Words Speak Louder Than Actions: Decoding Emotions Through NLP

Melda Paksoy
Department of Computer Engineering
Abdullah Gul University
Kayseri, Türkiye
melda.paksoy@agu.edu.tr

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

2024 9th International Conference on Computer Science and Engineering (UBMK) | 979-8-3503-6588-7/24/$31.00 ©2024 IEEE | DOI: 10.1109/UBMK63289.2024.10773493

***Abstract*—Emotion detection in text remains a significant challenge in Natural Language Processing due to human emotions' complexity and subtle nuances. This paper presents multiple experimental models for emotion classification using an up-to-date dataset curated to address 13 emotions implied in Twitter posts. We evaluated various machine learning (ML) models, including Logistic Regression, Random Forest, SVM, and XGBoost, alongside deep learning (DL) architectures such as LSTM and CNN. Our results demonstrate the efficacy of deep learning models, particularly the CNN model by achieving an impressive F1 score of 0.99. This study contributes to emotion detection capabilities, paving the way for more nuanced and accurate sentiment analysis (SA) in various text analysis applications.**

Keywords*—*emotion detection, text mining, machine learning, deep learning

I. Introduction

The exponentially growing digital communication has left us inundated with textual data, which encompasses a vast spectrum of human emotions and opinions. Interpreting these emotions is no longer solely the domain of human intuition; in parallel, SA and emotion classification have emerged as powerful tools to automatically extract and understand these subjective states from text. While SA lays the foundation by identifying the overall polarity of opinions (positive, negative, or neutral) [1, 2], emotion classification takes a deeper dive to address specific emotions, such as joy, sadness, anger, fear, and a range of others [3]. This granular understanding of emotions unlocks a richer comprehension of the intentions, motivations, and reactions embedded within the text, with far-reaching implications across diverse fields.

Through this in-depth analysis, we aimed to provide a holistic understanding of the traditional ML and well-known DL models in sentiment and emotion analysis. We will explore the strengths and limitations of existing approaches, identify critical areas for future research and development, and shed light on the transformative potential of these technologies across several applications, from enhancing customer service experiences and gauging public opinion to improving mental health support and enriching human-computer interaction. By delving into the intricacies of sentiment and emotion analysis, we embark on a journey toward a more nuanced and insightful comprehension of human communication in the digital age.

Specifically, the essential contribution of this study is to build multiple ML and DL models to classify text instances into correct emotion categories among unique 13 emotions.

Following this introduction, Section II provides foundational knowledge on sentiment and emotion analysis. Section III details the dataset utilized in the experiments, while Section IV delves into the model-building process, exploring various configurations. Section V presents the experimental results and comparative evaluations, culminating in Section VI with a concluding summary of the research efforts and findings.

II. Background & Related Work

Text analysis is a widely explored research area across various domains, encompassing tasks such as information retrieval, text classification, knowledge prediction, and machine translation [4, 5, 6, 7]. Within this broad field, text classification has gained prominence due to its applicability to numerous problems, including SA, spam detection, and document classification [2, 8, 9]. SA, in particular, has emerged as a fundamental area of study within text classification, attracting significant attention from researchers for its diverse applications. For example, SA has been utilized in product review analysis to gauge consumer opinions, in personal condition identification to monitor mental health states, and in fake news detection to discern the credibility of information [10, 11]. These varied applications highlight the versatility and importance of sentiment analysis in addressing unique challenges across different disciplines.

However, as a more challenging and derived task from sentiment analysis, emotion detection in text is even more crucial area of research in Natural Language Processing (NLP), with wide-ranging applications across various domains, including social media analysis, customer sentiment understanding, and mental health monitoring [12]. The ability to accurately emotion identification expressed in text provides valuable insights into user behavior, opinions, and overall sentiment, enabling better decision-making and personalized interactions. The complexity of human emotions poses significant challenges to automated emotion detection. Emotions are often expressed subtly through linguistic cues, including choice of words, syntax, and even emojis. Moreover, the same words can also convey different emotions depending on context due to the language complexity; thus, this situation makes accurate emotion classification a more challenging task.

