Journal of Engineering and Technology for Industrial Applications

ISSN ONLINE: 2447-0228

ITEGAM-JETIA

Manaus, v.11 n.53, p. 43-49. May/June., 2025.

DOI: https://doi.org/10.5935/jetia. v11i53.1443

RESEARCH ARTICLE OPEN ACCESS

MACHINE LEARNING TECHNIQUES FOR IDENTIFYING TEXTUAL PROPAGANDA ON SOCIAL MEDIA: DEVELOPMENT OF A DETECTING DIGITAL MANIPULATION SYSTEM

Belkacem mostefai[1], Tarek Boutefara[2] , Abid Chahinez[3] and Aberkane Marwa[4]

[1] Faculty of Exact Sciences and Computer Science, Djelfa University, 17000 DZ, Djelfa, Algeria.
[2,3,4] Faculty of Exact Sciences and Computer Sciences, University of Jijel,18000 DZ Jijel, Algeria.

[1]http://orcid.org/ 0000-0002-2118-8407, [2]http://orcid.org/0000-0002-7222-9387, [3] http://orcid.org/0009-0001-5469-5878, [4]http://orcid.org/0009-0001-7771-3098

Email: b.mostefai@univ-djelfa.dz, t_boutefara@univ-jijel.dz, abidshahinez@gmail.com, aberkanmerwa@gmail.com

ARTICLE INFO

Article History
Received: November 30, 2024
Revised: January 20, 2025
Accepted: May 15, 2025
Published: May 31, 2025

Keywords:
Propaganda Detection,
Social media,
Machine Learning Techniques,
Linguistic Analysis,
Digital manipulation,

ABSTRACT

The dissemination of propaganda on social media presents a significant challenge in today's digital age. Utilizing advanced tools and diverse methods, propaganda aims to influence public opinion on a massive scale. Social media platforms serve as prime channels for such messages, leveraging sophisticated strategies to shape public perceptions and attitudes. This research aims to develop an advanced system capable of evaluating whether the content disseminated on these platforms qualifies as propaganda. The hypothesis suggests that it is possible to distinguish propaganda from non-propaganda texts on social media by analyzing specific linguistic features. Employing advanced linguistic analysis and machine learning methods, this detection system achieves approximately 70% accuracy, indicating its promising potential for effectively identifying propaganda. This approach could significantly enhance the transparency and reliability of online information, encouraging a more informed and critical use of social media.

I. INTRODUCTION

Propaganda is a powerful tool for shaping collective perceptions and influencing individual behavior [1]. While it has historically been used in political, cultural, and commercial contexts, its integration with social media has amplified its impact, making it a potent mechanism for widespread influence [2], [3]. With vast user networks, social media platforms enable the rapid and viral spread of propaganda, posing significant threats to democracy, public health, and social cohesion. Initially rooted in political and religious campaigns, propaganda has evolved into sophisticated digital strategies [4]-[6]. The spread of propaganda on social media alters the global perception of events, particularly in conflict situations [7], [8]. The use of these platforms to disseminate unverified and sensationalist content misleads and reinforces unfounded prejudices. This distortion of reality severely hampers international conflicts, exacerbates societal divisions, and undermines trust in institutions, highlighting the urgent need to promote fact-checking and critical thinking in the contemporary media landscape. [9], [10].

Propaganda often employ persuasive techniques such as emotional appeals, misinformation, and targeted messaging to manipulate public opinion [11]. Social media enhances both top-down and peer-to-peer dissemination, utilizing tactics like astroturfing [12], bot-generated content [13], and algorithmic manipulation to amplify certain narratives while suppressing others [14]. Machine learning, also known as machine learning, is a field of artificial intelligence based on mathematical and statistical methods. Its goal is to enable computers to learn from data without requiring explicit programming for each task [15].

Recent advancements in artificial intelligence (AI) and natural language processing (NLP) have enabled innovative methods to detect and counteract propaganda [16], [17]. These technologies analyze text patterns to identify manipulative strategies, such as extreme polarization and emotional persuasion [18]-[20]. Recent research investigates the potential of an

automated system to detect propaganda in real-time, aiming to enhance transparency in digital communication. Our research focused on detecting propaganda in texts on social media, aiming to identify content that sought to manipulate users' opinions and behaviors. A structured approach was employed, leveraging a manually annotated dataset of texts (tweets) sourced from specified social platform (Twitter as one of leading social media platforms), where content was categorized based on its propaganda characteristics and type. To prepare the data, various preprocessing techniques were applied enabling text normalization and the extraction of relevant information. Additionally, TF-IDF (Term Frequency-Inverse Document Frequency [21]) feature extraction was used to identify the most significant terms within propaganda texts. Our methodology proposed an effective process for text preprocessing and feature engineering, transforming a collection of raw documents into a numerical feature matrix. These features were further refined through various machine learning models, which supported the classification of texts as manipulative or authentic. The system's efficacy was evaluated based on precision, recall, and overall accuracy. The obtained results showed acceptable performance of our propaganda detection system, which successfully detected and correctly classified the majority of propaganda in the texts used for evaluation testing.

