FUDMA Journal of Sciences (FJS)
ISSN online: 2616-1370
ISSN print: 2645 - 2944
Vol. 9 No. 2, February, 2025, pp 35 - 41
DOI: https://doi.org/10.33003/fjs-2025-0902-3058

INTEGRATION OF LAYER-WISE RELEVANCE PROPAGATION, RECURSIVE DATA PRUNING, AND CONVOLUTIONAL NEURAL NETWORKS FOR IMPROVED TEXT CLASSIFICATION

[1]Abubakar Ado, [2]Olalekan J. Awujoola, *[1]Sabiu Danlami Abdullahi and [1]Sulaiman Hashim Ibrahim

[1]Yusuf Maitama Sule University Kano, Computer Science Department
[2]Nigerian Defence Academy, Computer Science Department

*Corresponding authors' email: sabiudanlami@gmail.com

ABSTRACT

This research presents a significant advancement in text classification by integrating Layer-wise Relevance Propagation (LRP), recursive data pruning, and Convolutional Neural Networks (CNNs) with cross-validation. The study addresses the critical limitations of existing text classification methods, particularly issues of information loss and overfitting, which often hinder the efficiency and interpretability of models in natural language processing (NLP). To overcome these challenges, the proposed model employs LRP to enhance the interpretability of the classification process, allowing for precise identification of relevant features that contribute to decision-making. Additionally, the implementation of recursive data pruning optimizes model efficiency by dynamically eliminating irrelevant or redundant data, thereby reducing computational complexity without compromising performance. The effectiveness of the approach is further bolstered by utilizing cross-validation techniques to ensure robust evaluation across diverse datasets. The empirical evaluation of the integrated model revealed remarkable improvements in classification performance, achieving an accuracy of 94%, surpassing the benchmark of 92.88% established by the ReDP-CNN model proposed by Li et al. (2020). The comprehensive assessment included detailed metrics such as precision, recall, and F1-score, confirming the model's robust capability in accurately classifying text data across various categories.

Keywords: Natural Language Processing, Layer-wise Relevance Propagation, Convolutional Neural Network

INTRODUCTION

Text classification is a pivotal component of natural language processing (NLP), focusing on the automatic organization of textual data into predefined categories. It underpins applications across diverse domains, including spam filtering, sentiment analysis, legal document organization, and information extraction. These applications are integral to commercial enterprises, offering capabilities that reduce operational costs and enhance decision-making efficiency. For example, Hassan et al. (2022) illustrated how machine learning techniques in text classification streamline business operations and provide actionable insights. Despite its wide-ranging utility, processing complex and voluminous text data presents persistent challenges, as highlighted by Romero et al. (2022). Their findings underscore the growing demand for advanced classification methods that address these complexities effectively.

Text constitutes over 80% of unstructured data, representing a dominant and underutilized data category due to its inherently disorganized and intricate nature. Extracting meaningful insights from such data requires robust classification methodologies. Bashir et al. (2022) emphasized how text classification can address this challenge by automating the extraction of valuable information, fostering data-driven decision-making. Machine learning (ML) techniques significantly contribute to these endeavours, enabling the categorization of diverse text types, such as social media posts, emails, and academic articles. Studies by Abbasi et al. (2021, 2022) and Hina et al. (2021a) corroborate the efficacy of ML techniques in improving text organization and analysis, demonstrating their capacity to enhance productivity and efficiency.

Traditional machine learning approaches, including Support Vector Machines (SVM), Random Forest (RF), and Naive Bayes (NB), have shown significant success in text classification tasks. However, these methods often rely on manually crafted features, which demand extensive domain expertise and are computationally intensive. Recent advancements in deep learning (DL) have mitigated these limitations, as DL models can automatically learn intricate patterns from data without requiring manual feature engineering. Kim and Jeong (2019) demonstrated how convolutional neural networks (CNNs) excel in text classification by capturing hierarchical structures in textual data, outperforming traditional methods.

Several researchers have explored innovative strategies to enhance the effectiveness of DL models in text classification. Çoban et al. (2021) applied recurrent neural networks (RNNs) to sentiment analysis using Turkish Facebook data, achieving a remarkable accuracy of 91.6%. Their work highlighted the capability of RNNs to capture sequential dependencies in text, making them ideal for tasks requiring contextual understanding. Similarly, Dogru et al. (2021) utilized Doc2vec word embedding with CNNs, achieving 94.17% accuracy on Turkish datasets. Their approach underlines the importance of language-specific adaptations and the power of DL in addressing linguistic complexities. Zulqarnain et al. (2021) leveraged word2vec embeddings alongside CNNs, Long Short-Term Memory (LSTM), and Gated Recurrent Units (GRU) for Turkish question classification, achieving an impressive accuracy of 93.7%. These studies collectively illustrate the efficacy of combining advanced embedding techniques with DL models to enhance text classification performance.

