Mars: Jurnal Teknik Mesin, Industri, Elektro dan Ilmu Komputer
Volume 3 Nomor 1, Tahun 2025
e-ISSN : 3031-8742, dan p-ISSN : 3031-8750, Hal. 239-249
DOI: https://doi.org/10.61132/mars.v3i1.713
Available online at: https://journal.arteii.or.id/index.php/Mars

Fuzzy logic Deep Learning Control System for Detecting Arabic Tweets Spam based on Large Language Models

Ghosoon K.munahy
Department of Information Technology, College of Computer Science and Information Technology, University of Kerbala, Kerbala, Iraq. 0009-0007-4542-173X
Email : ghosoon.k@uokerbala.edu.iq

***Abstract**. spam is posting unsolicited messages or advertising on social media, particularly Twitter. These messages are normally designed to sell specific products and services or links. In this research, we developed a fuzzy control system to detect Arabic spam tweets based on deep learning with a large language model. Initially, we performed text cleaning and further transformed text into vectors with the help of AraGpt and AraBert. Subsequently, we employed a multi-layer perceptron network model in feature extraction of essential features. Finally, we adopted the fuzzy logic control system for classifying spam tweets using features filtered from deep networks. Employing the proposed Fuzzy logic control system provided nearly a 100% comparative to only utilizing the deep neural networks, which yielded an almost 99% throughput for both large language models Aragpt and Arabert, with a 100% F1 score for the Aragpt model and 99% for Arabert model respectively.*

***Keywords**: Spam Detecting, Arabic Tweets, Fuzzy Logic, Deep Learning, LLM*

1. INTRODUCTION

Twitter is a popular social media platform often targeted by scammers and spammers who post false information and malicious links. Researchers have developed multiple machine-learning models that can detect and prevent Twitter spam to combat this. Arabic, a member of the Semitic language family, is spoken in over 20 countries across Africa and the Middle East. Fuzzy logic offers a flexible approach to handling uncertainty and vagueness in decision-making processes. It bridges the gap between strict binary logic and the complexities of real-world scenarios. Researchers have explored its applications in optimisation problems, decision support, and more. IF-THEN rules are one of the applications of fuzzy logic for modeling knowledge in computers where Words are used instead of precise values to model these rules, making it closer to human reasoning, which does not require exact rules for decision-making in most daily activities. As can be inferred, this type of rule can be directly taken from experts to develop computer-assisted detecting tools. Shallow learning techniques and more intricate deep learning frameworks are balanced by simpler architectures like multilayer perceptrons (MLPs). These neural networks have been essential, especially for the initial developments of word embedding methods. In tasks such as text classification, MLPs have consistently shown themselves to be highly effective as standalone classifiers despite their relative simplicity. To capture complex word relationships, MLPs typically treat input text as an unordered collection of words and use methods like TF-IDF or modern word embeddings. MLPs are very helpful in many different natural language processing applications due to their adaptability, often outperforming more complex architectures. The ability of large language models (LLMs) to

Received: Desember 30, 2024; Revised: Januari 20, 2025; Accepted: Februari 10, 2025; Published: Februari 12, 2025;

convert text into vector representations—also referred to as embeddings—has grown significantly. These embeddings are helpful for many natural language processing (NLP) tasks, including text classification, sentiment analysis, and question answering, because they capture the semantic relationships between words and phrases. In contrast to conventional methods such as bag-of-words or term frequency-inverse document frequency (TF-IDF), LLMs produce contextualized embeddings through deep neural networks, providing more complex and insightful language representations. Better generalization across various domains has been made possible by this process, dramatically advancing NLP**.**In this paper, we propose a fuzzy logic control system based on a deep neural network for detecting spam tweets.