The field of emotion research has witnessed substantial growth in recent decades, leading to the development of various influential emotion models. This paper revisits some of these foundational models, highlighting their principal contributions and limitations. Early efforts, such as the work by Shaver et al. [13], focused on grouping emotions into

979-8-3503-6588-7/24/$31.00 ©2024 IEEE

prototypes based on semantic similarity. Through a rigorous process involving word selection, emotionality rating, and similarity annotation, they proposed a hierarchical model culminating in six basic emotions: joy, love, surprise, sadness, anger, and fear. This model suggests that most emotions are nuanced combinations of these fundamental six, a perspective challenged by subsequent research. Parrott [14] elaborated on this hierarchical structure, presenting a detailed tree-structured list of emotions, building upon Shaver et al.'s work. However, Ortony and Turner [15] critiqued the notion of "basic emotions" as psychologically primitive building blocks. They posited that emotions are discrete, independent entities interlinked through a hierarchical framework with no fundamental set serving as building blocks for others. Rejecting the premise of basic emotions, Ortony, Clore, and Collins [16] introduced the OCC model, named after its creators. This model categorizes emotions into 22 distinct types, grouping them based on arousal levels within a hierarchical structure. The model branches into three primary categories: reactions to events (e.g., pleased, displeased), judgments of agents (e.g., approving, disapproving), and appraisals of objects (e.g., liking, disliking). Recent advancements in deep learning have shown superior performance against traditional machine learning models, such as the Random Forest Classifier, i n e motion c lassification. De ep le arning models automatically learn hierarchical feature representations from raw data, which enables them to effectively capture subtle complexities encoded in emotional expressions. This superior performance is attributed to their capacity to model intricate non-linear relationships within data and effectively utilize large datasets, leading to improved accuracy in emotion recognition tasks [17].

III. Dataset Details

This research employed the "Emotion analysis based on text" dataset, readily accessible on the Kaggle platform [18]. This dataset comprises 840K instances, with their sentiment classes detailed in Table I. Each instance in the dataset belongs exclusively to a single class out of the 13 distinct emotion classes, ensuring that there are no multi-class instances. Furthermore, as previously discussed, traditional machine learning models and deep learning architectures were utilized to assess their efficacy in the sentiment analysis task using the aforementioned dataset.

TABLE I. Class distribution of dataset

Class Name	Instance Amount	Ratio
Neutral	674,538	80.3447%
Love	39,553	4.7112%
Happiness	27,175	3.2368%
Sadness	17,481	2.0822%
Relief	16,729	1.9926%
Hate	15,267	1.8185%
Anger	12,336	1.4893%
Fun	10,075	1.2%
Enthusiasm	9,304	1.1082%
Surprise	6,954	0.8283%
Empty	5,542	0.6601%
Worry	4,475	0.533%
Boredom	126	0.015%

The primary rationale for employing this dataset lies in its inclusion of 13 distinct emotional categories, rendering it more complex than conventional sentiment analysis studies. This complexity arises from contextually similar classes, such as happiness and fun, which necessitate a more nuanced understanding of emotional expression.

IV. Methodology

In this section, we describe the pre-processing steps applied to the dataset, experimental configuration information, and the models we built with their technical details.

A. Text Preprocessing

In this study, we implemented several pre-processing steps using the publicly available, open-source Natural Language Toolkit (NLTK) to ensure consistency. First, we converted all letters to lowercase, thereby standardizing the vocabulary, reducing complexity and diversity, and preventing the duplication of information in the model's vocabulary. This convention ensured consistent processing of words regardless of their case. Second, we tokenized the texts to decompose the data into standardized units. Third, we replaced multiple spaces with a single space and removed stop words. Finally, we lemmatized the data using the WordNet Lemmatizer by incorporating part-of-speech tagging to determine the grammatical contexts of word-token pairs.

B. Experimental Configuration Details

To evaluate the performance of the selected models across different dataset sizes, we divided the data into subsets containing 30,000, 15,000, and 5,000 instances by ensuring that the class ratios were preserved during subsampling. Additionally, we created balanced subsets by selecting an equal number of examples for each class, except for the boredom class, which had only 126 entries. Consequently, we generated three subsets with 4,998 (406x12+126), 14,994 (1239x12+126), and 29,994 (2489x12+126) instances, respectively. This setup resulted in six subclasses for each selected model.

For traditional models, we applied the Term Frequency-Inverse Document Frequency (TF-IDF) Vectorizer to convert the text data into numerical representations operating five distinct n-gram configurations (unigram, bigram, trigram, unigram-bigram, and unigram-bigram-trigram). Consequently, we created 30 different essential model configurations. Additionally, we randomly shuffled the dataset 10 times and split the shuffled dataset into 70% training, 20% testing, and 10% validation.

