Journal of Systemics, Cybernetics and Informatics (2025) 23(6), 36-41
https://doi.org/10.54808/JSCI.23.06.36

Detecting AI-Generated Text: A Comparative Study of Machine Learning Algorithms

Li-jing Arthur CHANG, PhD-Journalism, PhD-CDS&E[1]
Department of Journalism and Media Studies, Jackson State University
Jackson, MS 39204, USA

ABSTRACT

As ChatGPT and other large-language-model (LLM) tools have made the generation of text via AI much easier than before, there is an increasing need to determine if humans indeed write the text we are reading. The study used six machine learning and deep learning algorithms to detect AI-created text. Using a balanced sample of AI-generated and human-written text, the results showed that deep learning algorithms outperformed their machine learning counterparts. A hybrid deep learning algorithm achieved the top accuracy rate of 0.974 (or 97.4%). Post hoc analysis showed that with a small fraction, such as 10%, of the full sample used in the present study, the hybrid algorithm achieved 0.928.

Keywords: AI-generated text, text classification, machine learning, deep learning

1. INTRODUCTION

With the rising popularity of text generated by artificial intelligence (AI), the ability to spot AI text has become an urgent task for various social, educational, economic, and political reasons. In recent years, with the advent of ChatGPT and other LLM (large language model) AI tools, the influx of AI-generated text has been on the rise in various facets of society, including student assignments, social media posts, and news content[1][2][3]. This development has far-reaching impacts on multiple sectors of society, including education, research, news, financial markets, and politics[1][3][4][5][6].

For example, one of the teachers' major tasks in the past few years has been determining whether the students' submitted assignments are AI-generated. Discernment is essential because failure to detect AI-generated student submissions affects the fairness of the grading system[1]. As another example, AI-generated content may have impact on election campaigns[7]. In a related issue, AI-generated misinformation during a pandemic could have public health consequences[8]. Also, AI-fabricated misinformation may influence financial market movements[5].

As the impacts of AI-generated text have become more widespread and far-reaching, the ability to detect AI text from human-written text has become increasingly crucial to ease the effects of AI text in the education, financial, and political sectors, to name a few.

The study explores the use of various popular machine learning algorithms to detect AI-generated text. The performance of these algorithms is evaluated with common evaluation standards, such as detection accuracy rates and other statistical measures. The research also identified the top 10 words that appeared in AI-generated text and their human-written counterparts.

2. LITERATURE REVIEW

The study used several popular machine learning and deep learning algorithms to classify AI-generated and human-written text, including support vector machine, logistic regression, XGBoost, and Naïve Bayes[9][10][11][12][12][13][14][15][16]. Besides machine learning algorithms, deep learning algorithms such as Long Short-Term Memory (LSTM) have been widely used to detect AI text[14][17]. Another good candidate for this task is CNN-LSTM, a hybrid of CNN (Convolutional Neural Network) and LSTM that can often achieve a higher accuracy rate than either of the algorithms alone[18][19]. The present study aims to apply these ML and DL algorithms in AI text detection and compare the model's performance.

Support vector machine is suited for text classification because it can find hyperplanes in high-dimensional spaces to separate the text classes[20]. For the algorithm, text input is usually transformed into numerical features such as TF-IDF vectors[21]. Support vector machine can process sparse, high-dimensional data, such as the case of text[21].

Logistic regression requires low computational resources and produces interpretable models for text classification[22][23]. It transforms features like TF-IDF to class probabilities[22]. With its effectiveness on sparse, high-dimensional data like text, the algorithm is a strong candidate for text classification[22].

XGBoost is a gradient boosting method that can effectively process sparse, high-dimensional data like TF-IDF transformed text features[24]. It can minimize classification loss via regularization to limit overfitting[25]. The algorithm performs well in text classification tasks such as product review classification[24].

Naïve Bayes is derived from Bayes' theorem and is a probabilistic classifier with the conditional independence assumption for the features of the data[26]. It requires minimum resources for text classification, using feature representations like TF-IDF or bag-of-words[24][26]. The algorithm can still offer fast yet robust performance in situations with small data and limited computing power[26].

