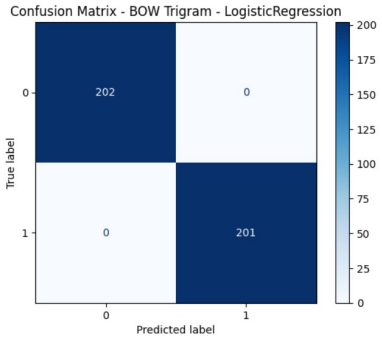

Figure 6: Confusion Matrix of Logistic Regression on WhatsApp Misinformation Messages.

6.4 Threats to Validity

During the execution of the experiments, several threats to validity were identified. These can be classified into internal validity, external validity, construct validity, and conclusion validity (Wohlin et al., 2012). Next, we discuss each of them in detail.

Internal Validity refers to factors that may influence the observations of the study. The message collection occurred during a period of intense political debate, which may have increased the number of misinformation messages, potentially affecting the distribution of the dataset.

External Validity poses a threat to the generalizability of the study's findings. The messages were collected from 179 public WhatsApp groups, primarily focused on political debates. This sample may not fully capture the general behavior of public groups in Brazil, potentially limiting the applicability of the results to other contexts.

Construct Validity concerns the relationship between theory and observation. One primary challenge is defining what constitutes a misinformation text within the specific context of messages shared on WhatsApp and Telegram. Another issue lies in the manual labeling of messages, as the complexity of determining whether a text contains misinformation may lead to misclassification.

Conclusion Validity relates to the reliability of the study's findings. The conclusions drawn in this work can only be considered valid if the messages were correctly labeled. Any labeling errors could directly impact the accuracy of the results.

7 CONCLUSIONS AND FUTURE WORK

In this paper, we introduced a large-scale, labeled, and publicly available dataset of WhatsApp messages, called FakeViralWhatsApp.Br, which contains 52,080 labeled messages along with their respective frequencies. Then, using the proposed dataset, we employed machine learning models to classify messages into two propagation levels: viral and non-viral. We conducted a series of binary classification experiments, combining three feature extraction methods, three distinct token generation strategies, two preprocessing approaches, and six classification algorithms. Our best results achieved an F1-score of 0.98 for general posts and 1.0 for misinformation messages, demonstrating the feasibility of predicting the virality of WhatsApp messages by analyzing its textual content.

ACKNOWLEDGMENTS

This work was partially funded by Lenovo as part of its R&D investment under the Information Technology Law. The authors would like to thank LSBD/UFC for the partial funding of this research.

REFERENCES

Arunkumar, P. and Jadhav, A. (2024). Predicting virality of tweets using ml algorithms and analyzing key determinants of viral tweets. In Sharma, H., Chakravorty, A., Hussain, S., and Kumari, R., editors, *Artificial Intelligence: Theory and Applications*, pages 155–165, Singapore. Springer Nature Singapore.

Cabral, L., Monteiro, J. M., da Silva, J. W. F., Mattos, C. L. C., and Mourao, P. J. C. (2021). Fakewhastapp. br: Nlp and machine learning techniques for misinformation detection in brazilian portuguese whatsapp messages. In *ICEIS (1)*, pages 63–74.

Dargahi Nobari, A., Sarraf, M. H. K. M., Neshati, M., and Erfanian Daneshvar, F. (2021). Characteristics of viral messages on telegram; the world's largest hybrid public and private messenger. *Expert Systems with Applications*, 168:114303.

de Sá, I. C., Galic, L., Franco, W., Gadelha, T., Monteiro, J. M., and Machado, J. C. (2023). BATMAN: A big data platform for misinformation monitoring. In Filipe, J., Smialek, M., Brodsky, A., and Hammoudi, S., editors, *Proceedings of the 25th International Conference on Enterprise Information Systems, ICEIS 2023, Volume 1, Prague, Czech Republic, April 24-26, 2023*, pages 237–246. SCITEPRESS.

de Sá, I. C., Monteiro, J. M., da Silva, J. W. F., Medeiros, L. M., Mourão, P. J. C., and da Cunha, L. C. C. (2021). Digital lighthouse: A platform for monitoring public groups in whatsapp. In Filipe, J., Smialek, M., Brodsky, A., and Hammoudi, S., editors, *Proceedings of the 23rd International Conference on Enterprise Information Systems, ICEIS 2021, Online Streaming, April 26-28, 2021, Volume 1*, pages 297–304. SCITEPRESS.

