A Comparative Analysis on Medical Article Classification Using Text Mining & Machine Learning Algorithms

Burak Kolukisa
Department of Computer Engineering
Abdullah Gul University
Kayseri, Turkey
burak.kolukisa@agu.edu.tr

Bilge Kagan Dedeturk
Research and Development Center
Kayseri Ulaşım A.Ş.
Kayseri, Turkey
kagandedeturk@gmail.com

Beyhan Adanur Dedeturk
Department of Computer Engineering
Abdullah Gul University
Kayseri, Turkey
beyhan.adanur@agu.edu.tr

Abdulkadir Gulsen
Department of Computer Engineering
Abdullah Gul University
Kayseri, Turkey
abdulkadir.gulsen@agu.edu.tr

Gokhan Bakal*
Department of Computer Engineering
Abdullah Gul University
Kayseri, Turkey
gokhan.bakal@agu.edu.tr

2021 6th International Conference on Computer Science and Engineering (UBMK) | 978-1-6654-2908-5/21/$31.00 ©2021 IEEE | DOI: 10.1109/UBMK52708.2021.9559001

***Abstract*—The document classification task is one of the widely studied research fields on multiple domains. The core motivation of the classification task is that the manual classification efforts are impractical due to the exponentially growing document volumes. Thus, we densely need to exploit automated computational approaches, such as machine learning models along with data & text mining techniques. In this study, we concentrated on the classification of medical articles specifically on common cancer types, due to the significance of the field and the decent number of available documents of interest. We deliberately targeted MEDLINE articles about common cancer types because most cancer types share a similar literature composition. Therefore, this situation makes the classification effort relatively more complicated. To this end, we built multiple machine learning models, including both traditional and deep learning architectures. We achieved the best performance (≈82% F score) by the LSTM model. Overall, our results demonstrate a strong effect of exploiting both text mining and machine learning methods to distinguish medical articles on common cancer types.**

***Index Terms*—document classification, text mining, machine learning, deep learning**

I. Introduction

Since the digitalization era began, the size of the digitally available documents has been dramatically increasing in every aspect of our lives, such as books, articles, newspapers, patient records, and social media posts. [1]. The report published by International Data Corporation (IDC) discloses the available data amount will increase more than 40 times between 2012 and 2020 [2]. This outcome not only brought us accessing data resources faster but also the processing challenge of a plethora of digital data. The reason for the challenge is that we cannot manually analyze the exponentially increasing enormous data volumes. Thus, researchers intensively exploit computational techniques to process digitalized data portions [3], [4]. Although text mining is known as a sub-branch of data mining, it generally comprises multiple approaches, including Natural Language Processing (NLP), Information Retrieval, and Machine Learning (ML) algorithms over various domains [5]–[7] for different tasks. From the text mining perspective, one of the widely performed research efforts is document/text classification since identifying the correct document among a document collection is a serious work [8]. Specifically, the separation of highly inter-related documents, such as medical articles, is an even more complicated task. For this reason, in this work, we targeted to classify cancer-related medical articles available in PubMed digital repository into 13 common cancer types† (***Bladder Cancer, Breast Cancer, Colon and Rectal Cancer, Endometrial Cancer, Kidney Cancer, Leukemia, Liver Cancer, Lung Cancer, Melanoma, Non-Hodgkin Lymphoma, Pancreatic Cancer, Prostate Cancer, and Thyroid Cancer***). Briefly, after gathering the corresponding articles, we exploited the abstract sections of the collected articles to conduct our ML-based experiments. In this study, the principal contributions are listed below:

- The multi-label classification power of unigram features is investigated using a Logistic Regression model (as a traditional ML model).
- A simple feed-forward Dense Neural Network model (DNN), basic Recurrent Neural Network (RNN), Convolutional Neural Network (CNN), and Long-Short Term Memory (LSTM) models are built to compare the classification performances.
- The articles about 13 common cancer types are chosen on purpose to work on a more complicated classification

*Corresponding author.

†https://www.cancer.gov/types/common-cancers

978-1-6654-2908-5/21/$31.00 ©2021 IEEE

task due to the inter-related literature in the documents.