LSTM can classify text with its ability to model sequential data and grasp long-term patterns[27]. After importing text as word embeddings, LSTM processes the data to learn contextual

[1] CDS&E stands for Computational and Data-Enabled Science and Engineering.

semantics. During the process, LSTM can handle variable-length sequences, deal with vanishing gradients, and capture linguistic patterns[27]. The final hidden states are used for classification[27].

A hybrid architecture combining CNN (Convolutional Neural Network) and LSTM can allow the algorithms to complement each other[28]. The strength of CNN is its ability to capture local features through convolution and pooling, while the advantage of LSTM is its capacity to capture sequential information and contextual dependencies[28][29]. In this architecture, CNN first extracts detailed information from the local features, and when LSTM takes over, it allows these extracted local features to learn their contextual and sequential information[28][29].

As the above literature review shows, all the above six machine learning and deep learning algorithms are suitable for text classification tasks such as detection of AI-generated text. As most (or five) of the algorithms have been used in previous research to detect AI text, it would interesting to see how all of the these six algorithms are compared in their performances for AI text detection. Therefore, the following research question is generated:

RQ: How do the machine learning and deep learning algorithms perform in detecting AI-generated text?

3. METHODS

The dataset for the study comes from the HC3 dataset, which has a sample size of 48,644, with half of the data as AI-generated text and the other half as human-written text. After dropping missing values for the dataset, the sample size of the data was reduced to 48,187, with 24,322 as human text and 23,865 as AI text. A randomized procedure is employed to balance the two classes to capitalize on the advantage of balanced classes for text classification[30]. As a result, the sample size of both AI and human text was equalized to be 23,865 each, with the total sample size at 47,730.

To test the machine learning models (i.e., support vector machines, logistic regression, XGBoost, and Naïve Bayes), the text of the full sample is tokenized through a procedure called TF-IDF, or Term Frequency–Inverse Document Frequency, which highlights words that are frequent in a specific document but infrequent across the whole text dataset[31]. This way, the preprocessed data will be more useful for keyword extraction, text classification, and information retrieval[31]. The tokenized text data is further transformed for feature reduction using principal component analysis (PCA)[32]. The data is converted to PCA components for feature reduction.

The preprocessed text is then randomly divided into train and test datasets with an 80:20 ratio. Each machine learning model is first trained through a grid search 10-fold cross-validation process using the train dataset to search for the best parameters for accuracy. Afterwards, each trained model is tested on the test dataset to evaluate the model.

For the deep learning algorithms of LSTM and CNN-LSTM, the original text data was first transformed into an embedded layer with the sentence sequence set at 1100. Then, the transformed text data is randomly split into train and test datasets with the same 80:20 ratio. The train dataset is then fed to each deep learning model for training in 10 epochs with early stopping specified. Afterwards, each trained model is tested on the test dataset to evaluate the model.

To assess the trained models, the study used two complementary sets of evaluation metrics: (1) ROC, or the Receiver Operating Characteristic curve, and AUC, or Area Under the Curve, and (2) accuracy, precision, recall, and F1 score[33].

The ROC and AUC metrics assess the models' capacity to discriminate between the classes for all likely classification thresholds. The ROC curve plots True Positive Rate (TPR, or recall) against False Positive Rate (FPR), showing the trade-off between sensitivity (TPR) and 1-specificity (FPR)[34]. The AUC is a statistic summarizing this trade-off, with higher values (closer to 1.0) showing stronger distinguishing performance[34].

Unlike the ROC and AUC metrics, accuracy, precision, recall, and F1 score are threshold-dependent and computed on a specified classification boundary[35]. Accuracy is a measure of overall accuracy[36]. Precision measures the rate of true positives among all positive predictions[36]. Recall reflects the rate of actual positive instances spotted[36]. To balance precision and recall, the F1 score is the harmonic mean of the two measures[36]. The F1 score is helpful when false positives and false negatives concern researchers[36].

By combining ROC/AUC and threshold-dependent metrics, the study offers a more comprehensive assessment of model performance. Such evaluation grasps both the fixed-threshold classification capability and the distinguishing effectiveness across different thresholds of a decision cutoff.

4. RESULTS

Before training the text data for the machine learning and deep learning models, the text data was first evaluated for feature importance for AI-generated and human-written text to see the top AI and human words in the dataset.