II. RESEARCH METHODOLOGY

We will hypothesize that it will be possible to distinguish propaganda texts from non-propaganda texts on social media by analyzing the specific linguistic features of these texts. To validate this hypothesis, we will use a supervised approach. The goal will be to build a classifier capable of categorizing propaganda texts into distinct categories, as well as identifying non-propaganda texts on social media. Our methodology follows a pipeline architecture in which a processing pipeline connects several transformation modules. Data pass through these modules sequentially, with the output of each module serving as the input for the next one [22]. In our research study, we have chosen to adopt a pipeline architecture as shown in Figure 1. The Different stages of our system pipeline: (1) Data collection (2) Data Preprocessing (3) Feature extraction (4) Classification (5) Evaluation. In the flowing subsections, we describe each stage of our system in detail.

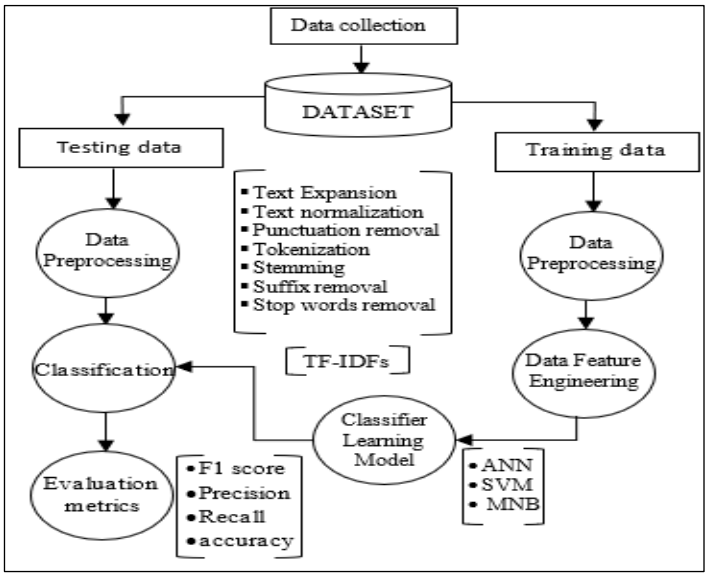

Figure 1: Pipeline Methodology Steps for a Textual Propaganda Classification System.
Authors, (2025).

II.1 DATASET COLLECTION

Given the unavailability of free access to social media databases, we took the initiative to create our own dataset. To achieve this, we adopted a manual approach to collect a substantial number of tweets on a timely and relevant issue that has become a prominent subject of propaganda on social media platforms. By leveraging targeted hashtags and keywords, we gathered a significant volume of tweets. After a thorough selection process, we retained a sample of one thousand tweets to ensure a representative analysis. To detect propaganda, tweets can be classified based on either content analysis or network analysis. In our research, we aimed to ensure that the corpus supports both classification methods to enable greater flexibility when designing the classifier. Accordingly, we gathered the necessary information to capture both the tweet content and the social network elements. Table 1 illustrates the structure of our corpus, which consists of twenty-one columns, each providing detailed information about the tweets. These columns are categorized into five categories of data Analysis: *Content Analysis, Network Analysis (Network Nodes and Network Locations), Technical Data Analysis, and Propaganda Nature Analysis*. As shown in Table1, in addition to the information about the tweets, we add columns to annotate the tweets according to their nature of propaganda, including: fake news, intimidation, concealment, manipulation and name calling. The created dataset support both classification methods, however, this paper focus on performing a classification based on content and leave the network aspect for future work.

Table 1: Column structure of the social media textual propaganda dataset.

Data Category	Column	Type	Description
Content Analysis	text of tweet	String	Full text of the tweet.
	hashtags	String	Hashtags included in the tweet.
	Views count	Integer	Number of views of the tweet.
	Retweets count	Integer	Number of retweets of the tweet.
	Likes count	Integer	Number of likes on the tweet.
	Responses count	Integer	Number of responses to the tweet.
Network Analysis	tweet author	String	Name of the tweet's author.
	profile	String	User's profile information.
	user followers	Integer	Tweet author's follower count
	user following	Integer	Tweet author's following count
	mention	String	Users mentioned in the tweet.
	retweet	String	Original tweet.
Technical Data Analysis	id	String	Unique tweet identifier.
	tweet link	String	URL of the tweet.
	date and hour	Date	Tweet's date and time.
	UserName	String	Tweet author's username
Propaganda Nature Analysis	fake news	Integer	Tweet contains false info?
	intimidation	Integer	Tweet contains intimidation?
	concealment	Integer	Tweet contains concealment?
	manipulation	Integer	Tweet contains manipulation?
	name calling	Integer	Tweet contains defamation?