The rest of the paper is organized as follows. In Section II, background knowledge on document classification is discussed, and related studies are briefly summarized. In Section III, the dataset curating approach and statistical details about the dataset as well as the preprocessing steps are explained. Section IV elaborately describes the methodologies used in classification and the models' configurations. Then, Section V shows the performance results of the models comparatively and discusses the outcomes. Ultimately, Section VI summarizes the effort and gives some future directions.

II. Background & Related Efforts

As a formal definition, the document classification task is a process to assign correct single or multiple labels. Borko and Bernick [9] proved that automatic document classification is possible using a factor analysis technique. After this milestone work, computational document classification gained more popularity among academic society. Later, Liang [10] demonstrated employing the Support Vector Machine (SVM) model (with a one-vs-all infrastructure) is an ideal solution for web page classification. Researchers also examined the strength of text/document classification over social media posts to predict the psychological situations of the users [11]. Behere et al. examined the performance of the popular state-of-the-art deep learning-based classification architectures for the automatic classification task on multiple biomedical datasets. In consequence, they showed that deep learning based models outperform traditional ML algorithms [12]. Plus, Du et al. conducted multi-label classification experiments on a biomedical dataset to show the superior classification performance of the deep neural network architecture [13]. Yet another multi-label document classification effort performed by Sovrano et al. showed that deep learning models supported by domain-specific Term Frequency - Inverse Document Frequency (TF-IDF) similarities yield reasonably good performance scores [14].

In this research effort, the fundamental motivation is to investigate the classification power of both extracted textual features and distinct machine learning algorithms, including Logistic Regression (LR) and Deep Learning (DL) models comparatively. To perform that more plausibly, we aimed to classify the medical articles about common cancer types, which are relatively hard-to-classify documents due to having a highly similar context.

III. Dataset Curation & Text Preprocessing

As we mentioned in Section I, we collected (by searching common cancer names) medical articles publicly available in PubMed digital repository to derive our dataset. To collect subsets of documents corresponding to distinct cancer types, we assumed the collected articles belonging to the cancer types (by searching them as keywords). It can be the main caveat of this study, such that we did not ensure that each document is presented in a single class. Nevertheless, this study can be seen as a more challenging effort due to the assumption we followed. This is because the classification becomes much more complicated. Based on the search responses, the number of articles is shown in Table I. Among each article/document collection, we decided to randomly pick 10K examples to form our final dataset consisting of 130K articles. The main reason why we opted for only 10K examples from each cancer subset is to have an efficient balanced scenario.

TABLE I
DATASET STATS

Cancer Type	Number of Articles
Bladder Cancer	85,218
Breast Cancer	435,545
Colon and Rectal Cancer	26,289
Endometrial Cancer	40,363
Kidney Cancer	132,146
Leukemia‡	342,019
Liver Cancer	282,637
Lung Cancer	370,355
Melanoma	143,095
Non-Hodgkin Lymphoma‡	118,792
Pancreatic Cancer	115,486
Prostate Cancer	115,486
Thyroid Cancer	84,033

After forming the dataset, we applied specific preprocessing steps. First, we tokenized (separating a piece of text into smaller units called tokens) and then lemmatized (converting each token/word into its base/root mode) each word in the dataset. To perform these steps, we utilized Natural Language Toolkit (NLTK) python package [15]. To clean the data, the punctuation marks, stopwords (using NLTK's stopwords list) and numeric terms were removed from the sentences. Also, the examples, which do not have any textual representation, were omitted from the dataset. This unexpected case happens when collected articles do not have any information (literally nothing) in the abstract section. After removing the examples having empty content, the size of the dataset dropped to 114,965. Besides, we need to convert the tokenized textual elements/words into numerical representations to give them to the ML models as inputs. This process is called feature extraction (also known as *vectorization*). Here, first, the *CountVectorizer* method, available under the Scikit-learn [16] ML package, is used to convert our documents into a vector of term/token counts. Then, we applied the uni-gram model to form our feature space. Ultimately, we obtained 169,355 distinct terms and 114,965 x 169,355 document-term matrix. Using only the frequency counts may not yield the significant characteristics because a feature element (either from BoWs or n-grams) with a relatively lower frequency may be more informative. Hence, we need to utilize a more robust representation approach for the feature encoding. Here, one practical way of solving this issue is to employing TF-IDF weighting [17]–[19] for the feature elements. The leading reason is that the TF-IDF weighting method assesses how each feature item is relevant to a document in the dataset. As a practical explanation, a high TF-IDF score of a term means having a higher term frequency