To obtain feature importance, the study preprocessed the text via tokenization, removal of stop words, and lemmatization. The preprocessed text underwent TF-IDF transformation to become vectors of highly informative terms[31]. The transformed data was split randomly into train and test data. A logistic regression model is trained on the train data to get the coefficients for all features (i.e., words) and rank their contribution by the absolute values of the coefficients.

The feature importance analysis showed the leading AI-generated words in the dataset are "important," "help," "including," "may," "might," "overall," "however," "difficult,", "able," and "used" (in the order of importance, see Figure 1). These words appear to be more related to logical reasoning.

The analysis also found the top human-written words are "etc," "basically," "pretty," "probably," "edit," "since," "thus," "really," "get," and "stuff" (in order of importance, see Figure 2). These words seem to be more related to people's daily conversation.

The present study aims to see which of the six popular machine learning and deep learning models would perform better. The evaluation was done with a model trained on the train dataset and

Figure 1. Top AI Words (Feature Importance)

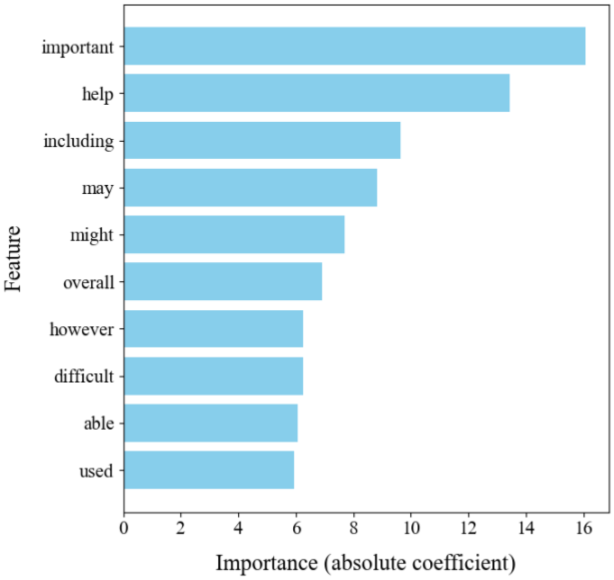
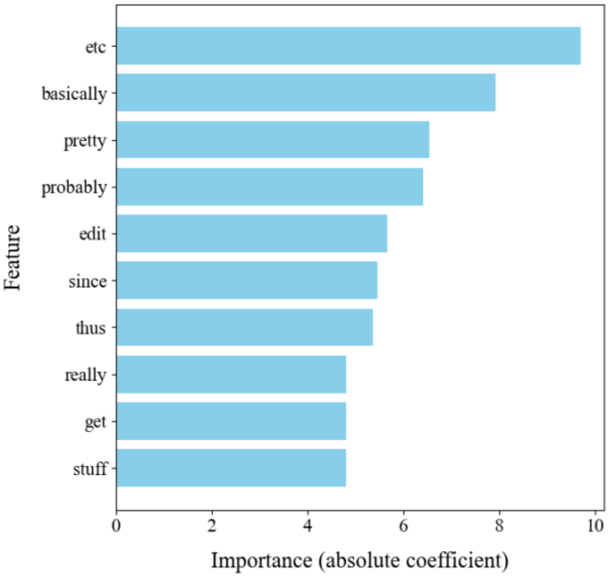


Figure 2. Top Human Words (Feature Importance)

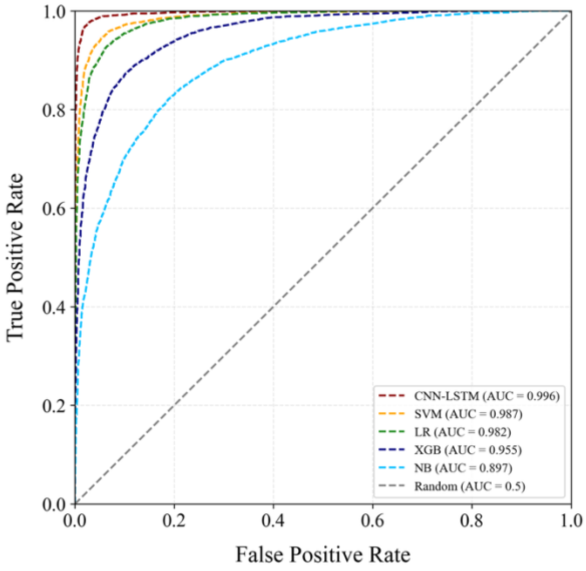


tested on the test dataset. Two sets of evaluation metrics are used: (1) ROC/AUC, and (2) accuracy, precision, recall, and F1 score.