Source: Authors, (2025).

II.2 DATA PREPROCESSING

There are various techniques for preprocessing textual data [21]. In our study, the goal of preprocessing is to prepare the text in a way that makes it easier to analyze and manipulate, facilitating the recognition and identification of propaganda types. To achieve this, we applied an automated preprocessing procedure to process both the textual data in our dataset. The preprocessing procedure executes the following seven operations in a single sequence:

(1) Expansion of Textual Contractions: To ensure clarity and optimal readability, certain text contractions are expended.
(2) Text normalization: To ensure the database is clear and readable, certain special characters are normalized.
(3) Punctuation removal: Removing commas, periods, and semicolons, as these marks are primarily used in written human communication.
(4) Tokenization: This is the process of dividing text into atomic units or tokens resembling words. This step aims to segment the text into individual words.
(5) Stemming: An automatic natural language processing technique that reduces words to their base or root form to ensure accurate representation of names and entities.
(6) Suffix removal: To ensure consistent, uniform, and precise normalization, specific suffixes such as "ist", "est", "less", "ian", and "en" are removed from proper nouns and specific entities.
(7) Stop words elimination: Stop words are removed because they play a minor role in classification tasks.

II.2 DATA FEATURE ENGINEERING

The pervious preprocessing step produces a unique vocabulary of terms present across the preprocessed documents. Following this, in order to evaluate the significance of each term within a document relative to the entire corpus, data features engineering is applied using Algorithm 1:

Algorithm 1: Feature Extraction from Textual Propaganda Data

```
1: Input: data
2: Start
3: pre-processed data ← pre-processing(data)
   vocabulary ← unique words from pre-processed data
   TF-IDFs ← []
4: For each document do
5:   For each term in vocabulary do
6:     Calculate TF (term frequency) in the current document
7:     Calculate IDF (inverse document frequency) for the term
       TF-IDF ← TF * IDF
       Add TF-IDF to TF-IDFs
8:   End For
9: End For
10: Convert TF-IDFs to a TF-IDF matrix
11: Return TF-IDF matrix
```

The proposed algorithm calculates the *TF-IDF* (**T**erm **F**requency-**I**nverse **D**ocument **F**requency) scores for each term in every document using the following equation:

$$TF - IDF\ \ (t, w, D) = TF(t, w) \times IDF(t, D)$$

- $TF(t,w) = o_t/n_w$, where o_t is the number of occurrences of term $\boldsymbol{t}$ in document $\boldsymbol{w}$, and n_w is the total number of terms in the document.
- $IDF(t,D): log(N/n)$, where $\boldsymbol{N}$ is the total number of documents, and $\boldsymbol{n}$ is the number of documents in which term $\boldsymbol{t}$ appears.
- $TF - IDF(t,w,D)$ Matrix of $TF - IDF$ scores are calculated for each term in each document. Each row represents a document, and each column represents a term from the vocabulary, with the corresponding $TF - IDF$ value

The resulting matrix is used for various natural language processing tasks such as text classification, information retrieval, or document similarity analysis.

II.3 CLASSIFICATION

To classify social network texts into categories (propaganda vs. non-propaganda, and various types of propaganda), we employ three supervised classification methods, well-known for their simplicity and efficiency: Artificial Neural Network (**ANN**) with Multi-Layer Perceptron [23], Support Vector Machine (**SVM**) [24], and Multinomial Naive Bayes (**MNB**)[25].

Before apply these classifiers, the data are splatted into training and testing sets. Then, we initialize the classifiers models using the training data in order to learn patterns in the TF-IDF scores associated with different document categories. Once trained, we use these models to predict the labels of the test data. Finally, to evaluate the models' performance, we calculate evaluation metrics such as accuracy, precision, recall, and F1-score.

Algorithm 2 describe the ANN classification. ANN classifier uses the TF-IDF matrix to detect patterns through multiple layers of perceptrons. These layers, including an input layer, one or more hidden layers, and an output layer, allow the network to learn complex relationships in the data. Texts are classified as propaganda or non-propaganda based on the patterns learned from the TF-IDF scores during training.

Algorithm 2: Artificial Neural Network (ANN) Classification.

