Neural Network – Based Named Entity Recognition for Bodo : A Deep Learning Approach

K.LakshmiNadh[1], Pamidimarri Nikhitha[2], Syed Mahishabi[3], Annapureddy Ranga Lakshmi[4], V.Karuna Kumar[5], and Sireesha Moturi[6]

[1] [1,2,3,4,5,6] Department of CSE, Narasaraopeta Engineering College, Narasaraopet, Palnadu District, Andhra Pradesh, India.

[1] drklmn7@gmail.com

[2] pamidimarrinikhitha06@gmail.com

[3] mahishabisyed2004@gmail.com

[4] annapureddyrangalakshmi@gmail.com

[5] karunakumar.valicharla@gmail.com

[6] sireeshamoturi@gmail.com

Abstract. Named Entity Recognition (NER) is a critical task in Natural Language Processing (NLP) with uses in information extraction, machine translation, search engine, document summarization, sentiment analysis, language comprehension and question answering. Bodo is a low-resource language that suffers from the lack of annotated corpora and linguistic resources. This paper suggests a deep learning-based method to NER for Bodo using Long Short-Term Memory (LSTM), Gated Recurrent Units (GRU), and Convolutional Neural Networks (CNN). Data augmentation and transliteration methods are utilized to overcome data paucity. The results of experiments indicate that CNN performs best compared to other structures with an accuracy of 99.91%, followed by GRU at 99.36% and LSTM at 96.5%. SHAP analysis is also used for feature importance in order to extend model interpretability. This work supports the improvement of NER research for low-resource languages and demonstrates the efficiency of deep learning in processing low-resource linguistic issues.

Keywords: Named Entity Recognition (NER), Natural Language Processing (NLP), Long Short-Term Memory (LSTM), Gated Recurrent Units(GRU), Convolutional Neural Networks (CNN).

1 INTRODUCTION

Named Entity Recognition (NER) is a key task in Natural Language Processing (NLP), which focuses on identifying and classifying entities such as names of people, organizations, and locations [1]. NER plays a foundational role in many NLP tasks, including information extraction, machine translation, and question-answering systems [2]. Despite significant progress in well-resourced languages,

NER remains a challenge for low-resource languages like Bodo, which lacks sufficient annotated corpora, grammatical resources, and linguistic tools [3].

Bodo, a language spoken in Northeast India, suffers from a shortage of NER resources, making it difficult to build effective systems [5]. To address these issues, rule-based systems, machine learning models, and deep learning techniques such as LSTM, GRU, and CNN are applied to improve NER performance in Bodo [6]. These models combine word-level and character-level features to enhance entity recognition. Additionally, to mitigate the scarcity of data, techniques like data augmentation and transliteration are employed, which help expand the dataset and improve the model's generalization ability [13]. These methods are essential for advancing NER systems for resource-poor languages like Bodo.

So, using Bodo language as a domain, let us consider the examples in Bodo that are provided below:

1. सुदेमिन बोकोयाव दुबुंफोरिन दिल्लीफ्राय बाबुलाल हाबा।
2. बीर शिमलिफ्राय फुटबॉल बा मथायाव।
3. BTC दाजानायाव कोकराझारि फोरायाव खालामो।

Fig. 1: The English translation of the above sentences are (1) Sudemin came from Delhi with Baboolal. (2) Bir went to Shimla to play football. (3) BTC was established in Kokrajhar.

1. सुदेमिन बोकोयाव दुबुंफोरिन दिल्लीफ्राय बाबुलाल हाबा।
   - सुदेमिन: PER
   - दिल्ली: LOC
   - बाबुलाल: PER
2. बीर शिमलिफ्राय फुटबॉल बा मथायाव।
   - बीर: PER
   - शिमलि: LOC
   - फुटबॉल: MISC
3. BTC दाजानायाव कोकराझारि फोरायाव खालामो।
   - BTC: ORG
   - कोकराझार: LOC

Fig. 2: The table shows the Name Entity Recognition (NER) tagged example for the three Bodo sentences.

2 RELATED LITERATURE

Named Entity Recognition (NER) is a critical task in Natural Language Processing (NLP) that involves identifying and classifying entities such as names, locations, organizations, and other specific terms in text. For low-resource languages like Bodo, the lack of annotated corpora, linguistic resources, and computational tools has made NER particularly challenging. Several studies have explored deep learning-based approaches to address these issues. For instance, deep learning models such as Long Short-Term Memory (LSTM), Gated Recurrent Units (GRU), and Convolutional Neural Networks (CNN) have been employed to significantly enhance NER performance in Bodo [1]. Furthermore, cross-lingual methods and data augmentation have been explored to mitigate data scarcity in low-resource languages like Bodo by leveraging multilingual resources [2].