2. RELATED WORK

Arabic tweets are Classified using classical machine learning algorithms such as support vector machine, logistic regression, and naive Bayesian algorithm, as well as neural networks including GloVe and fastText. The tweets were collected and manually labeled, and features such as N-grams were extracted to build the model. The results showed that the neural network outperformed other algorithms, and GloVe was more effective than fast Text in deep-learning neural networks. A DNFN is developed by using the ChSO algorithm for spam detection in SMS over social media platforms. The method of the authors analyses input data using the Box-Cox transformation to bring it closer to the normal distribution, and when using feature fusion, they use the Tversky index in combination with deep quantum neural networks based on ChSO—Deep Quantum Neural Network (DQNN). The Deep Neuro-Fuzzy Network (DNFN) is trained with the ChSO algorithm to detect spam in the context of the social medium Twitter. This approach shows a better result in identifying spam posts using the fusion of optimal features effectively.

The authors Worked on constructing and assessing different machine learning models, including the modeling of classifying YouTube comments as spam or ham. These include the Gaussian Naive Bayes model, logistic regression, KNN, SVM, MLP, decision tree, Random forest, Voting classifier, and more. Performance was evaluated based on precision, accuracy, and AUC-ROC, and Random Forest was almost impeccable with 100% precision and an AUC-ROC of 0.9841.

Employed a deep learning-based approach to detect Twitter spammers using two classifiers: One text-based classifier that has been trained is the classifier trained from the syntax of tweets which has used Word2Vec; the second is a combined classifier that considers both content and tweets metadata. The authors employed the Social Honeypot dataset and the

1KS−10KN dataset stating that text and user features should both be used for better classification. used multiple machine learning methods with report.csv dataset on email spam detection, where Multinomial Naïve Bayes was found to be the most effective However, ensemble methods were reported to be useful to minimize weakness, such as class-conditional independence.

in his work used the UtkML Twitter spam dataset to demonstrate how preprocessing methodology is crucial in detecting spam tweets. Measures that were found to accustom were TF-IDF vectorization, eliminating special character marks, and removing stop words which enhanced the result of the SVM classifier from 91% to 93%.

In their research developed a cost-sensitive approach to filter spam in social network. In their study they used Evolutionary NOn-dominated Radial slots-based technique (ENORA) aimed at feature selection and regularized deep neural networks (RDNNs) during the base learners model in the cost-sensitive ensemble learning framework. This approach also decreased the amount of features and the misclassification cost which enhances the spam detection on sets from Twitter and Hyves. puts forward Fuz-Spam, a label smoothing-based fuzzy detection technique for spam detection. For features combining, it uses deep representations, and for converting the label spaces, it implements generative adversarial learning. As experiments with real datasets indicate, their proposed method increases detection efficiency by 10-20% compared with previous approaches and shows stability . This work assesses Flan-T5, RoBERTa, and SetFit base on part first or few shot learning in spam message classification. It is apparent that the presented models offer very high accuracy in the spam detection, especially after using relatively small training set, which is highly suitable for the dynamic nature of spamming strategies. This research covers the employment of Zero-Shot Learning in spam detection using GPT-4 and FLAN-T5. The approach is based on the fact that those models, after training on large texts datasets, are able to classify spam messages without explicitly training on the data set. This method enables classification where there is scarce or no labeled data at all in the dataset at hand.This work outlines an improved model based on BERT to address the problem of spam and phishing emails. Balanced and imbalanced data datasets are handled, and spam email classification reveals general high performance as well as boosts in accuracy. It clears out the differentiation of between phishing and non-phishing emails hence increasing the overall performance as far as the classification accuracy is concerned. [20]the propose a spam detector that utilises the BERT pre-trained model, which assesses the content of emails and messages. Thus the model was trained on several corpuses including the SMS Spam Collection corpuses, Enron, SpamAssassin and Ling-Spam datasets.

The study attained striking performance reaching F1-scores of between 97% and 99% on the datasets(ar5iv)

3. METHODOLOGY

Dataset and Preprocessing

Arabic text processing presents unique challenges due to the language's inherent complexities (ambiguity, diglossia, script characteristics), normalization issues (diacritics, dialects, letter variations), and the intricacies of NLP tools (tokenization, stemming/lemmatization). These challenges demand ongoing research and development to advance the field and create practical Arabic NLP tools and techniques.in this paper, we apply some preprocessing techniques like normalization, tokenization, and stop-word removal using a Python library such as and NLTK.