‡Although Leukemia and Lymphoma are particular sub-types of blood cancer, we do not merge them since they are recognized as distinct common cancer diseases.

in the corresponding document and a relatively lower document frequency among the collection of all documents. Thus, we computed the TF-IDF weights to handle the issue caused by using raw frequency counts. To run and test our models, we randomly split the dataset into **training (70%), testing (15%), and validation (15%) corpora**. Also, we repeated this split process ten times to have multiple unique datasets and to derive the average performance metrics.

IV. Methods

In this section, we first briefly mention the technical definition of an ML model. Following that, we introduce the logistic regression model as a traditional approach with the specific configurations we imposed. Afterward, we describe a feed-forward dense neural network model along with distinct deep learning models, including RNN, CNN as well as LSTM architectures. To illustrate the methodology, we presented a diagram comprising flow steps in Figure 1.

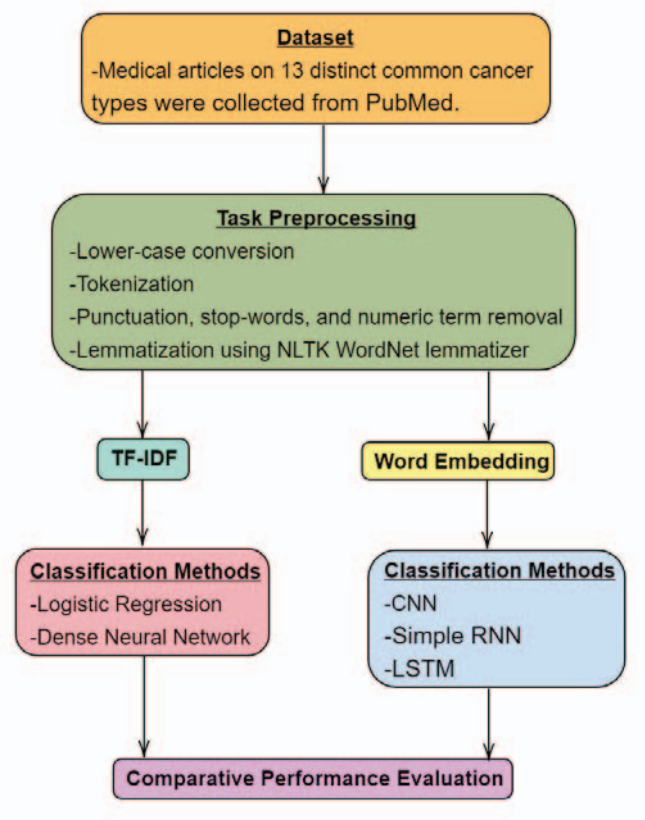

Fig. 1. Overall methodology diagram

A. Machine Learning Models

To concisely elucidate a supervised machine learning model, we can technically say that it is the model learning the contextual characteristics of the training examples to assign the correct label(s) to test examples.

B. Logistic Regression Models

Logistic regression (LR) is a well-known and widely used method for binary classification tasks. Technically, LR is a statistical machine learning algorithm that tries to identify a logarithmic line separating the differences of the samples at the best point [20]. Although in one way LR resembles linear regression, the LR algorithm creates a more complex decision boundary using the logistic function (also known as the sigmoid function) for the separation task. Hence, it yields better performance results on contextually interrelated datasets such as cancer types classification. In the LR models, firstly, TF-IDF weighting is applied to the dataset. The input vector size is limited to 200 because the increase in feature size did not provide any considerable gains for the performance results. Secondly, the *GridSearchCV* method is applied to the LR model to identify the best parameter combinations for achieving better performance results. These model parameters are penalty type (*penalty* $\in$ {"l2", "none"}), solver argument (*solver* $\in$ {"newton-cg", "lbfgs", "sag", and "saga"}), and multi-class option (*multi-class* $\in$ "auto", "ovr", "multinomial"). To conduct the LR model experiment, first, the model is trained, and the validation dataset is subjected to the trained model to obtain the best-performing model with the optimum parameters. Following that, the model is tested on the test dataset to evaluate the model.