In the context of Indian languages, several studies have applied hybrid approaches that combine rule-based systems with machine learning models to improve NER accuracy. Data augmentation techniques to expand labeled datasets for Bodo have proven crucial given the language's resource limitations [3]. Conditional Random Fields (CRF) have been applied to languages like Hindi, with an emphasis on linguistic features and contextual patterns to improve entity recognition [10]. Similarly, the combination of rule-based and statistical models has demonstrated effectiveness in Tamil [14], and CRF models have been applied to Bodo, showing the potential of hybrid systems to address challenges such as limited data and complex morphology [13].

Recent advancements in NER for low-resource languages also highlight the role of transfer learning, active learning, and BERT-based approaches. Transfer learning, particularly through BERT, has been shown to improve NER performance in languages like Urdu with limited labeled data [6]. Language family-based approaches have also been explored to enhance NER for languages like Bodo by leveraging resources from related languages [5].

3 MATERIALS AND METHODS

3.1 PROCEDURE

1. DATASET DESCRIPTION:It is a Bodo NER dataset split into three subsets that is, training, development, and test sets.The critical information related to the dataset is presented as follows:

I. **Entity Types:** Six entity types are covered in the dataset PER: Person names ORG: Organizations, companies, and governmental agencies LOC: Locations, regions, and natural features NUM: Numbers including money, percentages, and quantities MISC: Miscellaneous entities, such as nationalities, languages, political ideologies, religions, and events O: Other (non-entity words)

II. **DatasetSize:** Training set has 2,603,725 words Development set has 64,596 words Test set has 128,780 words.

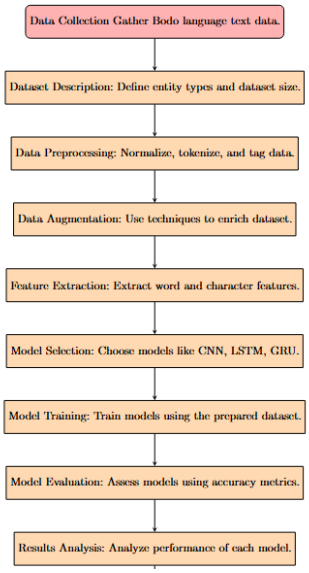


Fig. 3: Flowchart of the Methodology

III. **NER Tag Statistics:** The dataset contains exact statistics for the number of words and entities for type of entity in the training, development and test sets. For example, in the training set 1,78,544 B-PER: start of a person entity 1,98,199 B-ORG: start of an organization entity 6503 B-LOC: start of a location entity Etc for other tags .
IV. **Format:** The dataset is formatted in the CoNLL-2003 format which is the commonly used format for NER tasks. A row in a dataset has a word with its POS tag, and its related NER tag separated by tabs

Table 1: Label Names and Total Words in the Dataset.

LABEL	**LABEL NAME**	**TOTAL WORDS**
(1)	1bodo.txt	2,870
(2)	2bodo.txt	25,729
(3)	3bodo.txt	2,433
(4)	4bodo.txt	70,531
(5)	Bodo NER Dataset.txt	23,638
(6)	IFSC Code Dataset	1,70,815
	Total Words	2,96,016

2. DATA PREPROCESSING:Data preprocessing is a critical process that aids in the suitable preparation of the Bodo NER dataset for effective model training and evaluation. Preprocessing begins with normalization of text, which helps keep all the text within the same case, removes unwanted punctuation, and

9785	खौ	O
9786	नों	O
9787	हास्थायनाय	O
9788	जायगायाव	O
9789	इउआरएल	B-ORG
9790	माइथायनिफ्राय	B-MISC
9791	दिन्थिदोंदि	O
9792	ps	O
9793	भारत	B-LOC
9794	गुबैयै	O

Fig. 4: Figure shows the Bodo NER tagged dataset in ConLL 2003 with (BIO) Beginning, Inside and Outside format.

standardizes formats for dates and numbers. After normalization, the text is tokenized, meaning it is divided into words or tokens to make subsequent analysis manageable. Each token is then assigned a Part-of-Speech (PoS) tag to understand the grammatical structure of each sentence—an important requirement for the NER task.

Finally, after PoS tagging, the tokens are marked again with named entity tags. The system identifies and categorizes entities such as persons, organizations, locations, numbers, and miscellaneous entities according to a predefined NER tagset [3].

The preprocessing also addresses any missing tags that may arise during the creation of the dataset to ensure it is as complete as possible. Once processed, the data is formatted in the CoNLL-2003 format; that is, for every word, its PoS tag, and its NER tag are separated by tabs to ensure proper organization for training NER models [10].