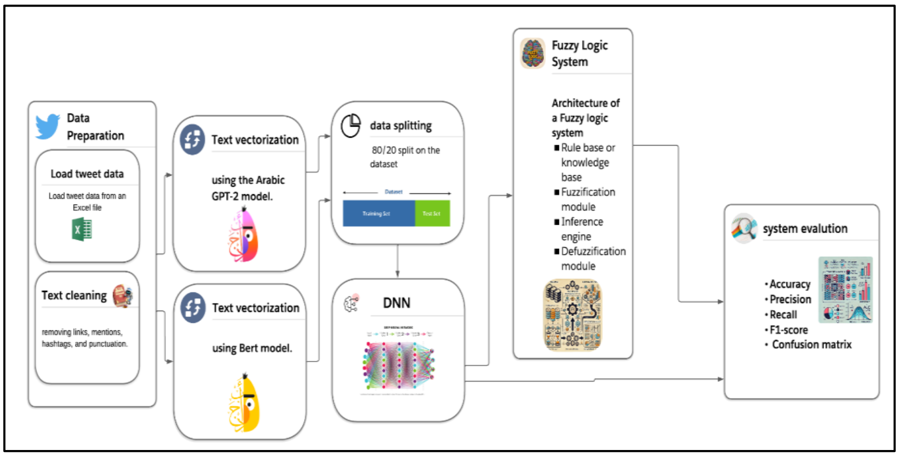

Figure 1. The system phases

Text Vectorization:

Since machines cannot process text directly, text vectorization is essential for NLP tasks. Machine learning algorithms can convert text into numerical vectors to perform classification, sentiment analysis, and machine translation tasks. We utilized two methods for text vectorization: one based on aragpt2 and the other based on the araBert model.

- AraGPT 2: Pre-trained AraGPT2 is a model that can be used directly for text vectorization since it has already learned to represent words and phrases in a high-dimensional space. To vectorize text using AraGPT2, the first step is to break the text into individual words or tokens. Next, AraGPT2 obtains a numerical representation (embedding) for each token. Finally, these embeddings are combined to get a single vector representation for the entire text.

- AraBert model: AraBERT is a powerful language model that generates numerical representations, known as embeddings, from text, capturing both syntactic and semantic meaning. To vectorize text using BERT, we begin by breaking down the text into smaller units and then feeding these tokens into the BERT model to obtain their respective embeddings. Afterward, we combine these embeddings to produce a single vector representation for the entire text.

Deep Neural Network:

This research used an artificial neural network with multiple layers between the input and output. The input layer receives data, the hidden layers learn complex patterns, and predictions are made by the output layer—which frequently uses a sigmoid function for binary classification. Each layer's neurons are linked to weights modified via backpropagation during training. Non-linearity is introduced by activation functions such as ReLU, which enable the network to capture more intricate relationships. Techniques like Selective Gradient Dropout, in which some neurons are arbitrarily ignored during training, are employed to prevent overfitting. Figure refers to the deep neural network architecture used in our system.