C. Neural Networks & Deep Learning Architectures

In this section, we will introduce the neural network models used in the experiments. These models are individually explained in detail in the following subsections.

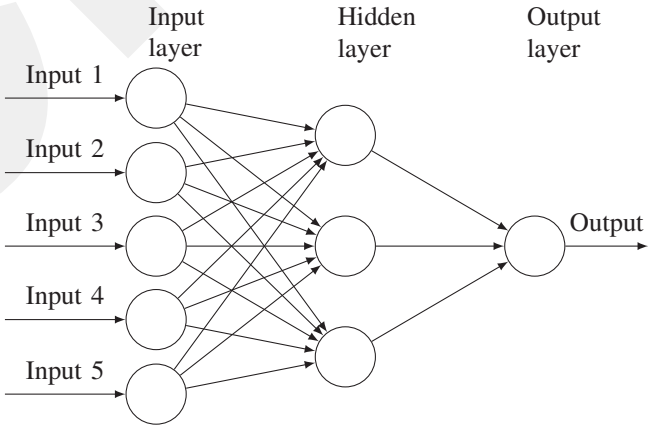

Fig. 2. An example of a basic feed-forward dense neural network

1) Dense Neural Network (DNN): In general, neural network-based (NN) algorithms are the methods that simulate biological neural systems to process input data as a learning procedure. Specifically, a dense neural network (DNN)

is a typical feed-forward NN model having densely interconnected artificial neurons (in the intermediate layer(s)) with the input and output layers as shown in Figure 2. Although a DNN model requires more training time due to the complexity, it yields better performance compared to LR models for multi-class classification studies [21].

$$\mathcal{L} = -\sum_{i=1}^{t} y_i \cdot log\, \hat{y}_i \quad (1)$$

In the DNN model, firstly, TF-IDF weights of the feature space are computed over the dataset, and the input vector size is limited to 200 simply due to the same situation that we expressed for the LR model. Secondly, a simple DNN architecture is constructed with four consecutive layers containing 256, 64, 32, and 13 neurons respectively. This structure is because the dataset has extensive correlated information pieces. In the first three layers Rectified Linear Unit (*ReLU*) activation function is applied while the last layer (as output) has a softmax activation function to generate final multi-class probabilities. During the learning/training process, the *Adam optimizer* is run with the `categorical cross-entropy loss` function as formulated in Equation 1 (where t is the number of classes, while y_i is the corresponding label value and $\hat{y}_i$ is the output value). Plus, the selected batch size is 8, and the number of epoch used is limited to 20 with the early-stopping mechanism for a better generalization. So, if the validation accuracy does not improve after five epochs, the model training is terminated, and the best weights are restored. To obtain the model's final performance, similar steps are followed as in the LR model. The most accurate model is specified by using the validation dataset, then the test dataset is subjected to the identified model.

2) Recurrent Neural Network Model: A recurrent neural network (RNN) model, also known as a Vanilla RNN, is a type of deep neural network that is especially more suitable and powerful for sequential/temporal data, such as time series and natural language processing analyses [22]. This ability is because the model does not solely consider the principal input data but also previously processed data through the memorizing capability. Technically, an RNN model utilizes the output of a particular hidden layer and yields it as input back to the network. In this study, we built a vanilla RNN model to examine its multinomial classification strength. To this end, we exploited the word embedding approach [23] with a word vector size of 80 and the sequence length (e.g., a sentence length) of 200, while the hidden state dimension is selected as 250. Plus, the *GlobalMaxPooling1D* operation is applied to transform/reduce the global input representation by taking the maximum value over each temporal input phase. The number of epochs is 20 for the training period. Additionally, a similar early-stopping approach (which is mentioned in the DNN model section) is applied to prevent over-fitting issues.