While deep learning offers significant advantages, it is not without challenges. Li et al. (2020) identified issues of overfitting and inefficiency in CNN-based text classification models. Their work introduced pruning techniques to eliminate task-irrelevant words, reducing model complexity and enhancing accuracy. Building on this, Layer-wise Relevance Propagation (LRP) has emerged as a promising solution to improve pruning effectiveness. LRP assigns relevance scores to individual neurons, enabling targeted pruning while preserving model performance.

Figure 1 illustrates the Flowchart of the LRP Method for CNN, outlining the step-by-step process of relevance score computation and propagation. The flowchart demonstrates how relevance scores are backpropagated from the output layer to the input layer, ensuring that only the most significant features contribute to the final classification. By integrating LRP with CNNs, an efficient pruning strategy can be achieved, addressing computational and scalability concerns while maintaining high classification accuracy.

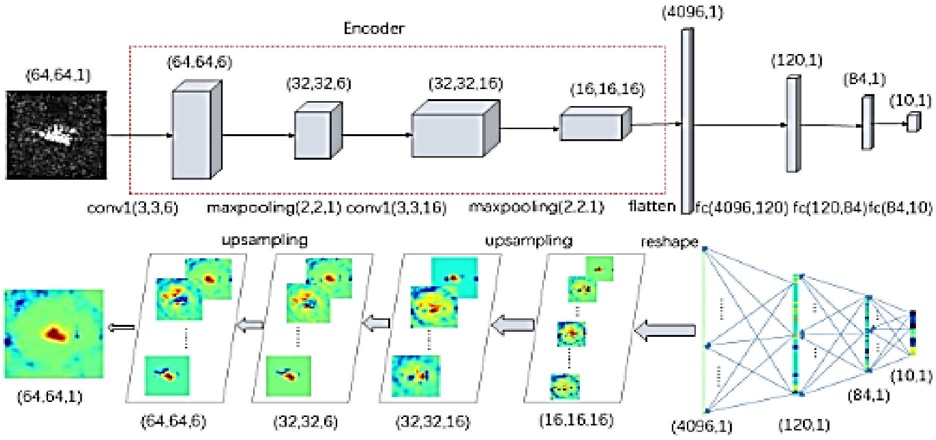

Figure 1: Flowchart of LRP method for CNN (Source: Zang et al.,2021)

Li and Li (2022) introduced the Trusted Platform Module (TPM) algorithm, integrating machine learning and NLP for multilingual text classification. Their study achieved over 95% accuracy in distinguishing spam from legitimate emails, showcasing the potential of hybrid algorithms in addressing language and complexity challenges.

Recent studies have also explored the application of DL models in niche domains. Alqahtani et al. (2022) compared traditional ML algorithms with advanced DL techniques like LSTM and GRU for text categorization, highlighting the superior performance of LSTM, which achieved 92% accuracy. This underscores the adaptability of DL models to various datasets and applications.

Building on the strength of CNN architectures in text classification, the model integrates LRP for pruning, recursive data pruning techniques for reducing complexity, and cross-validation to ensure model robustness. This approach presents a significant advancement in handling complex text classification problems, particularly in datasets with high-dimensional features.

MATERIALS AND METHODS

In this section, the intricate process of building the model for the study is explored. Text classification, a complex task, relies on a variety of methods, each with its own nuances and considerations. The procedure adopted in this study is meticulously outlined in Figure 2, providing a visual representation of the methodology flow. Given the supervised nature of the machine learning algorithms employed, the cornerstone of the approach lies in the availability of labeled documents. These documents serve as the foundation upon which the classification system is built, providing the necessary groundwork for training and evaluation.