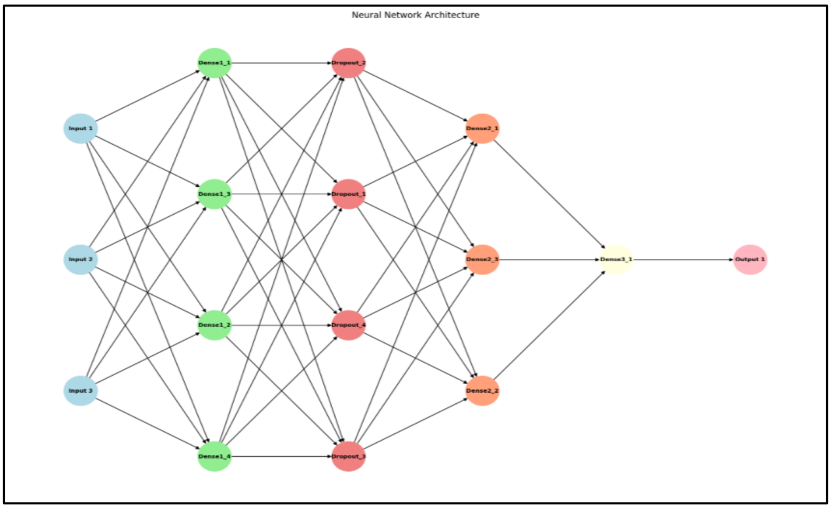

Figure 2. Deep neural network architecture

Logic Fuzzy System

During this step, the system utilizes the derived feature vectors from tweets, employing a pre-trained deep-learning model. This approach produced a numerical representation of tweets, including significant textual attributes like structure, linguistic trends, and essential terminology. The collected feature vectors are subsequently input into a fuzzy logic control system to enhance the decision-making process by providing interpretability and flexibility. In the fuzzy logic system, input membership functions are established to classify the attributes as "low," "medium," or "high." Triangular membership functions delineate these groups, with low characteristics represented by the interval [0, 0, 0.5], medium features by [0, 0.5, 1], and high features by [0.5, 1, 1]. In this way, the output of the fuzzy logic algorithm is qualified as “low”, “medium” or “high” which shows the possibility of the given tweet to be classified as “ham” or “spam”. A set of fuzzy control rules then further controls the relationship between the input feature categories and the output. These regulations are explicitly formulated as follows: - Rule 1: A low feature value is associated with a low Output, which normally means “ham”. - Rule 2: If the characteristic is moderate, the output is moderate (which can be interpreted as moderate probability of ‘spam’. - Rule 3: This result indicates that high feature value makes a high output level (which seems to represent “spam”). These regulations enable the use of a fluent, step-by-step approach to decision-making, by the system, making it easier to put in to operation when dealing with variations in the feature values. They both weigh and assess the rules that this fuzzy logic system applies to each feature vector of the given Tweet. The ultimate categorization is dictated by the system's output: if the result is greater than 0.5 then it labeled as “spam tweet” otherwise it labeled as “ham” I implement an integration of the ability of the deep learning model to discover detailed characteristics of the image and the self-organized fuzzy logic system to make rule-based decisions resulting in an enriched classification model that is also easier to understand. As the current work demonstrates, this approach enhances classifiability and provides decision-performing information, making it a good middle ground between assets demanding high model efficiency and interpretability.

4. RESULTS AND DISCUSSION

This section will evaluate the outcome of the three models applied to classifying the tweets as ham and spam. Based on the assessment of performance metrics such as accuracy, recall, and F1-score, the strengths and weaknesses of each approach will be discussed. The performance table shows the performance evaluation of three used methods.

Initially, we employed a neural network model for classification and the large language model AragPt for text vectorization. Large language models have shown to be very effective at comprehending textual context and translating it into accurate representations, according to a recent study on their application in natural language processing. This method demonstrated exceptional efficacy in classifying accurate tweets (Ham), with high accuracy and 100% recall, indicating that the model successfully identified the correct tweets with few mistakes. The classification accuracy for spam tweets was high, although the recall was 96%, suggesting that the model incorrectly detected specific spam tweets. The confusion matrix for this method in

Figure indicates that 2 legitimate communications (Ham) were erroneously categorized as spam, whereas 14 spam tweets were incorrectly identified as legitimate.

In the second time, we also used a deep neural network but with Arbbert to encode the texts, According to, BERT-based models perform well on a variety of natural language processing tasks, such as text classification. This approach also performed well, but slightly less than the first approach. The recall accuracy for correct tweets (ham) was 100%, but the recall for spam tweets was 95%, which reflects a slight decrease compared to the first approach. From the confusion matrix of this approach figure, it is clear that 7 correct tweets were misclassified as spam, while 19 spam tweets were misclassified as correct. This suggests a slightly higher error rate for this method than the first one. Last but not least, we employ a fuzzy classification system with deep neural network feature extraction on texts transformed into vectors using large language models. According to , fuzzy systems offer flexibility in handling ambiguous or complex data, which makes this a valuable method for text classification. These vectors indicate the dependency of words in the texts, so the learner will be in a good position to contextualize the discourses accordingly. The extracted representations were fed to the neural network to produce a set of features pertinent to these representations, and the fuzzy system classified the texts in terms of flexible affiliation rather than the categorical membership degree. This integrated approach ensured that the combined model yielded the best outcome of high accuracy of a neural network and the flexibility of the fuzzy classification.