Input: TF-IDF matrix, yyy
Begin
3. X←X \leftarrowX← TF-IDF matrix
4. Y←Y \leftarrowY← Class label vector
5. Split XXX and YYY into training and testing sets
6. Convert labels into a 1D array
7. Select and initialize the Multi-Layer Perceptron model
8. Predict the labels of the test data using the trained Multi-Layer Perceptron model
9. Calculate accuracy, precision, recall, and F1-score between the predicted labels and the true labels
10. Return the model's performance (accuracy, precision, recall, F1-score)
End.

Algorithm 3 describe the SVM classification, SVM uses the TF-IDF matrix to represent data in a multi-dimensional space, grouping texts of the same category close together. Categories are separated by a clear margin that is as wide and distinct as possible. New texts are classified as propaganda or non-propaganda based on which side of the margin they fall.

Algorithm 1: SVM Classification

Input: TF-IDF matrix, yyy
Begin
3. X←X \leftarrowX← TF-IDF matrix
4. Y←Y \leftarrowY← Class label vector
5. Split XXX and YYY into training and testing sets
6. Convert labels into a 1D array
7. Select and initialize the Support Vector Machine model
8. Predict the labels of the test data using the trained Support Vector Machine model
9. Calculate accuracy, precision, recall, and F1-score between the predicted labels and the true labels
10. Return the model's performance (accuracy, precision, recall, F1-score)
End

Algorithm 4 describe the MNB classification, MNB classifiers use the TF-IDF matrix with Bayes' theorem to estimate the probability of a text belonging to the propaganda or non-propaganda category. Despite assuming feature independence, MNB achieves reliable results with minimal training data, making it an efficient approach for text classification.

Algorithm 4: Naive Bayes Classification

Input: TF-IDF matrix, yyy
2. **Begin**
3. X←X \leftarrowX← TF-IDF matrix
4. Y←Y \leftarrowY← Class label vector
5. Split XXX and YYY into training and testing sets
6. Convert labels into a 1D array
7. Select and initialize the Multinomial Naive Bayes model
8. Predict the labels of the test data using the trained Multinomial Naive Bayes model
9. Calculate accuracy, precision, recall, and F1-score between the predicted labels and the true labels
10. Return the model's performance (accuracy, precision, recall, F1-score)
End

Algorithm 5 is designed to classify new texts. It uses the three supervised classification methods with the TF-IDF matrix to classify texts as propaganda or not, while also considering the probability predicted by the models for more precise and nuanced classification.

III. MATERIALS AND METHODS

In our work, we used Python to implement our propaganda detection system on the X platform. Python stands out as a powerful and elegant programming language, offering easy readability and comprehension. It shares most of its features with many other languages, making it a versatile tool for real-world applications.

Additionally, being open-source, it benefits from a unique standard implementation and a welcoming, extensive developer community [67]. Several Python libraries were utilized to develop the propaganda detection system, supporting tasks from data preprocessing to model evaluation and interface development:

Algorithm 5: Machine Learning for Propaganda Text Classification

Input: text, yyy, threshold
Begin
3. Train the classification model on the TF-IDF matrix and training labels
4. Preprocess the input text
5. Engineer features from the preprocessed text
6. Predict the label of the engineered features using the trained classification model
7. Use the prediction with probability
8. If the probability >>> threshold, then
9. **Propaganda**, probability of the class
10. Else
11. **Not propaganda**
12. End If
End

1. **NumPy**: NumPy is a Python library specializing in array manipulation, including multidimensional arrays and matrices, as well as mathematical functions to operate on them [68]. It was used to convert target data into one-dimensional arrays, ensuring compatibility with the machine learning models in scikit-learn.
2. **NLTK (Natural Language Toolkit)**: NLTK is a suite of Python modules dedicated to natural language processing research and teaching. It provides tools, datasets, and tutorials to facilitate linguistic analysis and algorithm development [69]. It was used for natural language preprocessing tasks such as tokenization, stemming, and stopword removal. This preprocessing helped simplify the analysis of text in the project.
3. **Scikit-learn**: Scikit-learn is an open-source Python library for machine learning that offers tools for classification, regression, clustering, dimensionality reduction, and more [70]. It was employed to evaluate model performance using metrics such as accuracy, precision, F1-score, and recall. It was also used to split datasets and build models, particularly with the MLPClassifier.
4. **Pandas**: Pandas is a Python library designed for data manipulation and analysis, particularly for tabular data like spreadsheets and relational databases [71]. It was used to efficiently and consistently read and handle our dataset, which was stored in a CSV file.
5. **Flask**: Flask is a lightweight web development framework for Python, known for its simplicity and flexibility [72]. It was utilized to develop the user interface for the propaganda detection system, providing a simple and accessible front-end for interaction.
6. **CSV Format**: The CSV (Comma-Separated Values) format is a type of plain text file used to store tabular data, where each row represents a record, and fields are separated by a delimiter, typically a comma [73]. It was used to store the dataset in a tabular format, with semicolon delimiters. This format allowed easy manipulation and seamless import of data into our program.