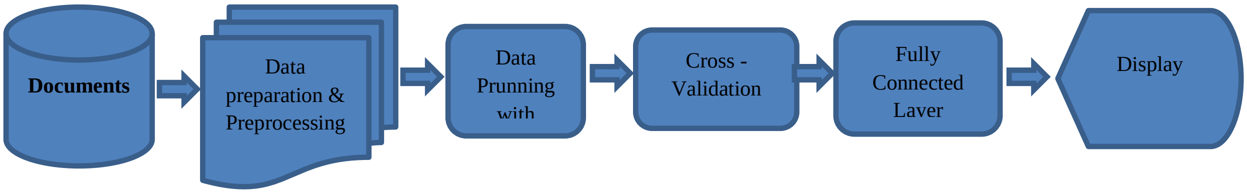

Figure 2: Methodology Flow

Data Preparation and Preprocessing

Text preparation and preprocessing are crucial steps in any natural language processing (NLP) task, including text classification. These steps involve transforming raw textual data into a clean and structured format suitable for analysis and modeling. In this proposal, we outline the importance of text preparation and preprocessing in the context of building a text classification model using modified recursive data pruning with LRP into convolutional neural networks (CNNs).

Text preparation begins with cleaning the raw text data by removing any irrelevant characters, symbols, or formatting, such as HTML tags, punctuation marks, and special characters. This ensures that the text is free from noise and inconsistencies that could interfere with the modeling process. Following cleaning, the text is tokenized, breaking it down into individual words, phrases, or tokens. This step helps in understanding the structure of the text and facilitates further analysis.

Lowercasing is another essential preprocessing step where all text is converted to lowercase to maintain consistency in word representations. By doing so, variations of the same word with different cases are treated as identical entities, preventing redundancy in the data. Additionally, stopwords, which are common words that do not carry significant meaning, such as articles, conjunctions, and prepositions, are removed to reduce noise and improve the efficiency of the model.

Lemmatization or stemming is employed to normalize words to their base or root form, reducing inflectional and derivational forms. This ensures that different variations of the same word are treated as a single entity, enhancing the model's understanding of the text. Furthermore, numerical data within the text is handled by converting it into a standard format or scale, such as normalization or standardization, to ensure uniformity and comparability with text data.

Missing values within the text data are addressed through imputation techniques or by removing rows with missing values, depending on the extent of missingness and its impact on the analysis. Categorical variables are encoded into numerical representations using techniques such as one-hot encoding or label encoding, enabling the model to process them effectively.

Text preparation and preprocessing are critical in ensuring that the textual data is clean, structured, and suitable for subsequent analysis and modeling tasks. By implementing these preprocessing steps, we aim to enhance the performance and efficiency of our text classification model, ultimately leading to more accurate and meaningful results.

Proposed Methodology

In the pursuit of advancing text classification models, an integrated approach leveraging Layer-wise Relevance Propagation (LRP), Recursive Data Pruning (RDP), and Convolutional Neural Networks (CNNs) with cross-validation emerges as a promising avenue. This methodology delineates the construction and functionality of this research model, emphasizing its layers, specifications, and parameter configurations.

Data Preprocessing

Model development begins with data preprocessing, a crucial step in preparing input text data for ingestion. Tokenization, vectorization, and TF-IDF weighting transform raw text into numerical representations. LRP is seamlessly integrated into this pipeline, attributing relevance scores to individual features, enhancing interpretability.

Pseudocode for Preprocessing with LRP

```
# Tokenization and Vectorization
vectorizer = TfidfVectorizer()
X = vectorizer.fit_transform(corpus)
```

Layer-wise Relevance Propagation (LRP) Integration

The LRP technique is employed to enhance feature selection by assigning relevance scores to different features, allowing for an interpretable classification process. The relevance score of neuron in the previous layer can be computed by propagating the relevance scores from the output to the input layer. The equation for Layer-wise Relevance Propagation (LRP) is given by:

the relevance score $\mathbb{R}_j$ of neuron j in the previous layer can be computed by propagating the relevance scores from the output to the input layer. The equation for Layer-wise Relevance Propagation (LRP) is given by:

$$\mathbb{R}_j = \sum_i \sum_j \frac{w_{ij.R_i}}{w_{ij..x_j}}. \tag{1}$$

Where,

$\mathbb{R}_j$ as the relevance score of neuron *i* in the output layer.

w_{ij} as the weight connecting neuron *i* in the output layer to neuron j in the previous layer.

$x_j$as the activation of neuron j in the previous layer.