C. Machine Learning Models

Considering the commonly used powerful machine learning (ML) algorithms, we conducted several experiments by constructing unique ML models. These models are presented in the following sublist.

1) **Logistic Regression**: Logistic Regression (LR) is a widely employed ML model for classification tasks [19]. It operates by calculating the natural logarithm of probabilities and subsequently transforming these values into discrete outcomes ranging from 0 to 1 [20]. This model is applicable to both binary and multi-class classification problems [21]. In this study, logistic regression was utilized for multi-class classification using the Scikit-learn library, an open-source ML framework for the Python programming language [22].

2) **Random Forest**: Random Forest (RF) is a conventional ensemble model composed of numerous decision trees operating independently. It integrates the predictions of these individual trees, ultimately returning the values with the highest scores [23]. For this study, Random Forest was implemented using the Scikit-learn library, specifying the construction of 100 decision trees.
3) **Support Vector Machine**: Support Vector Machine (SVM) stands as a widely utilized model for supervised learning, despite its origins in unsupervised learning. It operates by identifying the optimal hyperplane that separates data points into distinct classes. This is achieved by maximizing the margin between the hyperplane and the closest data points from each class [24]. In this study, the SVM implementation from Scikit-learn was employed. Furthermore, both test and validation data were scaled using MaxAbsScaler to maintain sparsity and StandardScaler to mitigate data leakage.
4) **XGBoost**: XGBoost is a scalable, end-to-end tree boosting system widely employed and recognized for its effectiveness in classification tasks [25]. This study utilized the XGBoost implementation from Scikit-learn, incorporating a label encoder to transform categorical class labels into numerical values ranging from 0 to 12.

D. Deep Learning Architectures

In addition to traditional ML models, this study explored the efficacy of deep learning architectures for sentiment analysis. Specifically, we focused on two prominent neural network architectures known for their ability to capture intricate patterns in sequential data: Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTMs). These architectures are detailed in the following subsections.

1) **Convolutional Neural Networks**: Convolutional Neural Networks (CNNs) are a class of deep learning architectures renowned for their capacity to discern intricate patterns within data. CNNs typically consist of convolutional layers (which apply learnable filters to the input data), nonlinear activation layers (such as ReLU), and pooling layers (responsible for reducing the spatial dimensions of the data). This hierarchical structure allows CNNs to progressively identify higher-level features, detecting simple patterns at lower layers and composing them into more complex representations at higher layers. This property has made CNNs highly effective for both image and text classification tasks [26]. In this study, given the textual nature of the data, instances were represented as one-dimensional arrays, and consequently, one-dimensional convolutional and pooling layers were employed. The CNN model was constructed using Keras, an open-source neural network library in Python [27]. The network architecture incorporated an embedding layer to transform input integer indices into dense vectors of size 100. Subsequently, a convolutional layer with 128 filters and a kernel size of 5 was applied, utilizing the ReLU activation function. A global max pooling layer was employed to extract the most salient features from each feature map. Finally, a dense layer with a Softmax activation function and 13 units was used for the classification task.
2) **Long Short-Term Memory**: Long Short-Term Memory (LSTM) networks are a specialized form of recurrent neural networks designed to effectively capture long-range dependencies within sequential data. LSTMs achieve this through a sophisticated gating mechanism, enabling them to selectively retain and update information over extended time steps. These gates, namely the forget gate, input gate, and output gate, govern the flow of information through the network:
   - *Forget Gate:* Determines which information from the previous cell state should be discarded.
   - *Input Gate:* Regulates the integration of new information into the cell state.
   - *Output Gate:* Controls the information extracted from the cell state to form the final output [28].

   In this study, an LSTM network was implemented using the Keras library [27]. The model architecture included an embedding layer to map integer-represented words into dense vectors of a fixed size (50 dimensions), with a maximum input sequence length of 100. This was followed by an LSTM layer with 64 units. Finally, a dense layer with a unit count equal to the number of classes and a Softmax activation function was used for classification. The LSTM model was compiled with the Adam optimizer, an adaptive learning rate optimization algorithm known for its efficiency in neural network training [29, 30]. Training was conducted for 6 epochs.