Table 1. The performance evaluation of three used methods

Method	Precision	Recall	F1-Score
DNN+ Aragpt	0.99	0.98	0.99
DNN+ AraBert	0.99	0.97	0.98
DNN+ Fuzzy Logic+ LLM	1.00	1.00	1.00

5. CONCLUSION

In this research, the strategy to differentiate the text tweets into valid tweets (Ham) and the spam tweets was developed with the help of modern techniques such as deep learning and fuzzy systems integration. Good results were obtained as the third model offered perfect performance as a neural network for feature extraction employed in conjunction with a fuzzy system for classification following the conversion of texts to vectors through large language models. Maintaining this combination of advanced techniques, the texts were well-processed and correctly classified. The current research evaluated and reiterated the effectiveness of implementing such algorithms as fuzzy systems with neural networks and large language models that can complement the enhanced nature of classification systems concerning given data's ambiguity and linguistic aspects. This work can be generalized for harder applications or implemented with new algorithms to enhance it further. Figures [5,6,7,8] show Performance Evolution Over Epochs and classification metrics chart, respectively, for the first and second methods

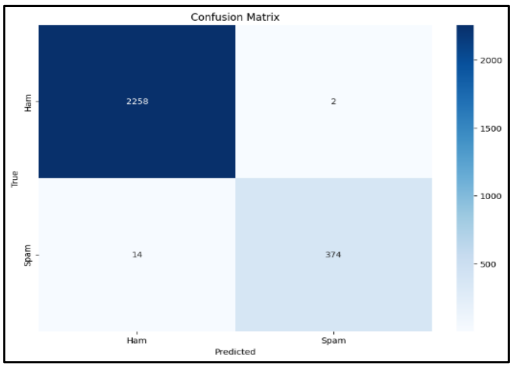

Figure 3. DNN+ AraGpt confusion matrix

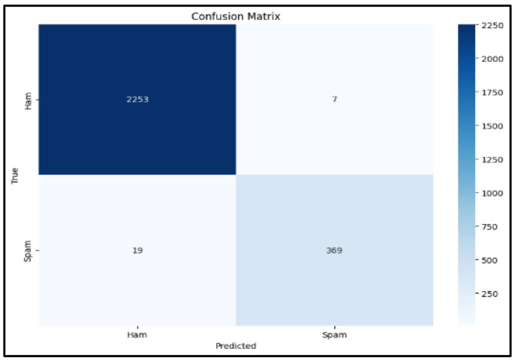

Figure 4. DNN+ AraBert confusion matrix

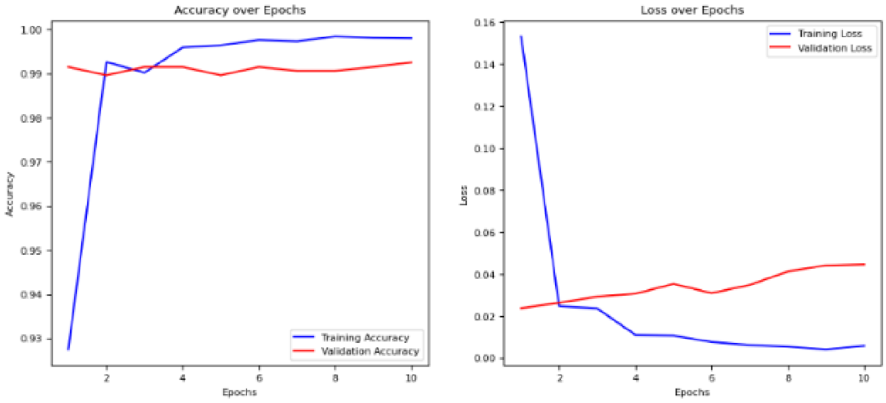

Figure 5. DNN+ Aragpt Performance Evolution Over Epochs

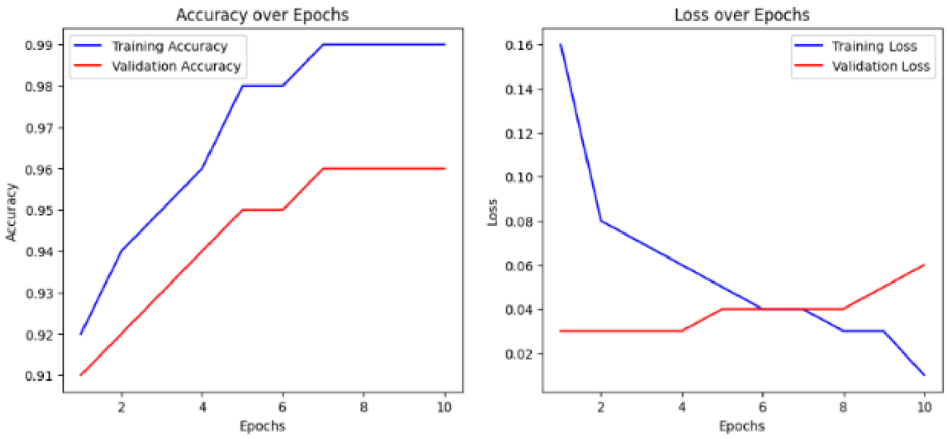

Figure 6. DNN+ AraBert Performance Evolution Over Epochs

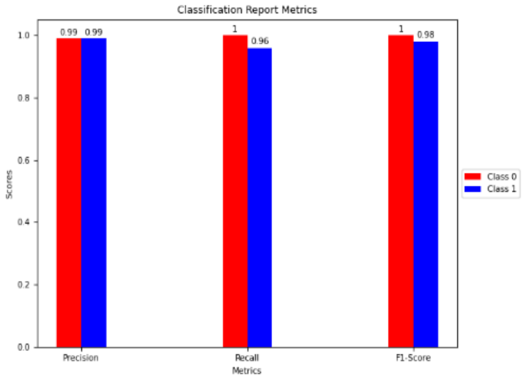