Then, the relevance score $\mathbb{R}_j$ of neuron j in the previous layer can be computed as:

This equation represents the relevance propagation from the output layer to the previous layer, taking into account the weights connecting neurons in the output layer to neurons in the previous layer, as well as the activations of neurons in the previous layer. By applying LRP, the relevance scores can be propagated back through the network, thereby enabling targeted pruning of less important features and improving the model's efficiency.

```
# Applying LRP to compute relevance scores
def compute_lrp_scores(model, X):
    relevance_scores = model.get_relevance(X)
    return relevance_scores
```

Recursive Data Pruning (RDP)

RDP enhances model efficiency by identifying and pruning less relevant features from the dataset. LRP guides the pruning process by attributing relevance scores to words or features, ensuring only the most salient ones are retained, optimizing performance while reducing complexity.

Pseudocode for RDP

```
def recursive_pruning(features, relevance_scores, threshold=0.2):
    pruned_features = [f for f, r in zip(features, relevance_scores) if
            r > threshold]
    return pruned_features

X_pruned = recursive_pruning(X, compute_lrp_scores(model))
```

CNN Architecture

With the preprocessed and pruned data in hand, attention turns to constructing the CNN architecture. The model comprises several layers, each serving a specific function in classification:

Input Layer

Receives the preprocessed data as input.

```
inputs = Input(shape=(max_length,))
```

Embedding Layer

Converts tokenized words into dense vector representations.

```
embedding = Embedding(input_dim=vocab_size, output_dim=128, input_length=max_length)(inputs)
```

Convolutional Layers

Extract hierarchical features from input data using multiple filter sizes.

```
conv1 = Conv1D(filters=64, kernel_size=3, activation='relu', padding='same')(embedding)
conv2 = Conv1D(filters=128, kernel_size=5, activation='relu', padding='same')(conv1)
```

Pooling Layers

Reduce feature dimensionality while retaining critical information.

```
pool = GlobalMaxPooling1D()(conv2)
```

Dense Layers (Integrated with LRP)

Perform nonlinear transformations on extracted features, learning complex patterns and relationships.

```
dense1 = Dense(128, activation='relu')(pool)
dropout = Dropout(0.3)(dense1)
outputs = Dense(num_classes, activation='softmax')(dropout)
```

Model Compilation and Training

The model training process incorporates cross-validation, a robust technique for assessing performance and generalization. The dataset is split into training and validation sets, with CNN trained on the training data and evaluated on the validation set.

```
model = Model(inputs=inputs, outputs=outputs)
model.compile(optimizer='adam',
loss='categorical_crossentropy', metrics=['accuracy'])
model.fit(X_train, y_train, validation_data=(X_val, y_val), epochs=10, batch_size=32)
```

Model Evaluation

After training, evaluation on a separate test dataset assesses performance metrics: accuracy, precision, recall, and F1-score. The confusion matrix provides a visual representation of predictions compared to actual classes.

y_pred = model.predict(X_test)
conf_matrix = confusion_matrix(y_test, np.argmax(y_pred, axis=1))

The integrated model, combining LRP-integrated recursive data pruning, CNN architecture with specified layers, and cross-validation, offers a powerful approach to text classification. By leveraging LRP for interpretability, RDP for efficiency, and CNNs for feature extraction, the model achieves high performance and generalization across diverse text classification tasks and datasets.

Dataset Source

The datasource comprises a vast collection of over 1 million news articles sourced from an extensive network of 2000+ news outlets over a span of more than one year. This comprehensive dataset has been meticulously curated by *ComeToMyHead, (http://www.di.unipi.it/~gulli/AG_corpus_of_news_articles.html)* an academic news search engine operating since July 2004. It serves as a valuable resource for the academic community, facilitating research in various domains such as data mining (clustering, classification, etc.), information retrieval (ranking, search, etc.), XML, data compression, data streaming, and other non-commercial activities. The dataset is generously provided by the academic community for research purposes, enabling scholars to explore and innovate in diverse fields leveraging real-world data.

The AG's news topic classification dataset is meticulously constructed, selecting the four largest classes from the original corpus to ensure representativeness and diversity. Each class encompasses 30,000 training samples and 1,900 testing samples, yielding a total of 120,000 training samples and 7,600 testing samples.

Evaluation Matrix

The experimental comparison of classification algorithms will be using confusion matrix. A confusion matrix is a table that is often used to describe the performance of a classification model on a set of test data for which the true values are known. In the context of this research, tt provides valuable insights into an algorithm's performance, allowing for assessment of its ability to accurately classify transactions as fraudulent or non-fraudulent. In the confusion matrix, the rows represent the actual classes, and the columns represent the predicted classes. Table 1 shows the confusion matrix for a two-class classifier (Amin and Mahmoud, 2022).