3) Long Short-Term Memory (LSTM) Model: LSTM is a specific RNN algorithm which was developed to solve vanished gradient descent or bursting problem [24]. The fundamental difference is that the LSTM cell has exclusive gates. As illustrated in Figure 3, the input gate quantifies how important the information is, yielded by the input. The forget gate decides which information will be erased (the irrelevant ones) from the cell state, while the output gate denotes the information to feed-forward to the next hidden state. Considering these characteristics, the LSTM architecture can easily manage the long-term dependency among the input data elements to relax the vanishing gradient problem to some extend, unlike vanilla RNN models.

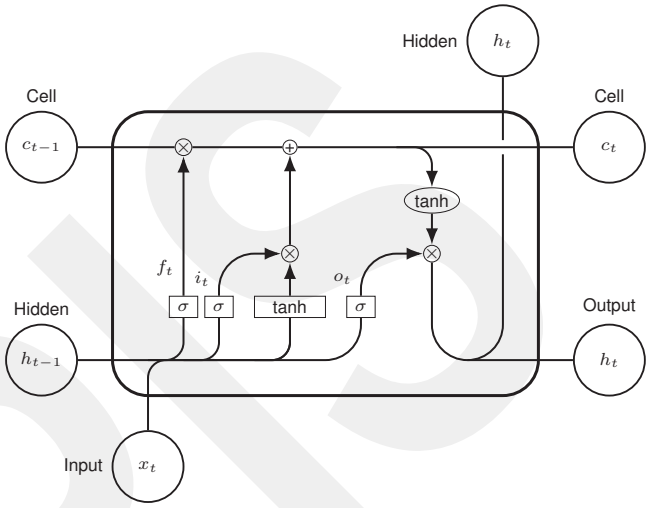

Fig. 3. An LSTM cell structure

In the LSTM model experiments, the word embedding approach is administered, and the embedding vector size is selected as 80, while the sequence length is specified as 200. Similar to the vanilla RNN model, the hidden state dimension is determined as 250. Besides, the *GlobalMaxPooling1D* operation is also performed, and the output layer is built by the softmax function with 13 neurons. Also, the early-stopping procedure is configured based on the validation accuracy with five controlling steps over 20 epochs.

4) Convolutional Neural Network (CNN) Model: A typical CNN architecture is a feed-forward deep neural network comprising an input layer, various hidden layers, and an output layer. Technically, the hidden layers consist of convolution layers, an activation layer, pooling, and fully connected dense layers. The convolution layer is the central part of the CNN model in which extracts discriminative/special features happening in the input [25]. For a more concrete example, the convolution operator identifies meaningful patterns/associations between textual data with predefined convolution filters. The activation layer provides the non-linearity to the model with an activation function (e.g., *ReLU*, *tanh*, and *sigmoid*), which helps to solve the vanishing gradient problem disrupting the weights update during the training. The principal purpose of a pooling layer is to reduce the size of the data representation, the number of

parameters, and the computation cost in the model.

In the CNN model, we also applied the word embedding technique with the embedding vector size of 80 and the sequence length of 200. To construct the CNN architecture, first, the input data is subjected to the embedding layer. Then, the first convolutional layer is employed with a filter size of 128, a kernel size of 3, and the *ReLU* activation function. After the convolution part, a max-pooling operation is performed with a pool size window of 3. In the next two steps, similar convolution and max-pooling layers with the same configuration parameters are applied. Afterward, a dense layer having 128 units and the *ReLU* activation function is employed before the output layer containing 13 units/neurons with the sigmoid activation function.

V. Results & Discussion

In this section, we demonstrate the experimental performance metrics for each model. The performance scores are derived from 10 distinctly split datasets and presented in Table II. Although the LR model yielded the lowest performance metrics among distinct neural network models, it indeed achieved nearly 4% less F1 score compared to the best model. This situation is because linear models with the unigram feature set can capture highly discriminative features. Unlike the LR model, the DNN, the basic nonlinear model, improved the performance scores by around 2% for each metric.