The ROC/AUC findings revealed CNN-LSTM has the most significant AUC, or area under the ROC curve of 0.996, followed by LSTM (0.994), support vector machine (0.987), logistic regression (0.982), XGBoost (0.955), and Naïve Bayes (0.897) (see the Figure 3; LSTM figure is omitted due to its almost identical value compared to the CNN-LSTM figure). The ROC/AUC readings show that CNN-LSTM, a hybrid deep learning algorithm, has the highest capacity to distinguish between AI and human text, regardless of the decision threshold.

Unlike ROC/AUC testing, the accuracy, precision, recall, and F1 score metrics require a fixed threshold. With its preprocessed data as a balanced sample, the study adopted the traditional default threshold of 0.5 for both the machine learning algorithms (i.e., support vector machine, logistic regression, XGBoost, and Naïve Bayes) and their deep learning counterparts (LSTM and CNN-LSTM)[37].

Figure 3. Top AI Words (Feature Importance)



Note: LSTM is omitted because its AUC value (0.994) is almost indistinguishable from that of CNN-LSTM.
Abbreviations: SVM (support vector machine), LR (logistic regression), XGB (XGBoost), and NB (Naïve Bayes).

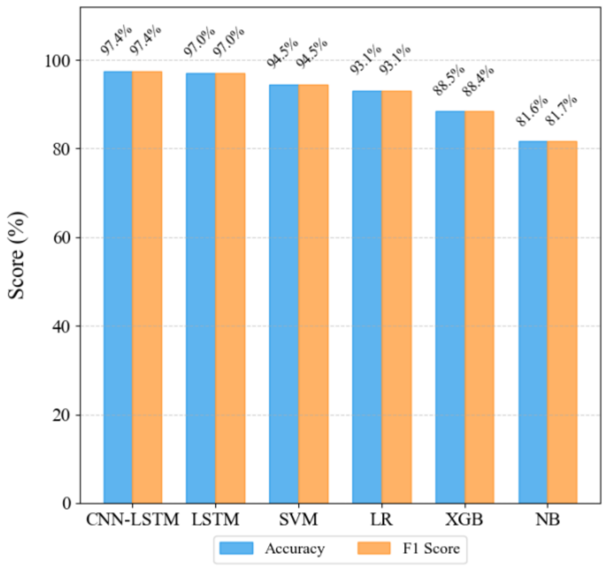
As the F1 score summarize precision and recall measures, a good summary of the model performance can be demonstrated by accuracy and the F1 scores. Among the six algorithms tested, CNN-LSTM leads with 0.974 (or 97.4%) for both accuracy and F1 score, followed by LSTM (0.970 for both statistics), support vector machine (0.945 for both figures), logistic regression (0.931 for both numbers), XGBoost (0.885 and 0.884 for accuracy and F1 score, respectively), and Naïve Bayes (0.816 and 0.817 for accuracy and F1 score) (see Figure 4).

When looking at both sets of evaluation metrics together, CNN-LSTM is the top performer, with the highest accuracy, recall, F1 score, and AUC (see the Table 1, with top figures of each metric in boldface). LSTM has the highest score in precision.

5. DISCUSSION

The study aims to experiment with six machine learning and deep learning models to detect AI-generated text. Before the start of the experiment, the study used feature importance analysis to identify the top 10 words for AI-generated text and its human-written text. From their appearances, the AI text’s top words seem more logistically oriented, while their human text counterparts appear more conversational. Post hoc analysis of these groups regarding their linguistic distance showed that they are two groups, each occupying a different linguistic area (see Figure 5).