Each of these tools played a critical role in achieving the project's objectives, contributing to data processing, model building, and user interface design. By combining their functionalities, the propaganda detection system was efficiently implemented. Furthermore, we integrated our Python code into a local web page using Flask for communication. The user interface is built with HTML, CSS, JavaScript, and Bootstrap, providing an interactive and smooth user experience.

IV. RESULTS AND DISCUSSIONS

IV.1 SYSTEM USER INTERFACE OVERVIEW

The propaganda classification system was implemented as web-based platform. Our site includes an intuitive homepage that allows quick access to the next page (see Figure 2). The propaganda detection interface includes an area to enter text, a detection button, a button to display more details, and another button to return to the homepage (see Figure 3).

Figure 1: Screenshot : propaganda detection system homepage.
Source: Authors, (2025).

Figure 2: Screenshot: the propaganda detection interface.
Source: Authors, (2025).

IV.2 EVALUATION RESULTS

To classify the text as propaganda or not, as well as to determine the specific categories of propaganda among the five defined, we used an MLPClassifier model. This model was initialized with two hidden layers of sizes 100 and 50, a relu activation function, and an "adam" solver. We partitioned our data with an 80/20 ratio and set the seed to 42 to ensure the reproducibility of the results. We utilized the predict_proba method from scikit-learn classification models to obtain an array of probabilities for each class.

This method provides the likelihood associated with each class for every text instance, enabling us to evaluate the model's confidence in its predictions for each specific propaganda category. In our approach, and after extensive testing, we defined specific thresholds for each propaganda category to fine-tune the model's sensitivity and determine whether a text qualifies as propaganda.

For instance, the threshold for the Fake News category was set at 0.0333, while Intimidation required a higher threshold of 0.108. Similarly, Concealment had a threshold of 0.004, Manipulation was set at 0.008, and Name Calling at 0.005. These thresholds ensure that texts with classification probabilities below the specified values for each category are not classified as belonging to that propaganda type, enhancing the precision and reliability of our classification system.

We conducted a comprehensive evaluation by comparing the performance of selected classifier algorithms, including the ANN, SVM and MNB classifier. Through this evaluation, we used a variety of performance metrics for each model: F1 score, precision, recall, and accuracy. In the following, we present the classification results through various tables (Table 2, Table 3, Table4 et Table 5) and figures (Figure 4, Figure 5, Figure 6 and Figure 7).

In the comparison tables below, we detail the performance of each classifier for each propaganda category. The obtained results clearly demonstrate the superiority of the ANN model for our specific context. This meticulous evaluation process allowed us to make an informed decision regarding the choice of the propaganda classification model in five distinct categories : fake news, intimidation, dissimulation, manipulation and name calling.

Table 2: Accuracy of classifiers by propaganda category.

Accuracy of :	ANN	SVM	Naive bayes
fake news	0.72	0.71	0.70
intimidation	0.69	0.67	0.66
dissimulation	0.78	0.77	0.77
manipulation	0.72	0.70	0.70
name calling	0.63	0.61	0.61

Source: Authors, (2025).

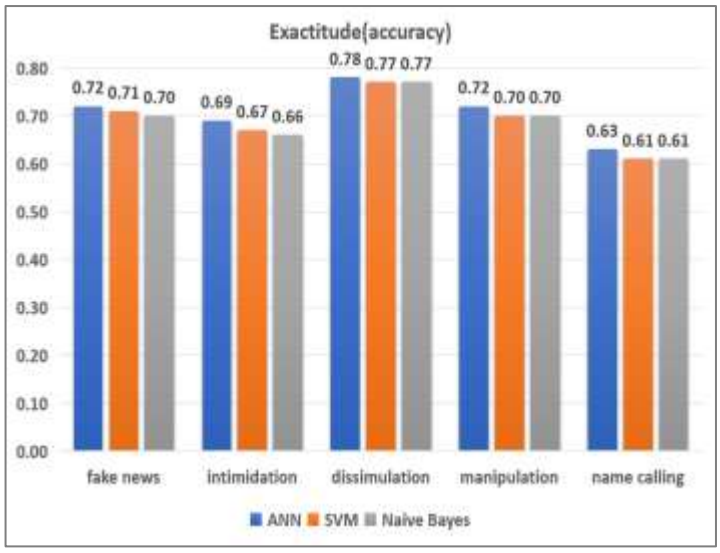

Figure 4: Accuracy comparative representation.
Source: Authors, (2025).

Table 3: Precision of classifiers by propaganda category.