Table 1: Confusion Matrix for two class classifiers

		PREDICTED	
ACTUAL		Positive	Negative
	Positive	A (TP)	B (FN)
	Negative	C (FP)	D (TN)

TP = True Positive, FP = False Positive, TN = True Negative, FN = False Negative

After the confusion matrix is generated for each of the implemented algorithm, the Accuracy, Sensitivity, Specificity Recall and Error rate values are calculated from the confusion matrix as follows;

Accuracy: It is the percentage of accurate predictions, that is, the ratio of number of correctly classified instances to the total number of instances and it can be defined as:

$$\text{Accuracy} = \frac{\text{TP + TN}}{\text{TP + FN + FP + TN}} \qquad (2)$$

Precision: Precision is the ratio of positively predicted instances among the retrieved instances

$$\text{Precision} = \frac{\text{TP}}{\text{TP +FP}} \qquad (3)$$

False Positive rate (FPR). This measures the rate of wrongly classified instances. A low FP-rate signifies that the classifier is a good one.

$$\text{FPR} = \frac{\text{FP}}{\text{FP + TN}} \qquad (4)$$

True Positive Rate: It is the proportion of positives that are correctly identified

$$TPR = \frac{TP}{TP + FN} \qquad (5)$$

Specificity: It is the proportion of negatives that are correctly identified. It is calculated as the number of correct negative predictions divided by the total number of negatives. It is also called true negative rate. The worst is 0.0 while the best is 1.0.

$$Specificity = \frac{TN}{TN + FP} \qquad (6)$$

Recall: It is the ratio of positively predicted instances among all the instances

$$\text{Recall} = \frac{\text{TP}}{\text{TP + FN}} \qquad (7)$$

Kappa Score: It is a measure of agreement between the predicted and actual classes, taking into account the agreement that could occur by chance alone.

Receiver Operating Characteristic (ROC) curve. The true positive rate is constructed against the false positive rate, that is, a plot of False Positive Rate vs True Positive Rate.

RESULTS AND DISCUSSION

The experimental results provide insights into the model's performance on the AGFNews datasets, highlighting its strengths in achieving high classification accuracy, precision, recall, and F1-score, while simultaneously reducing computational complexity. The results are organized to provide a clear understanding of the model's performance across various metrics. Table 2 offers a detailed classification report, including metrics such as accuracy, precision, recall, and F1-score, which collectively reflect the effectiveness of the model in distinguishing between different classes. In addition to the tabular data, Figure 3 visualizes the confusion matrix, offering a graphical representation of the model's predictions versus the actual class labels. This visualization aids in identifying specific areas where the model excels or encounters challenges, thereby providing deeper insights into its strengths and potential areas for improvement.

Table 2: Classification Report for the text classification

Class	Precision	Recall	F1-Score	Support
World	1.00	0.94	0.97	5956
Sports	0.96	0.98	0.97	6058
Business	0.93	0.92	0.92	5911
Science	0.90	0.90	0.90	6075
Accuracy			0.94	24000
Macro Avg	0.95	0.93	0.94	24000
Weighted Avg	0.95	0.94	0.94	24000

The classification report presents the performance metrics of the developed model across four distinct classes: World, Sports, Business, and Science as shown in Table 2. These metrics include precision, recall, F1-score, and support, each providing valuable insights into the model's effectiveness in classifying text data.

For the "World" class, the model achieved a precision of 1.00, indicating a high accuracy in predicting this class when it was indeed present. The recall for this class was 0.94, signifying the model's effectiveness in identifying the "World" class when it was present in the dataset. The F1-score, a harmonic mean of precision and recall, stood at 0.97, reflecting a strong balance between precision and recall. The support, representing the number of actual instances of the "World" class in the dataset, was 5956.

The "Sports" class exhibited the highest performance among all classes, with a precision of 0.96, indicating excellent accuracy in prediction. The recall was even higher at 0.98, showing that the model almost perfectly identified the "Sports" class when present. The F1-score of 0.97 further underscores the model's exceptional performance in this category. The support for the "Sports" class was 6058.

In the "Business" class, the model achieved a precision of 0.93 and the recall of 0.92, resulting in an F1-score of 0.92. This consistency in precision and recall suggests that the model performed well in both identifying and correctly predicting instances of the "Business" class. The support for this class was 5911.

The "Science" class had a precision and recall of 0.90, with an F1-score of 0.90. This indicates that the model maintained a consistent level of performance in classifying instances of "Science," with a support of 6075.