Figure 4. Model Comparison: Accuracy vs. F1 Score



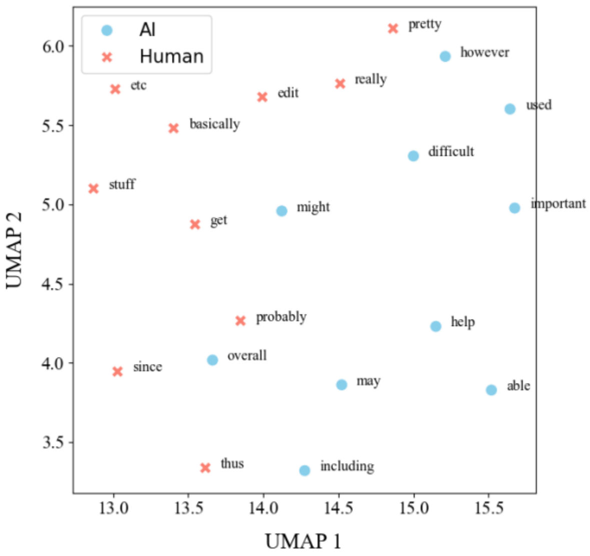
Abbreviations: SVM (support vector machine), LR (logistic regression), XGB (XGBoost), and NB (Naïve Bayes).

Table 1. Model Performances Comparison

Algorithm	Accuracy	Precision	Recall	F1	AUC
CNN-LSTM	**0.974**	0.971	**0.978**	**0.974**	**0.996**
LSTM	0.970	**0.978**	0.962	0.970	0.994
SVM	0.945	0.944	0.947	0.945	0.987
LR	0.931	0.930	0.933	0.931	0.982
XGB	0.885	0.890	0.879	0.884	0.955
NB	0.816	0.815	0.819	0.817	0.897

Abbreviations: SVM (support vector machine), LR (logistic regression), XGB (XGBoost), and NB (Naïve Bayes).

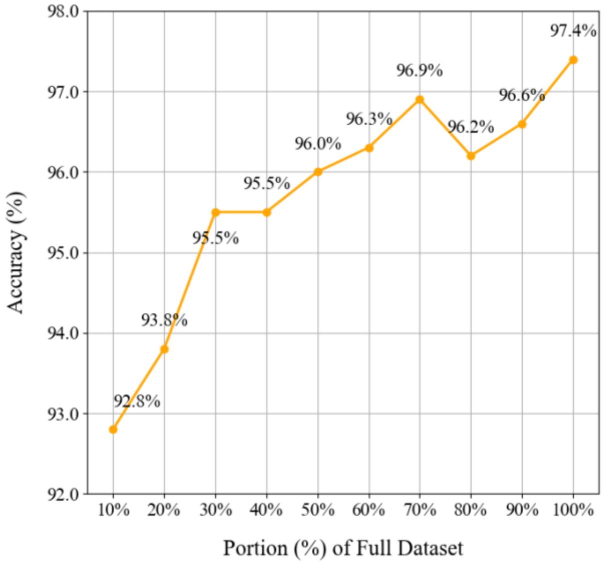
Figure 5. Semantic Analysis of AI vs. Human Words



After experimenting with six machine learning and deep learning models and comparing their performances, the results showed that overall, when both classification threshold independent and dependent metrics are considered, CNN-LSTM performs the best, almost across the board (i.e., in terms of accuracy, recall, F1 score, and AUC). LSTM is the top performer for the precision metric. In overall performance, deep learning algorithms (CNN-LSTM and LSTM) outshine their machine learning counterparts (support vector machine, logistic regression, XGBoost, and Naïve Bayes).

Further examination of CNN-LSTM showed that the algorithm can perform well with a portion of the dataset (see Figure 6). Compared to the 97.4% (or 0.994) accuracy at the whole dataset, the algorithm scored 96.0% (or 0.960) accuracy with 50% of the data. As another example, with 10% of the full sample, the accuracy is 92.8% (or 0.928).

Figure 6. Best Model Accuracy vs. Portion of Full Dataset



6. CONCLUSIONS

The study showed the effectiveness of using machine learning and deep learning models to detect AI-generated text. Regarding the threshold-independent metric (ROC/AUC), all but one of the six models tested achieved an AUC score over 0.950 (or 95%). Regarding the fixed threshold metrics like accuracy, precision, recall, and F1 score, four models obtained scores of at least 0.930 (or 93%). The performances highlighted the models' capacity to capture AI text's complex semantic and linguistic patterns. Furthermore, the feature importance analysis via TF-IDF and logistic regression offered insights into the distinguishing lexical characteristics contributing to the classification.