V. Results & Discussion

As detailed in Section IV, the performance of both traditional ML models and deep learning architectures was evaluated across a range of experimental configurations, varying the n-gram values, dataset sizes, and subsampling techniques. The average F1 scores obtained for each configuration are presented in Table II (ML models) and Table III (DL models). The F1 score was selected as the primary evaluation metric for this multi-class sentiment classification task due to its balanced assessment of classifier performance. By considering both precision and recall, the F1 score effectively captures the trade-off between these two aspects, providing a more comprehensive evaluation than accuracy or individual precision and recall values alone. This aligns with the emphasis in sentiment analysis research on achieving robust and balanced classification performance across all sentiment categories. Analysis of the experimental results reveals noteworthy trends. Among all configurations, the CNN model trained on imbalanced subsets containing 30,000 instances attained the highest average F1 score of 99.50%. Conversely, the lowest average F1 score of 30.50% was observed with the Logistic Regression model utilizing trigrams and a balanced subset of 4,998 instances. Within the traditional machine learning models, the XGBoost model achieved the highest average F1 score of 97.86% when trained on unigram representations using balanced subsets of 30,000 instances. In terms of the deep learning architectures, the lowest average F1 score of 85.58% was obtained by the CNN model utilizing balanced subsets with 5,000 instances. While the performance difference between the CNN and LSTM architectures was not substantial overall, CNNs generally exhibited higher average F1 scores. Notably, the average F1 scores of the two architectures converged as the dataset size

TABLE II. AVERAGE F1 SCORES OF THE MACHINE LEARNING MODELS ACROSS VARIOUS DATA SIZES AND N-GRAM CONFIGURATIONS

Scenario	N-Gram	Data Size	Logistic Regression	Random Forest	SVM	XGBoost
Imbalanced	Unigram	5,000	0.82546	0.89436	0.83375	0.83375
		15,000	0.88449	0.954386	0.96179	0.97714
		30,000	0.92003	0.96074	0.97852	**0.97862**
	Bigram	5,000	0.71947	0.83332	0.81481	0.75827
		15,000	0.80451	**0.85898**	0.83499	0.82744
		30,000	0.82416	0.88074	0.80822	0.85037
	Trigram	5,000	0.71258	0.74606	0.75240	0.71350
		15,000	0.71592	0.81508	0.81100	0.71360
		30,000	0.71468	**0.83367**	0.83278	0.73931
	Unigram-Bigram	5,000	0.78411	0.87921	0.85516	0.92089
		15,000	0.84042	0.91144	0.90055	0.97031
		30,000	0.86885	0.92846	0.93304	**0.97861**
	Unigram-Bigram-Trigram	5,000	0.74991	0.85200	0.82786	0.91956
		15,000	0.82162	0.88984	0.86528	0.96795
		30,000	0.84838	0.91261	0.89347	**0.97556**
Balanced	Unigram	[(406x12)+126]= 4,998	0.91467	0.92459	0.88062	0.90062
		[(1239x12)+126]= 14,994	0.94518	0.94851	0.95415	0.94971
		[(2489x12)+126]= 29,994	0.95982	0.96419	**0.97195**	0.96901
	Bigram	[(406x12)+126]= 4,998	0.52959	0.52177	0.54990	0.33067
		[(1239x12)+126]= 14,994	0.62852	0.62158	0.65522	0.45885
		[(2489x12)+126]= 29,994	0.71675	0.68342	**0.74087**	0.53574
	Trigram	[(406x12)+126]= 4,998	0.30501	0.30429	0.32602	0.40393
		[(1239x12)+126]= 14,994	0.45035	0.44257	0.46551	0.32657
		[(2489x12)+126]= 29,994	0.57396	0.57085	0.58878	**0.68444**
	Unigram-Bigram	[(406x12)+126]= 4,998	0.89248	0.91232	0.85922	0.89841
		[(1239x12)+126]= 14,994	0.94613	0.95277	0.94611	**0.96694**
		[(2489x12)+126]= 29,994	0.94613	0.95144	0.94611	**0.96694**
	Unigram-Bigram-Trigram	[(406x12)+126]= 4,998	0.88441	0.90369	0.80481	0.89151
		[(1239x12)+126]= 14,994	0.91899	0.93067	0.88277	0.94435
		[(2489x12)+126]= 29,994	0.93965	0.94906	0.91836	**0.96305**

increased, with both achieving 99% average F1 scores on subsets containing 30,000 instances.