The overall accuracy of the model across all classes was 0.94, demonstrating its ability to correctly classify the majority of instances in the dataset. The macro average, which considers the average performance of the model across all classes, was 0.95 for precision, 0.93 for recall, and 0.94 for F1-score, indicating that the model performed well across the different categories. The weighted average, which accounts for the support of each class, also yielded a precision of 0.95, 0.94 for recall, and F1-score of 0.94, further confirming the robustness and reliability of the model's predictions across the entire dataset of 24000 instances.

The analysis of the classification report highlights the model's strong performance in text classification tasks, particularly in the "Sports" and "World" classes, while maintaining consistent accuracy and reliability across all categories. This comprehensive evaluation underscores the model's effectiveness in handling diverse text data with varying levels of complexity.

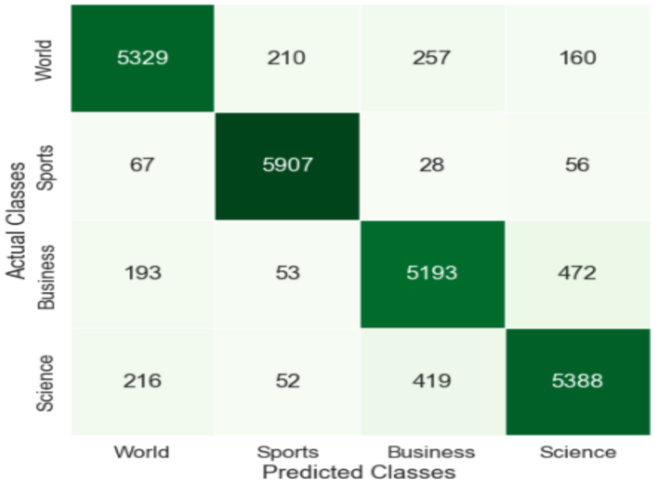

Figure 3: Confusion Matrix

The confusion matrix obtained from the developed model provides a detailed breakdown of its classification performance across four classes: World, Sports, Business, and Science as visualized in Figure 3. This matrix is crucial in understanding how well the model distinguishes between different categories, as it reveals not only correct predictions but also the nature of misclassifications.

For the "World" class, the model correctly identified 5329 instances out of a total of 5956. However, 210 instances of "World" were incorrectly classified as "Sports," 257 as "Business," and 160 as "Science." These misclassifications suggest that while the model performs well overall, there are cases where it confuses the "World" class with other categories, particularly with "Business," possibly due to overlapping content or context within the text data.

In the case of the "Sports" class, the model shows strong performance, correctly predicting 5907 out of 6058 instances. Only 67 instances were misclassified as "World," 28 as

"Business," and 56 as "Science." The low number of misclassifications indicates that the model is highly effective in identifying "Sports" content, with minimal confusion between this class and the others.

The "Business" class presents a slightly different scenario, with 5193 correct predictions out of 5911 instances. However, 193 instances were incorrectly classified as "World," 53 as "Sports," and 472 as "Science." The relatively higher number of instances misclassified as "Science" suggests a closer semantic or contextual similarity between the "Business" and "Science" classes, leading to more frequent confusion by the model.

For the "Science" class, the model correctly classified 5388 out of 6075 instances. Misclassifications include 216 instances incorrectly labeled as "World," 52 as "Sports," and 419 as "Business." Similar to the "Business" class, there appears to be a significant overlap in features between "Science" and "Business," resulting in a notable number of misclassifications between these two categories.

The confusion matrix overall highlights the model's strong performance in distinguishing between "Sports" and the other classes, with fewer errors compared to other categories. However, it also reveals areas where the model struggles, particularly in differentiating between "Business" and "Science," as well as between "World" and "Business." These patterns of misclassification provide valuable insights into the model's strengths and limitations, suggesting that while the model is generally effective, there is room for improvement in refining the distinctions between certain classes, especially where semantic overlap exists.

Comparison with the Benchmark

This section compares the results of the developed model with that of the benchmark models to evaluate its relative performance and determine its superiority. Table 3 shows the detailed result comparison.

Table 3: Results comparison with the Benchmark

S/N	Author	Methodology	Accuracy (%)
1	Li et al.,(2020)	ReDP-CNN	92.88
2	Developed Model	ReLRP-CNN	94

The comparison between the developed model and benchmark models is presented in Table 3, highlighting the accuracy percentages of each approach. The benchmark model, as proposed by Li et al. (2020), employs the Recursive Data Pruning Convolutional Neural Network (ReDP-CNN) methodology, which achieved an accuracy of 92.88%.