Precision of:	ANN	SVM	Naive bayes
fake news	0.70	0.62	0.67
intimidation	0.63	0.52	0.62
dissimulation	0.67	0.66	0.66
manipulation	0.67	0.65	0.65
name calling	0.61	0.59	0.58

Source: Authors, (2025).

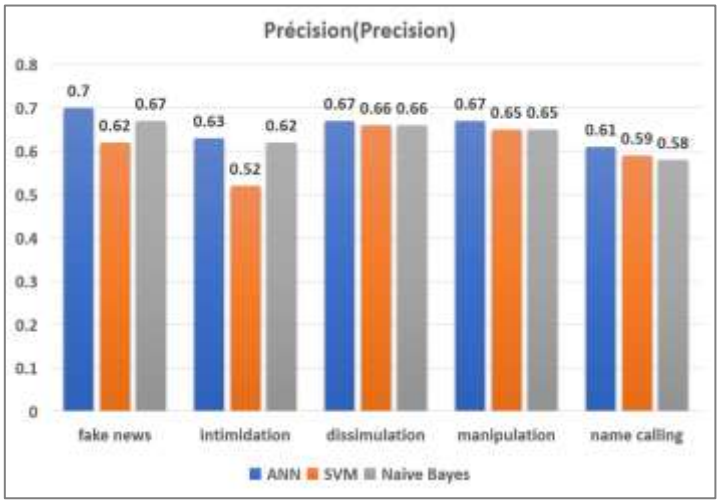

Figure 5: Precision Comparative representation.
Source: Authors, (2025).

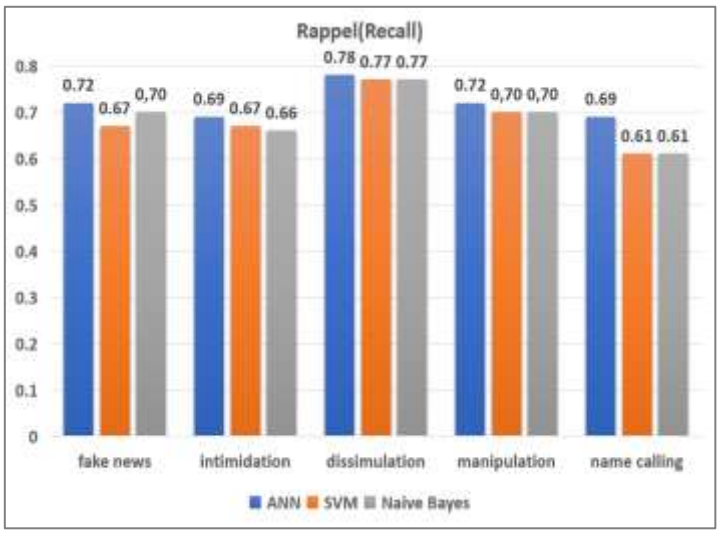

Figure 6: Recall Comparative representation.
Source: Authors, (2025).

Table 4: Recall of classifiers by propaganda *category*.

Rappel :	ANN	SVM	Naive bayes
fake news	0.72	0.67	0.70
intimidation	0.69	0.67	0.66
dissimulation	0.78	0.77	0.77
manipulation	0.72	0.70	0.70
name calling	0.69	0.61	0.61

Source: Authors, (2025).

Table 5: F1 Scores of classifiers *by propaganda category*.

F1-score :	ANN	SVM	Naive bayes
fake news	0.71	0.70	0.68
intimidation	0.66	0.64	0.63
dissimulation	0.71	0.70	0.70
manipulation	0.69	0.67	0.67
name calling	0.61	0.58	0.58

Source: Authors, (2025).

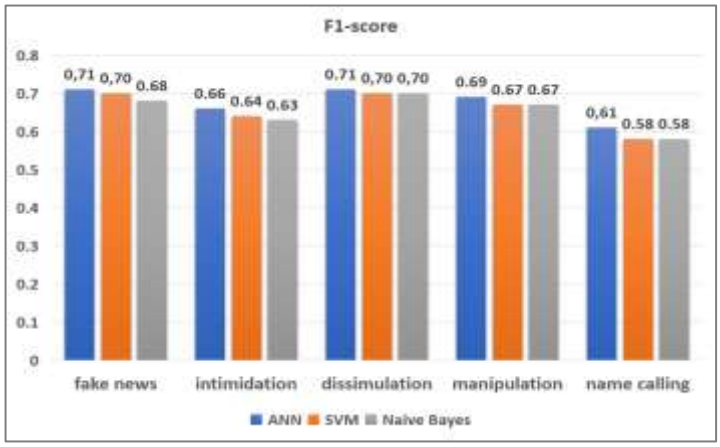

Figure 7: Comparative representation F1-score.
Source: Authors, (2025).