The results presented in Tables II and III reveals a consistent trend favoring CNNs, unigram models, and larger datasets in achieving higher average F1 scores. Conversely, Logistic Regression, trigram models, and models trained on smaller datasets generally exhibited lower average F1 scores. Furthermore, balanced subsets tended to yield higher average F1 scores with unigram, unigram-bigram, and unigram-bigram-trigram models compared to imbalanced subsets. Conversely, imbalanced subsets demonstrated superior performance with bigram and trigram models.

TABLE III. AVERAGE F1 SCORES FOR THE DL MODELS

	Data Size	CNN	LSTM
Imbalanced	5,000	0.85585	0.90278
	15,000	0.98924	0.97719
	30,000	**0.99501**	0.99081
Balanced	[(406x12)+126]= 4998	0.96220	0.86296
	[(1239x12)+126]= 14994	0.98863	0.96811
	[(2489x12)+126]= 29994	**0.99220**	0.98060

VI. CONCLUSION

This study conducted a comparative analysis of traditional ML models and DL architectures for SA, utilizing the "Emotion analysis based on text" dataset. Models were evaluated across various n-gram configurations and data subsets to provide a comprehensive understanding of their performances. Our findings demonstrate the superior performance of DL architectures, particularly CNNs and LSTMs, in SA tasks. While CNNs generally outperformed LSTMs, the performance gap narrowed as the dataset size increased. Notably, both models achieved a remarkable 99% average F1 score on subsets containing 30,000 instances. This effort underscores the potential of DL approaches for advancing SA capabilities. Future research can build upon these findings by exploring more sophisticated DL architectures and investigating their generalization abilities across diverse datasets.

ACKNOWLEDGMENT

We gratefully acknowledge the insightful feedback and valuable suggestions provided by the paper reviewers.

REFERENCES

[1] M. Birjali, M. Kasri, and A. Beni-Hssane, "A comprehensive survey on sentiment analysis: Approaches, challenges and trends," *Knowledge-Based Systems*, vol. 226, p. 107134, 2021.