In contrast, the developed model, utilizing the integrated approach of Layer-wise Relevance Propagation (LRP) and Convolutional Neural Networks (ReLRP-CNN), attained a higher accuracy of 94%. This improvement signifies an enhancement in performance, indicating that the incorporation of LRP into the CNN framework contributes to a more effective classification model.

The increase in accuracy reflects the benefits of integrating LRP with recursive data pruning techniques. By leveraging LRP, the model can better identify and retain the most relevant features, thereby improving its ability to accurately classify text data. The developed model's superior performance over the benchmark demonstrates its potential for more precise and robust text classification.

CONCLUSION

The study successfully advanced text classification models by integrating Layer-wise Relevance Propagation (LRP), recursive data pruning, and Convolutional Neural Networks (CNNs) with cross-validation. This novel approach addressed key limitations of existing methods, such as information loss and overfitting, and demonstrated substantial improvements in model performance. The use of LRP facilitated precise identification of relevant features, while recursive data pruning optimized model efficiency. Cross-validation further ensured robust performance evaluation.

In comparison to the benchmark model proposed by Li et al. (2020), which achieved an accuracy of 92.88% using the Recursive Data Pruning Convolutional Neural Network (ReDP-CNN), the developed ReLRP-CNN model achieved a higher accuracy of 94%. This improvement underscores the superiority of the developed model over the benchmark, highlighting the effectiveness of integrating LRP and CNN with pruning techniques. This enhancement in both accuracy and efficiency offers valuable insights for future research and applications in natural language processing.

RECOMMENDATION

To advance the field of text classification further, several areas warrant additional investigation and refinement. Building upon the successes of the current approach, future research should explore the integration of more sophisticated techniques to address the evolving challenges in natural language processing. One area of potential development involves enhancing the recursive data pruning methodology. While the current study demonstrated improvements in efficiency and model performance, exploring alternative pruning strategies or hybrid approaches could yield even more effective results. Investigating different criteria for feature selection and pruning, along with adaptive methods that respond dynamically to data characteristics, may help to refine the model's ability to retain critical information while reducing complexity.

Additionally, the application of LRP could be expanded to more complex neural network architectures beyond CNNs. Incorporating LRP into other types of deep learning models, such as Transformers or hybrid architectures, might reveal new insights into feature relevance and model interpretability. This exploration could offer broader applicability and enhanced performance across various text classification tasks.

REFERENCES

Abbasi, A., Chakraborty, C., Nebhen, J., Zehra, W., and Jalil, Z. (2021). Elstream: an ensemble learning approach for concept drift detection in dynamic social big data stream learning. IEEE Access 9, 66408–66419. https://doi.org/10.1109/ACCESS.2021.3076264

Abdullah Alqahtani Habib Ullah Khan, Shtwai Alsubai1, Mohemmed Sha, Ahmad Almadhor ,Tayyab Iqbal and Sidra Abbas(2022) An efficient approach for textual data classification using deep learning

Amin, F., & Mahmoud, M. (2022). Confusion matrix in binary classification problems: A step-by-step tutorial. *Journal of Engineering Research*, *6*(5), 0-0.

Bashir, M. F., Javed, A. R., Arshad, M. U., Gadekallu, T. R., Shahzad, W., and Beg, M. O. (2022). “Context aware emotion detection from low resource urdu language using deep neural network,” in Transactions on Asian and Low-Resource Language Information Processing.

Çoban, Ö.; Özel, S.A.; ˙Inan, A. Deep learning-based sentiment analysis of Facebook data: The case of Turkish users. Comput. J. 2021, 64, 473–499.

Dogru, H.B.; Tilki, S.; Jamil, A.; Hameed, A.A. Deep learning-based classification of news texts using doc2vec model. In Proceedings of the 1st International Conference on Artificial Intelligence and Data Analytics (CAIDA), Riyadh, Saudi Arabia, 6–7 April 2021; IEEE: Riyadh, Saudi Arabia, 2021; pp. 91–96.

Hassan, S. U., Ahamed, J., & Ahmad, K. (2022). Analytics of machine learning-based algorithms for text classification. *Sustainable Operations and Computers*, *3*, 238-248.

Hartmann, J.; Huppertz, J.; Schamp, C.; Heitmann, M. Comparing automated text classification methods. Int. J. Res. Mark. 2019, 36, 20–38.