IV.3 DISCUSSION AND RELATED WORK

The results of our study demonstrate that the ANN-based propaganda classification model performs well, achieving an accuracy of around 70%, which indicates a robust capacity for detecting propaganda in text-based content. However, accuracy can still be improved. Our approach effectively identifies nuanced manipulation techniques through the use of tailored thresholds for classification, which gives it an edge in adaptability compared to other models.

When comparing our work to previous studies, several key differences emerge. In [26], authors focuses on detecting COVID-19-related propaganda using an improved ANN and hybrid feature engineering, achieved 77.15% accuracy. However, its reliance on binary classification and focus on specific hashtags restricts its ability to capture the full range of propaganda types. In [27], SVM and MNB classifiers are used to distinguish propagandist from non-propagandist text, achieved an accuracy of 69.2%. While the study employed a hybrid approach to feature engineering, its reliance on news-based datasets and a binary classification model limits its applicability to diverse social media content. In [28], authors introduced a framework combining TF-IDF and sentiment analysis for propaganda detection, achieving 69.2% accuracy. However, its binary classification model and focus solely on textual data limit its ability to capture the complex, multi-dimensional nature of modern propaganda. In [29], to recognize propaganda on Social Networks, authors explore machine learning models like SVM, Random Forest, and advanced deep learning techniques such as RoBERTa, primarily focuses on theoretical aspects and lacks practical implementation, methods, or results.

In contrast to other studies [26-28], our approach categorizes various forms of digital manipulation, employing tailored thresholds and adapting effectively to a wider range of propaganda types. This makes our approach more versatile and capable of identifying nuanced manipulation techniques across diverse social media platforms. Furthermore, while authors in [29] presented a theoretical proposition without actual implementation or results, our study provides a concrete methodology with data collection, preprocessing, feature extraction, and model evaluation, demonstrating the practical application and effectiveness of our system in detecting propaganda.

V. CONCLUSIONS

Propaganda, while often used to communicate ideas and messages, carries significant risks when exposed excessively, especially when it involves manipulative techniques and

aggressive rhetoric. These risks include misinformation, manipulation of opinions, and other forms of deception, all of which can compromise objective understanding, erode trust in institutions, and destabilize democratic processes.

The main objective of our study was to develop an accessible tool to help users distinguish between information and propaganda on social media. We proposed a classification system that detects propaganda by analyzing its linguistic characteristics. Through theoretical research and the use of preprocessing techniques like tokenization, stemming, and stop-word elimination, we created a corpus of manually annotated tweets from the social platform X (Twitter) for training our classifier. Our system, which employed supervised classification algorithms, achieved an accuracy of approximately 70%. While further improvements remain necessary, the proposed approach, if integrated into web platforms and social networks, could play a crucial role in identifying and mitigating digital manipulation, thereby fostering greater transparency, accountability, and trust in digital communication.

For future work, a larger datasets and more in-depth testing are essential for enhancing its accuracy. Additionally, integrating network analysis would improve its ability to differentiate between propaganda and non-propaganda content. Future developments should also focus on adapting the system for real-world use, allowing internet users to verify the accuracy of the information they encounter, thus strengthening the ability to classify and address propaganda in social media environments.

VI. AUTHOR'S CONTRIBUTION

Conceptualization: Belkacem mostefai and Tarek Boutefara.
Methodology: Belkacem mostefai and Tarek Boutefara.
Investigation: Abid Chahinez and Aberkane Marwa.
Discussion of results: Belkacem mostefai,Tarek Boutefara, Abid Chahinez and Aberkane Marwa.
Writing – Original Draft: Belkacem mostefai**.**
Writing – Review and Editing: Belkacem mostefai and Tarek Boutefara.
Resources: Tarek Boutefara.
Supervision: Belkacem mostefai and Tarek Boutefara.
Approval of the final text: Belkacem mostefai and Tarek Boutefara

VII. REFERENCES

[1] Y. Zhu and K.-w. J. D. J. Fu, "How propaganda works in the digital era: soft news as a gateway," vol. 12, no. 6, pp. 753-772, 2024.

[2] P. Baines, N. Snow, and N. OâÂ Â²Shaughnessy, "The Sage handbook of propaganda," 2019.

[3] D. J. P. A. Q. Walton, "What is propaganda, and what exactly is wrong with it," vol. 11, no. 4, pp. 383-413, 1997.

[4] M. Alhouseini and M. Saaideh, "The Influence of Social Media on Contemporary Global Politics," 2023.

[5] S. Okafor and J. Asogwa, "USE OF SOCIAL MEDIA AS A TOOL FOR POLITICAL PROPAGANDA IN ENUGU METROPOLIS."