The findings show the models' utility in detecting AI text in educational settings and Internet misinformation. However, testing the models should be ongoing as AI-generated text continuously rises. As the findings showed the promise of deep learning algorithms, incorporating transformer-based deep learning classifiers such as BERT and RoBERTa may further enhance detection accuracy and classification robustness[38].

7. REFERENCES

[1] R. Safi and A. J. Naini, "The Work of Students and ChatGPT Compared: Using Machine Learning to Detect and Characterize AI-Generated Text.," in *AMCIS*, 2023.

[2] D. Xu, S. Fan, and M. Kankanhalli, "Combating misinformation in the era of generative AI models," in *Proceedings of the 31st ACM International Conference on Multimedia*, 2023, pp. 9291–9298.

[3] A. F. Sonni, H. Hafied, I. Irwanto, and R. Latuheru, "Digital Newsroom Transformation: A Systematic Review of the Impact of Artificial Intelligence on Journalistic Practices, News Narratives, and Ethical Challenges," *Journalism and Media*, vol. 5, no. 4, pp. 1554–1570, 2024.

[4] A. Korinek, "Language models and cognitive automation for economic research," 2023.

[5] Z. Karaş, "Effects of AI-Generated Misinformation and Disinformation on the Economy," *Düzce Üniversitesi Bilim ve Teknoloji Dergisi*, vol. 12, no. 4, pp. 2349–2360, 2024.

[6] P. L. Kharvi, "Understanding the Impact of AI-Generated Deepfakes on Public Opinion, Political Discourse, and Personal Security in Social Media," *IEEE Secur Priv*, 2024.

[7] A. K. Shukla and S. Tripathi, "AI-generated misinformation in the election year 2024: measures of European Union," *Front Polit Sci*, vol. 6, p. 1451601, 2024.

[8] J. Zhou, Y. Zhang, Q. Luo, A. G. Parker, and M. De Choudhury, "Synthetic lies: Understanding ai-generated misinformation and evaluating algorithmic and human solutions," in *Proceedings of the 2023 CHI conference on human factors in computing systems*, 2023, pp. 1–20.

[9] O. Sharif, M. M. Hoque, A. S. M. Kayes, R. Nowrozy, and I. H. Sarker, "Detecting suspicious texts using machine learning techniques," *Applied Sciences*, vol. 10, no. 18, p. 6527, 2020.

[10] H. Alamleh, A. A. S. AlQahtani, and A. ElSaid, "Distinguishing human-written and ChatGPT-generated text using machine learning," in *2023 Systems and Information Engineering Design Symposium (SIEDS)*, 2023, pp. 154–158.

[11] G. Levin, R. Meyer, P.-A. Guigue, and Y. Brezinov, "It takes one to know one—Machine learning for identifying OBGYN abstracts written by ChatGPT," *International Journal of Gynecology & Obstetrics*, vol. 165, no. 3, pp. 1257–1260, 2024.

[12] T. T. Nguyen, A. Hatua, and A. H. Sung, "How to detect AI-generated texts?," in *2023 IEEE 14th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*, 2023, pp. 464–471.

[13] F. Greco, G. Desolda, A. Esposito, A. Carelli, and others, "David versus Goliath: Can Machine Learning Detect LLM-Generated Text? A Case Study in the Detection of Phishing Emails," in *The Italian Conference on CyberSecurity*, 2024.

[14] K. Hayawi, S. Shahriar, and S. S. Mathew, "The imitation game: Detecting human and AI-generated texts in the era of ChatGPT and BARD," *J Inf Sci*, p. 01655515241227531, 2024.

[15] N. Prova, "Detecting ai generated text based on nlp and machine learning approaches," *arXiv preprint arXiv:2404.10032*, 2024.

[16] M. M. Oghaz, L. B. Saheer, K. Dhame, and G. Singaram, "Detection and classification of ChatGPT-generated content using deep transformer models," *Front Artif Intell*, vol. 8, p. 1458707, 2025.

[17] M. A. Wani, M. ElAffendi, and K. A. Shakil, "AI-Generated Spam Review Detection Framework with Deep Learning Algorithms and Natural Language Processing.," *Computers (2073-431X)*, vol. 13, no. 10, 2024.