Hina, M., Ali, M., Javed, A. R., Srivastava, G., Gadekallu, T. R., and Jalil, Z. (2021b). “Email classification and forensics analysis using machine learning,” in 2021 IEEE SmartWorld, Ubiquitous Intelligence and Computing, Advanced Trusted Computing, Scalable Computing and Communications, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/IOP/SCI) (Atlanta, GA: IEEE), 630–635.

Kim, H., & Jeong, Y. S. (2019). Sentiment classification using convolutional neural networks. *Applied Sciences*, *9*(11), 2347..

Kohlbrenner, M., Bauer, A., Nakajima, S., Binder, A., Samek, W., & Lapuschkin, S. (2020, July). Towards best practice in explaining neural network decisions with LRP. In *2020 International Joint Conference on Neural Networks (IJCNN)* (pp. 1-7). IEEE.

Kuyumcu, B.; Aksakalli, C.; Delil, S. An automated new approach in fast text classification (fastText): A case study for Turkish text classification without pre-processing. In Proceedings of the 3rd International Conference on Natural Language Processing and Information Retrieval, ACM, Tokushima, Japan, 28–30 June 2019; pp. 1–4

Lapuschkin, S. (2019). *Opening the machine learning black box with layer-wise relevance propagation* (Doctoral dissertation, Dissertation, Berlin, Technische Universität Berlin, 2018).*Applied Sciences*, *9*(11), 2347.

Li, Q.; Peng, H.; Li, J.; Xia, C.; Yang, R.; Sun, L.; Yu, P.S.; He, L. A survey on text classification: From traditional to deep learning. ACM Trans. Intell. Syst. Technol. (TIST) 2022, 13, 1–41.

Li, Q., Li, P., Mao, K., & Lo, E. Y. M. (2020). Improving convolutional neural network for text classification by recursive data pruning. *Neurocomputing*, *414*, 143-152.

Macukow, B. (2016). Neural networks–state of art, brief history, basic models and architecture. In *Computer Information Systems and Industrial Management: 15th IFIP TC8 International Conference, CISIM 2016, Vilnius, Lithuania, September 14-16, 2016, Proceedings 15* (pp. 3-14). Springer International Publishing.

O'Shea, K., & Nash, R. (2015). An Introduction to Convolutional Neural Networks. ArXiv, abs/1511.08458.

Romero, M., Guédria, W., Panetto, H., & Barafort, B. (2022). A hybrid deep learning and ontology-driven approach to perform business process capability assessment. *Journal of Industrial Information Integration*, *30*, 100409.

Salter, C. (2020). Neuronal acts. *Performance Research*, *25*(3), 104-113.

Toofani, A., Singh, L., & Paul, S. (2024). From interpretation to explanation: An analytical examination of deep neural network with linguistic rule-based model. *Computers and Electrical Engineering*, *117*, 109258

Uysal, A.K.; Gunal, S. The impact of preprocessing on text classification. Inf. Process. Manag. 2014, 50, 104–112

Vukadin, D., Afrić, P., Šilić, M., & Delač, G. (2024). Advancing Attribution-Based Neural Network Explainability through Relative Absolute Magnitude Layer-Wise Relevance Propagation and Multi-Component Evaluation. *ACM Transactions on Intelligent Systems and Technology*.

Xiao, Y., Duan, Z., & Lei, P. (2024). Explaining Multiple Types of Crash Injury Severity Predictions with Layer-wise Relevance Propagation in Multi-task Deep Neural Networks.

Yamashita, R., Nishio, M., Do, R. K. G., & Togashi, K. (2018). Convolutional neural networks: an overview and application in radiology. Insights into imaging, 9(4), 611-629

Yıldırım, S.; Yıldız, T. A comparative analysis of text classification for Turkish language. Pamukkale Univ. J. Eng. Sci. 2018, 24, 879–886

Zang, B., Ding, L., Feng, Z., Zhu, M., Lei, T., Xing, M., & Zhou, X. (2021). CNN-LRP: Understanding convolutional neural networks performance for target recognition in SAR images. *Sensors*, *21*(13), 4536.

Zulqarnain, M.; Alsaedi, A.K.Z.; Ghazali, R.; Ghouse, M.G.; Sharif, W.; Husaini, N.A. A comparative analysis on question classification task based on deep learning approaches. PeerJ Comput. Sci. 2021, 7, e570.