[6] J. J. P. Chan and S. Criticism, "Online astroturfing: A problem beyond disinformation," vol. 50, no. 3, pp. 507-528, 2024.

[7] D. Sotiraki, "Media Framing by The Washington Post on TikTok: Shaping Public Perception of the Ukraine Conflict," ed, 2023.

[8] C. Ghosh, "The Impact of Social Media on Conflict Perception: Case Studies of Russia-Ukraine and Gaza Conflicts."

[9] A. J. J. o. D. Juneström, "An emerging genre of contemporary fact-checking," vol. 77, no. 2, pp. 501-517, 2021.

[10] A. Acht, "Who is benefitting from fact-checking on social media–user or platform? Examining the impact of different fact-checking approaches on social media platforms on user's perception of trust," 2024.

[11] M. P. J. M. i. s. C. E. P. Goswami, "Fake News and Cyber Propaganda: A study of manipulation and abuses on Social Media," pp. 535-544, 2018.

[12] B. J. P. d. l. i. García-Orosa, "Disinformation, social media, bots, and astroturfing: the fourth wave of digital democracy," vol. 30, no. 6, 2021.

[13] M. Orabi, D. Mouheb, Z. Al Aghbari, I. J. I. P. Kamel, and Management, "Detection of bots in social media: a systematic review," vol. 57, no. 4, p. 102250, 2020.

[14] E. Spencer, Narrative Control: Media Bias, Censorship, and Elections 2024: The Fight for Truth Amidst Media Narratives and Election Propaganda. Centurion Press, 2024.

[15] L. Rouvière, "Apprentissage supervisé-Machine learning," 2022.

[16] G. D. S. Martino, S. Cresci, A. Barrón-Cedeño, S. Yu, R. Di Pietro, and P. J. a. p. a. Nakov, "A survey on computational propaganda detection," 2020.

[17] B. M. Almotairy, M. Abdullah, and D. Alahmadi, "Detection of Computational Propaganda on Social Networks: A Survey," in Science and Information Conference, 2023, pp. 244-263: Springer.

[18] J. J. I. r. o. s. Serrano-Puche, "Digital disinformation and emotions: exploring the social risks of affective polarization," vol. 31, no. 2, pp. 231-245, 2021.

[19] V.-A. Oliinyk, V. Vysotska, Y. Burov, K. Mykich, and V. Basto-Fernandes, "Propaganda Detection in Text Data Based on NLP and Machine Learning," in MoMLeT+ DS, 2020, pp. 132-144.

[20] P. N. Ahmad, L. Yuanchao, K. Aurangzeb, M. S. Anwar, Q. M. u. J. S. O. C. Haq, and Applications, "Semantic web-based propaganda text detection from social media using meta-learning," pp. 1-15, 2024.

[21] F. J. U. h. w. l. c. g. t.-i.-t.-f.-i.-d. Karabiber, "Tf-idf—term frequency-inverse document frequency," 2020.

[22] H. Hapke and C. Nelson, Building machine learning pipelines. O'Reilly Media, 2020.

[23] A. Rana, A. S. Rawat, A. Bijalwan, and H. Bahuguna, -"Application of multi layer (perceptron) artificial neural network in the diagnosis system: a systematic review," in 2018 International conference on research in intelligent and computing in engineering (RICE), 2018, pp. 1-6: IEEE.

[24] D. A. Pisner and D. M. Schnyer, "Support vector machine," in Machine learning: Elsevier, 2020, pp. 101-121.

[25] F. Sabry, Naive Bayes Classifier: Fundamentals and Applications. One Billion Knowledgeable, 2023.

[26] A. M. U. D. Khanday, B. Bhushan, R. H. Jhaveri, Q. R. Khan, R. Raut, and S. T. J. M. I. S. Rabani, "NNPCov19: Artificial Neural Network-Based Propaganda Identification on Social Media in COVID-19 Era," vol. 2022, no. 1, p. 3412992, 2022.

[27] A. M. U. D. Khanday, Q. R. Khan, and S. T. J. B. S. J. Rabani, "Detecting textual propaganda using machine learning techniques," vol. 18, no. 1, pp. 0199-0199, 2021.

[28] A. M. U. D. Khanday, M. A. Wani, S. T. Rabani, Q. R. Khan, and A. A. J. P. o. Abd El-Latif, "HAPI: An efficient Hybrid Feature Engineering-based Approach for Propaganda Identification in social media," vol. 19, no. 7, p. e0302583, 2024.

[29] R. R. Yellu, S. B. Dodda, B. K. Sharma, S. Dhar, and S. J. L. P. I. Temara, "A Machine Learning Approach to Recognising Propaganda on Social Networks," vol. 44, no. 3, pp. 3184-3190, 2024.