[18] A. Ghourabi, M. A. Mahmood, and Q. M. Alzubi, "A hybrid CNN-LSTM model for SMS spam detection in Arabic and English messages," *Future Internet*, vol. 12, no. 9, p. 156, 2020.

[19] I. S. Mambina, J. D. Ndibwile, D. Uwimpuhwe, and K. F. Michael, "Uncovering sms spam in swahili text using deep learning approaches," *IEEE Access*, vol. 12, pp. 25164–25175, 2024.

[20] W. Zhang, T. Yoshida, and X. Tang, "Text classification based on multi-word with support vector machine," *Knowl Based Syst*, vol. 21, no. 8, pp. 879–886, 2008.

[21] T. Joachims, "Text categorization with support vector machines: Learning with many relevant features," in *European conference on machine learning*, 1998, pp. 137–142.

[22] R. Kholwal, "Text-Classify: A comprehensive comparative study of logistic regression, random forest, and knn models for enhanced text classification performance," *Int J Adv Eng Technol*, vol. 16, no. 5, pp. 415–433, 2023.

[23] T. Ling, L. Jake, J. Adams, K. Osinski, X. Liu, and D. Friedland, "Interpretable machine learning text classification for clinical computed tomography reports–a case study of temporal bone fracture," *Computer Methods and Programs in Biomedicine Update*, vol. 3, p. 100104, 2023.

[24] I. R. Hendrawan, E. Utami, and A. D. Hartanto, "Comparison of na\"\ive bayes algorithm and XGBoost on local product review text classification," *Edumatic: Jurnal Pendidikan Informatika*, vol. 6, no. 1, pp. 143–149, 2022.

[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*, 2016, pp. 785–794.

[26] H. Zhang and D. Li, "Naive Bayes text classifier," in *2007 IEEE international conference on granular computing (GRC 2007)*, 2007, p. 708.

[27] X. Bai, "Text classification based on LSTM and attention," in *2018 Thirteenth international conference on digital information management (ICDIM)*, 2018, pp. 29–32.

[28] A. Ghourabi, M. A. Mahmood, and Q. M. Alzubi, "A hybrid CNN-LSTM model for SMS spam detection in Arabic and English messages," *Future Internet*, vol. 12, no. 9, p. 156, 2020.

[29] I. S. Mambina, J. D. Ndibwile, D. Uwimpuhwe, and K. F. Michael, "Uncovering sms spam in swahili text using deep learning approaches," *IEEE Access*, vol. 12, pp. 25164–25175, 2024.

[30] E. Stamatatos, “Author identification: Using text sampling to handle the class imbalance problem,” *Inf Process Manag*, vol. 44, no. 2, pp. 790–799, 2008.

[31] S. Qaiser and R. Ali, “Text mining: use of TF-IDF to examine the relevance of words to documents,” *Int J Comput Appl*, vol. 181, no. 1, pp. 25–29, 2018.

[32] J. Jotheeswaran, R. Loganathan, and B. Madhu Sudhanan, “Feature reduction using principal component analysis for opinion mining,” *International Journal of Computer Science and Telecommunications*, vol. 3, no. 5, pp. 118–121, 2012.

[33] T. Hastie, R. Tibshirani, J. Friedman, and others, “The elements of statistical learning,” 2009, *Citeseer*.

[34] T. Fawcett, “An introduction to ROC analysis,” *Pattern Recognit Lett*, vol. 27, no. 8, pp. 861–874, 2006.

[35] F. Provost and T. Fawcett, “Robust classification for imprecise environments,” *Mach Learn*, vol. 42, pp. 203–231, 2001.

[36] M. Sokolova and G. Lapalme, “A systematic analysis of performance measures for classification tasks,” *Inf Process Manag*, vol. 45, no. 4, pp. 427–437, 2009.

[37] E. A. Freeman and G. G. Moisen, “A comparison of the performance of threshold criteria for binary classification in terms of predicted prevalence and kappa,” *Ecol Modell*, vol. 217, no. 1–2, pp. 48–58, 2008.

[38] J. Briskilal and C. N. Subalalitha, “An ensemble model for classifying idioms and literal texts using BERT and RoBERTa,” *Inf Process Manag*, vol. 59, no. 1, p. 102756, 2022.