[2] B. Erkantarci and G. Bakal, “An empirical study of sentiment analysis utilizing machine learning and deep learning algorithms,” *Journal of Computational Social Science*, pp. 1–17, 2023.
[3] P. Nandwani and R. Verma, “A review on sentiment analysis and emotion detection from text,” *Social net-work analysis and mining*, vol. 11, no. 1, p. 81, 2021.
[4] J. Guo, Y. Fan, L. Pang, L. Yang, Q. Ai, H. Zamani, C. Wu, W. B. Croft, and X. Cheng, “A deep look into neural ranking models for information retrieval,” *Information Processing & Management*, vol. 57, no. 6, p. 102067, 2020.
[5] A. Fan, S. Bhosale, H. Schwenk, Z. Ma, A. El-Kishky, S. Goyal, M. Baines, O. Celebi, G. Wenzek, V. Chaudhary *et al.*, “Beyond english-centric multilin-gual machine translation,” *Journal of Machine Learning Research*, vol. 22, no. 107, pp. 1–48, 2021.
[6] G. Bakal and O. Abar, “On comparative classification of relevant covid-19 tweets,” in *2021 6th Interna-tional Conference on Computer Science and Engineer-ing (UBMK)*. IEEE, 2021, pp. 287–291.
[7] G. Bakal and R. Kavuluru, “Predicting treatment rela-tions with semantic patterns over biomedical knowledge graphs,” in *International Conference on Mining Intelli-gence and Knowledge Exploration*. Springer, 2015, pp. 586–596.
[8] B. Bozkurt, K. Coskun, and G. Bakal, “Building a challenging medical dataset for comparative evaluation of classifier capabilities,” *Computers in Biology and Medicine*, vol. 178, p. 108721, 2024.
[9] S. Rao, A. K. Verma, and T. Bhatia, “A review on social spam detection: Challenges, open issues, and future directions,” *Expert Systems with Applications*, vol. 186, p. 115742, 2021.
[10] B. Bhutani, N. Rastogi, P. Sehgal, and A. Purwar, “Fake news detection using sentiment analysis,” in *2019 twelfth international conference on contemporary com-puting (IC3)*. IEEE, 2019, pp. 1–5.
[11] T. Ü. Şen and G. Bakal, “A transfer learning application on the reliability of psychological drugs’ comments,” in *2023 International Conference on Smart Applications, Communications and Networking (SmartNets)*. IEEE, 2023, pp. 1–6.
[12] L. P. Hung and S. Alias, “Beyond sentiment analysis: A review of recent trends in text based sentiment analysis and emotion detection,” *Journal of Advanced Computational Intelligence and Intelligent Informatics*, vol. 27, no. 1, pp. 84–95, 2023.
[13] P. Shaver, J. Schwartz, D. Kirson, and C. O’connor, “Emotion knowledge: further exploration of a prototype approach.” *Journal of personality and social psychology*, vol. 52, no. 6, p. 1061, 1987.
[14] W. G. Parrott, *Emotions in social psychology: Essential readings*. psychology press, 2001.
[15] A. Ortony and T. J. Turner, “What’s basic about basic emotions?” *Psychological review*, vol. 97, no. 3, p. 315, 1990.
[16] A. Ortony, G. L. Clore, and A. Collins, *The cognitive structure of emotions*. Cambridge university press, 2022.
[17] S. Peng, L. Cao, Y. Zhou, Z. Ouyang, A. Yang, X. Li, W. Jia, and S. Yu, “A survey on deep learning for textual emotion analysis in social networks,” *Digital Communications and Networks*, vol. 8, no. 5, pp. 745–762, 2022.
[18] S. Anjali, “emotion analysis based on text,” https://www.kaggle.com/datasets/simaanjali/emotion-analysis-based-on-text, 2024, accessed: March 22, 2024.
[19] J. Liu, J. Chen, and J. Ye, “Large-scale sparse logistic regression,” *Knowledge Discovery and Data Mining*, 2009.
[20] M. P. LaValley, “Logistic regression,” *Circulation*, vol. 117, no. 18, pp. 2395–2399, 2008. [Online]. Available: https://doi.org/10.1161/circulationaha.106.682658
[21] N. Agarwal, S. Kale, and J. Zimmert.(2021, 10) Efficient methods for online mul-ticlass logistic regression. [Online]. Available: https://doi.org/10.48550/arXiv.2110.03020
[22] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg *et al.*, “Scikit-learn: Machine learning in python,” *the Journal of machine Learning research*, vol. 12, pp. 2825–2830, 2011.
[23] L. Breiman, “Random forests,” *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001. [Online]. Available: https://doi.org/10.1023/a:1010933404324
[24] M. Goudjil, M. Koudil, M. Bedda, and N. Ghoggali, “A novel active learning method using svm for text classification,” *International Journal of Automation and Computing*, vol. 15, no. 3, pp. 290–298, 2016. [Online]. Available: https://doi.org/10.1007/s11633-015-0912-z
[25] T. Chen and C. Guestrin, “Xgboost: a scalable tree boosting system,” in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2016, pp. 785–794. [Online]. Available: https://doi.org/10.1145/2939672.2939785
[26] Y.-D. Zhang, Z. Dong, X. Chen, W. Jia, S. Du, K. Muhammad, and S.-H. Wang, “Image based fruit category classification by 13-layer deep convolutional neural network and data augmentation,” *Multimedia Tools and Applications*, vol. 78, no. 3, pp. 3613–3632, 2017. [Online]. Available: https://doi.org/10.1007/s11042-017-5243-3
[27] A. Gulli and S. Pal, *Deep learning with Keras*. Packt Publishing Ltd, 2017.
[28] Y. Luan and S. Lin, “Research on text classification based on cnn and lstm,” in *2019 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA)*, 2019. [Online]. Available: https://doi.org/10.1109/icaica.2019.8873454
[29] A. Sherstinsky, “Fundamentals of recurrent neural net-work (rnn) and long short-term memory (lstm) net-work,” *Physica D: Nonlinear Phenomena*, vol. 404, p. 132306, 2020.
[30] L. Muflikhah, I. Cholissodin, N. Widodo, F. E. Herman, T. L. Wargasetia, H. Ratnawati, and R. Sarno, “Single nucleotide polymorphism based on hypertension potential risk prediction using lstm with adam optimizer,” *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 33, no. 2, pp. 1126–1139, 2024. [Online]. Available: https://doi.org/10.11591/ijeecs.v33.i2.pp1126-1139