Figure 7. DNN+ AraBert classification metrics

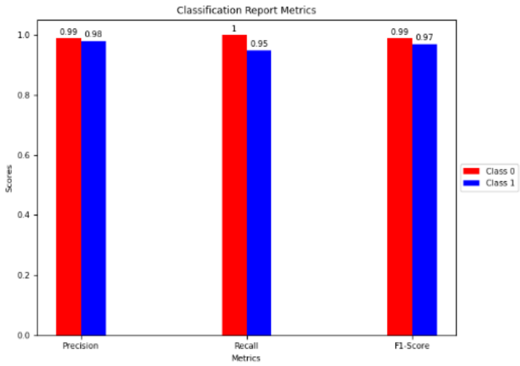

Figure 8. DNN+ AraGpt classification metrics

REFERENCE

Alfaidi, A., Alwadei, H., Alshutayri, A., & Alahdal, S. (2023). Exploring the performance of Farasa and CAMeL taggers for Arabic dialect tweets. *International Arab Journal of Information Technology*, *20*(3), 349–356.

Alom, Z., Carminati, B., & Ferrari, E. (2020). A deep learning model for Twitter spam detection. *Online Social Networks and Media*, *18*, 100079.

Antoun, W., Baly, F., & Hajj, H. (2020). AraBERT: Transformer-based model for Arabic language understanding. *arXiv preprint arXiv:2003.00104.*

Antoun, W., Baly, F., & Hajj, H. (2020). AraGPT2: Pre-trained transformer for Arabic language generation. *arXiv preprint arXiv:2012.15520.*

Avgerinos, C., Vretos, N., & Daras, P. (2023). Less is more: Adaptive trainable gradient dropout for deep neural networks. *Sensors*, *23*(3), 1325.

Barushka, A., & Hajek, P. (2020). Spam detection on social networks using cost-sensitive feature selection and ensemble-based regularized deep neural networks. *Neural Computing and Applications*, *32*(9), 4239–4257.

Bird, S. K., & Loper, E. (n.d.). Natural Language Toolkit (NLTK). *University of Pennsylvania*. Retrieved from https://www.nltk.org

Chae, Y., & Davidson, T. (2023). Large language models for text classification: From zero-shot learning to fine-tuning. *Open Science Foundation.*

Devlin, J. (2018). BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805.*

Gasparetto, A., Marcuzzo, M., Zangari, A., & Albarelli, A. (2022). A survey on text classification algorithms: From text to predictions. *Information*, *13*(2), 83.