Elmas, T., Stephane, S., and Houssiaux, C. (2023). Measuring and detecting virality on social media: The case of twitter's viral tweets topic. In *Companion Proceedings of the ACM Web Conference 2023*, WWW '23 Companion, page 314–317, New York, NY, USA. Association for Computing Machinery.

Esteban-Bravo, M., d. l. M. Jiménez-Rubido, L., and Vidal-Sanz, J. M. (2024). Predicting the virality of fake news at the early stage of dissemination. *Expert Systems with Applications*, 248:123390.

Haber, J., Kawintiranon, K., Singh, L., Chen, A., Pizzo, A., Pogrebivsky, A., and Yang, J. (2023). Identifying high-quality training data for misinformation detection. In Gusikhin, O., Hammoudi, S., and Cuzzocrea, A., editors, *Proceedings of the 12th International Conference on Data Science, Technology and Applications, DATA 2023, Rome, Italy, July 11-13, 2023*, pages 64–76. SCITEPRESS.

Kim, J. W. (2018). Rumor has it: The effects of virality metrics on rumor believability and transmission on twitter. *New Media & Society*, 20(12):4807–4825.

Liu, Y., Garimella, K., and Rahimian, M. A. (2024). Virality of information diffusion on whatsapp.

Maarouf, A., Pröllochs, N., and Feuerriegel, S. (2024). The virality of hate speech on social media. *Proc. ACM Hum.-Comput. Interact.*, 8(CSCW1).

Martins, A. D. F., Cabral, L., Mourão, P. J. C., Monteiro, J. M., and Machado, J. C. (2021a). Detection of misinformation about COVID-19 in brazilian portuguese whatsapp messages. In Métais, E., Meziane, F., Horacek, H., and Kapetanios, E., editors, *Natural Language Processing and Information Systems - 26th International Conference on Applications of Natural Language to Information Systems, NLDB 2021, Saarbrücken, Germany, June 23-25, 2021, Proceedings*, volume 12801 of *Lecture Notes in Computer Science*, pages 199–206. Springer.

Martins, A. D. F., Cabral, L., Mourão, P. J. C., Monteiro, J. M., and Machado, J. C. (2021b). Detection of misinformation about COVID-19 in brazilian portuguese whatsapp messages using deep learning. In *Proceedings of the 36th Brazilian Symposium on Databases, SBBD 2021, Rio de Janeiro, Brazil (Online), October 4-8, 2021*, pages 85–96. SBC.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Rameez, R., Rahmani, H. A., and Yilmaz, E. (2022). Viralbert: A user focused bert-based approach to virality prediction. In *Adjunct Proceedings of the 30th ACM Conference on User Modeling, Adaptation and Personalization*, UMAP '22 Adjunct, page 85–89, New York, NY, USA. Association for Computing Machinery.

Resende, G., Melo, P. H. C., Sousa, J. M. A., and Benevenuto, F. (2019). Analyzing textual (mis)information shared in whatsapp groups. In *Proceedings of the 10th ACM Conference on Web Science*, pages 225–234. ACM.

Rubin, V. L., Chen, Y., and Conroy, N. K. (2015). Deception detection for news: three types of fakes. *Proceedings of the Association for Information Science and Technology*, 52(1):1–4.

Wohlin, C., Runeson, P., Höst, M., Ohlsson, M. C., Regnell, B., and Wesslén, A. (2012). *Experimentation in software engineering*. Springer Science & Business Media.

Zhang, X. and Gao, W. (2024). Predicting viral rumors and vulnerable users with graph-based neural multi-task learning for infodemic surveillance. *Information Processing & Management*, 61(1):103520.