TABLE II
Performance scores for all models

	Precision	**Recall**	**F1-Score**††
LR	78.00	77.54	77.59
DNN	79.88	79.79	79.84
CNN	81.32	80.57	80.94
RNN	81.73	81.38	81.55
LSTM	**81.91**	**81.91**	**81.91**

As expected, we achieved a better classification performance (nearly 2%) with the DNN setup compared to the LR model. However, surprisingly the CNN configuration slightly outperformed the DNN model. This case is probably because the convolution operation barely captures neighborhood associations of the words in the sentences. Predictably, the CNN model obtained a 3% improvement on the F1 measure over the traditional LR model. Performances, when we use the vanilla RNN and LSTM models, are superior among all models as indicated in the table. Nevertheless, the LSTM model (only 0.36% better) yielded a slightly higher F1 score than the vanilla RNN model. This consequence is not surprising since the LSTM architecture is the most robust model when compared with other models for the text classification task.

When comparing the performance scores, it is clear that we obtained a regular performance gain, although the improvement level is not as high when we employed more advanced models. Furthermore, another substantial outcome is that the neural network-based models have a lower difference between precision and recall in comparison with the LR model. For instance, we obtained similar precision and recall scores when we administered the LSTM model. This observation shows that the LSTM algorithm is more powerful for classifying relevant examples by reducing the false-negative counts.

†† We computed the final average F1-score by using average precision and average recall scores.

VI. Conclusion

In this effort, we aimed to classify a medical corpus which is a hard-to-classify dataset due to the intensively interrelated context among the instances. First, we collected the data examples to create our specific dataset comprising abstract sections of the medical articles about 13 common cancer types from PubMed medical article repository. Then, we applied several preprocessing steps (such as cleaning, tokenization, unigram modeling, TF-IDF scores generation, and so on) to the dataset before constructing the ML models. Following the preprocessing phase, we built distinct ML models, including LR, DNN, CNN, RNN, and LSTM. Afterward, we performed our experiments by running the ML models. Overall, we obtained nearly F1 score of **82%** with the best LSTM model configuration. The comparative results indicate that the deep learning architectures outperformed the traditional ML model, the logistic regression model. Thus, this consequence proved that DL models have high potential on the classification task for a challenging dataset.

Beyond the existing effort, we would like to list the potential future works as stated below,

- Although the LR model yielded the lowest classification performance, we would like to apply a broader feature space with a larger threshold value. This idea is because we believe that the LR model can improve performance if it has more informative features.
- Beyond the unigram model, we can extend the n-gram model features where $n \in \{2, 3, 4, 5\}$ in the models (using their individual and compound sets).
- In this study, we created the dataset such that each cancer type has a similar amount of documents for having a balanced scenario. However, we also want to examine the classification performance for the imbalanced dataset scenario where each cancer class has either the actual amount or the corresponding ratio amount of documents over existing articles.
- Another potential future direction is that we can build even more advanced state-of-the-art deep learning architectures, such as the Bidirectional Encoder Representations from Transformers (BERT) model and the BioBERT model, which is exclusively designed for biomedical domains.

Authors' Contributions

G.B. conceived of the research idea, and wrote the paper with input from all authors. A.G. and B.K. collected the data and created the dataset. B.A.D. analyzed the dataset. B.A.D. and B.K.D. generated word embeddings and TF-IDF weights. A.G., B.K., and B.K.D. designed the LR model. B.K.D. designed CNN and LSTM models, while B.K. developed

RNN and DNN models. B.K. and B.K.D. conducted the experimental models.

REFERENCES

[1] G. Bakal and R. Kavuluru, "On quantifying diffusion of health information on twitter," in *2017 IEEE EMBS International conference on biomedical & health informatics (BHI)*. IEEE, 2017, pp. 485–488.

[2] J. Gantz and D. Reinsel, "The digital universe in 2020: Big data, bigger digital shadows, and biggest growth in the far east," *IDC iView: IDC Analyze the future*, vol. 2007, no. 2012, pp. 1–16, 2012.

[3] A. Labrinidis and H. V. Jagadish, "Challenges and opportunities with big data," *Proceedings of the VLDB Endowment*, vol. 5, no. 12, pp. 2032–2033, 2012.

[4] G. Bakal, P. Talari, E. V. Kakani, and R. Kavuluru, "Exploiting semantic patterns over biomedical knowledge graphs for predicting treatment and causative relations," *Journal of biomedical informatics*, vol. 82, pp. 189–199, 2018.