Gong, Q., Kang, W., & Fahroo, F. (2023). Approximation of compositional functions with ReLU neural networks. *Systems & Control Letters, 175,* 105508.

Guo, Z., Yu, K., Jolfaei, A., Ding, F., & Zhang, N. (2021). Fuz-spam: Label smoothing-based fuzzy detection of spammers in Internet of Things. *IEEE Transactions on Fuzzy Systems, 30*(11), 4543–4554.

Hadi, M. U., et al. (2024). Large language models: A comprehensive survey of its applications, challenges, limitations, and future prospects. *Authorea Preprints.*

Hegazi, M. O., Al-Dossari, Y., Al-Yahy, A., Al-Sumari, A., & Hilal, A. (2021). Preprocessing Arabic text on social media. *Heliyon, 7*(2).

Jamal, S., & Wimmer, H. (2023). An improved transformer-based model for detecting phishing, spam, and ham: A large language model approach. *arXiv preprint arXiv:2311.04913.*

Jana, C., Pal, M., Muhiuddin, G., & Liu, P. (n.d.). Fuzzy optimization, decision-making and operations research.

Kaddoura, S., Alex, S. A., Itani, M., Henno, S., AlNashash, A., & Hemanth, D. J. (2023). Arabic spam tweets classification using deep learning. *Neural Computing and Applications, 35*(23), 17233–17246.

Kamyab, M., Liu, G., & Adjeisah, M. (2021). Attention-based CNN and Bi-LSTM model based on TF-IDF and GloVe word embedding for sentiment analysis. *Applied Sciences, 11*(23), 11255.

Kardaş, B., Bayar, İ. E., Özyer, T., & Alhajj, R. (2021). Detecting spam tweets using machine learning and effective preprocessing. In *Proceedings of the 2021 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining* (pp. 393–398).

Keraghel, I., Morbieu, S., & Nadif, M. (2024). Beyond words: A comparative analysis of LLM embeddings for effective clustering. In *International symposium on intelligent data analysis* (pp. 205–216). Springer.

Kumar, N., & Sonowal, S. (2020). Email spam detection using machine learning algorithms. In *2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA)* (pp. 108–113). IEEE.

Kumar, P. (2024). Large language models (LLMs): Survey, technical frameworks, and future challenges. *Artificial Intelligence Review, 57*(10), 260. https://doi.org/10.1007/s10462-024-10888-y

Patil, R. G. (2024). A review of current trends, techniques, and challenges in large language models (LLMs). *Applied Sciences, 14*(5), 2074.

Reyes-García, C. A. T.-G., & A. A. (2022). Fuzzy logic and fuzzy systems. In *Biosignal processing and classification using computational learning and intelligence: Principles, algorithms, and applications*. Elsevier.

Rojas-Galeano, S. (2024). Zero-shot spam email classification using pre-trained large language models. *arXiv preprint arXiv:2405.15936.*

Rutkowski, L., Cpalka, K., Nowicki, R., Pokropinska, A., & Scherer, R. (2023). Neuro-fuzzy systems. In T.-Y. Lin, C.-J. Liau, & J. Kacprzyk (Eds.), *Granular, fuzzy, and soft computing* (pp. 843–858). Springer.

Sahmoud, T., & Mikki, D. M. (2022). Spam detection using BERT. *arXiv preprint arXiv:2206.02443.*

Soltanifar, M., Sharafi, H., Hosseinzadeh Lotfi, F., Pedrycz, W., & Allahviranloo, T. (2023). Introduction to fuzzy logic. In *Preferential voting and applications: Approaches based on data envelopment analysis* (pp. 31–45). Springer.

Thomas, M., & Meshram, B. (2023). Chso-DNFNet: Spam detection in Twitter using feature fusion and optimized deep neuro-fuzzy network. *Advances in Engineering Software, 175*, 103333.

Xiao, A. S., & Liang, Q. (2024). Spam detection for YouTube video comments using machine learning approaches. *Machine Learning with Applications, 16*, 100550.