[5] C. Manning and H. Schutze, *Foundations of statistical natural language processing*. MIT press, 1999.

[6] R. Grishman, "Information extraction," *IEEE Intelligent Systems*, vol. 30, no. 5, pp. 8–15, 2015.

[7] G. Bakal, H. Abar, I. Ozturk, and O. Abar, *Text Mining Applications Using Real-World Data in Python*. Nobel Press, 2021.

[8] T. M. Mitchell, *Machine Learning*. New York: McGraw-Hill, 1997.

[9] H. Borko and M. Bernick, "Automatic document classification," *Journal of the ACM (JACM)*, vol. 10, no. 2, pp. 151–162, 1963.

[10] J.-Z. Liang, "Svm multi-classifier and web document classification," in *Proceedings of 2004 International Conference on Machine Learning and Cybernetics (IEEE Cat. No. 04EX826)*, vol. 3. IEEE, 2004, pp. 1347–1351.

[11] S. G. Burdisso, M. Errecalde, and M. Montes-y Gómez, "A text classification framework for simple and effective early depression detection over social media streams," *Expert Systems with Applications*, vol. 133, pp. 182–197, 2019.

[12] B. Behera, G. Kumaravelan, and P. Kumar, "Performance evaluation of deep learning algorithms in biomedical document classification," in *2019 11th International Conference on Advanced Computing (ICoAC)*. IEEE, 2019, pp. 220–224.

[13] J. Du, Q. Chen, Y. Peng, Y. Xiang, C. Tao, and Z. Lu, "ML-Net: multi-label classification of biomedical texts with deep neural networks," *Journal of the American Medical Informatics Association*, vol. 26, no. 11, pp. 1279–1285, 06 2019. [Online]. Available: https://doi.org/10.1093/jamia/ocz085

[14] F. Sovrano, M. Palmirani, and F. Vitali, "Deep learning based multi-label text classification of unga resolutions," in *Proceedings of the 13th International Conference on Theory and Practice of Electronic Governance*, ser. ICEGOV 2020. Association for Computing Machinery, 2020, p. 686–695.

[15] S. Bird, E. Klein, and E. Loper, *Natural Language Processing with Python*, 1st ed. O'Reilly Media, Inc., 2009.

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

[17] D. Jurafsky and J. H. Martin, *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*, 1st ed. USA: Prentice Hall PTR, 2000.

[18] B. K. Dedeturk and B. Akay, "Spam filtering using a logistic regression model trained by an artificial bee colony algorithm," *Applied Soft Computing*, vol. 91, p. 106229, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1568494620301691

[19] B. K. Dedeturk, B. Akay, and D. Karaboga, *Artificial Bee Colony Algorithm and Its Application to Content Filtering in Digital Communication*. Singapore: Springer Singapore, 2021, pp. 337–355.

[20] D. G. Kleinbaum and M. Klein, *Logistic Regression: A Self-Learning Text*, ser. Statistics for Biology and Health. Springer New York, 2010.

[21] S. Jain, A. K. Jain, and S. P. Singh, "Building a machine learning model for unstructured text classification: Towards hybrid approach," in *Rising Threats in Expert Applications and Solutions*, V. S. Rathore, N. Dey, V. Piuri, R. Babo, Z. Polkowski, and J. M. R. S. Tavares, Eds. Singapore: Springer Singapore, 2021, pp. 447–454.

[22] J. Farkas, "Document classification and recurrent neural networks," in *Proceedings of the 1995 conference of the Centre for Advanced Studies on Collaborative research*, 1995, p. 21.

[23] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in Neural Information Processing Systems*, C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, Eds., vol. 26. Curran Associates, Inc., 2013.

[24] K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink, and J. Schmidhuber, "Lstm: A search space odyssey," *IEEE transactions on neural networks and learning systems*, vol. 28, no. 10, pp. 2222–2232, 2016.

[25] M. Z. Afzal, S. Capobianco, M. I. Malik, S. Marinai, T. M. Breuel, A. Dengel, and M. Liwicki, "Deepdocclassifier: Document classification with deep convolutional neural network," in *2015 13th international conference on document analysis and recognition (ICDAR)*. IEEE, 2015, pp. 1111–1115.