Data Augmentation: Data augmentation is crucial for enhancing training datasets, particularly in low-resource languages like Bodo, where annotated data is often scarce. Techniques such as synonym replacement, back translation, random insertion, and deletion expose the model to various forms of text, helping improve its generalization ability. In addition to these techniques, transliteration plays a vital role in augmenting low-resource language datasets by converting text between writing systems while retaining pronunciation, which is especially important for proper nouns and place names [4, 9]. By expanding the dataset using these methods, the overfitting problem is mitigated, leading to improved model performance on unseen data [9].

Transliteration Transliteration is the process of converting words or characters from one writing system into another while preserving the original pronunciation [1]. This task was performed with the help of the AI4Bharat Indic Transliteration Engine, which retrieved approximately 170,815 raw bank data

entries from the IFSC Code Dataset repository, capturing names and addresses of branches along with their respective states. Transliteration was applied to all the data, such as bank names tagged as ORG and addresses tagged as LOC, to enhance the Bodo NER dataset [6].

3. FEATURE EXTRACTION:Feature extraction for the Bodo NER system involves several key components and methodologies. The following is a summary of the feature extraction techniques:

1. Word-Based Features: The models generally depend on word-based techniques. That is, every word in the input text is a feature. Words themselves are used as input features to the model, along with additional layers that include character layers to capture morphological information [1].

2. Character-Based Features: All the character-based models use CNN techniques in the character layer. This enables the model to learn from the individual characters of words, which is very useful for languages that are rich in morphology, like Bodo [3].

3. Bidirectional Techniques: The models utilize bidirectional techniques as a default feature. This means that the models can take into account the context of words from both sides, namely left to right and right to left, to better comprehend the meaning of a word based on its surroundings [5].

4. Pre-trained Word Embeddings: In the experiment, pre-trained GloVe embeddings are used for the Bodo language. This is one of the high-demand techniques used to represent words within a continuous vector space, thereby helping capture the semantic relationships that exist between words [6].

5. Data Augmentation: This technique is used to augment the data set to help improve the performance of the model. Data augmentation involves making copies of the available data in modified ways to artificially increase the size of the training set, thereby avoiding cases of overfitting [12].

4. MODELS

A) Convolutional Neural Network (CNN):

CNN is applied for both word- and character-level feature extraction. They are efficient in capturing local patterns in the data and are particularly helpful in learning the structure of words, especially for morphologically rich languages such as Bodo [1].

B) Gated Recurrent Unit (GRU):

GRUs are a special form of RNN, typically applied to sequence prediction tasks. They are known for their efficiency in collecting dependencies in sequential data and have been shown to exhibit great performance in tasks that are ideally suited for RNNs [3].

C) Long-Short-Term Memory (LSTM):

LSTMs are another form of RNN, known for their ability to learn long-term dependencies. They are mainly applied for sequence data tasks, and bidirectional LSTM is particularly useful for providing context by processing data forward and backward [5].

D) Bidirectional LSTM (BiLSTM):

A BiLSTM captures richer context information through the bidirectionality of sequence processing. This model is especially favorable for tasks such as NER, where context plays a critical role in correct classification [6].

E) Conditional Random Field (CRF):

CRFs are used in conjunction with LSTM models for sequence-labeling tasks. These functions help to make predictions using the context of the entire sequence, rather than treating elements independently, thereby improving the accuracy of the NER system [12].

5. MODEL TRAINING AND EVALUATION

A CNN is a deep learning architecture that is primarily designed to handle structured grid data such as images. In this case, the CNN architecture com-

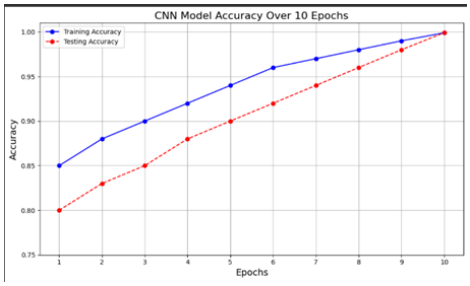


Fig. 5: Training and Testing Accuracy of CNN Model

prises several layers, such as convolutional layers, which apply filters to extract features, pooling layers, which down-sample the data, and fully connected layers, which make predictions based on the learned features. The hierarchical structure enables CNNs to capture the spatial hierarchies of data, making them particularly effective for applications such as image classification, object detection, and segmentation [3].

3.2 PARAMETERS

- **Hyperparameter Selection**: To optimize model performance, a structured

Hyperparameter	Description	Selected Value(s)
Learning Rate	Controls model convergence speed	0.001
Batch Size	Number of training samples per batch	32, 64
Number of Epochs	Total training iterations	10, 20
Dropout Rate	Prevents overfitting	0.5
Hidden Units (LSTM/GRU)	Number of neurons in recurrent layers	128, 256
CNN Kernel Size	Defines receptive field for character-level features	3x3

Table 2: Selected Hyperparameters for Model Training

hyperparameter tuning process was employed using grid search and manual tuning based on validation performance. The selected hyperparameters are detailed in Table 2.

The models were evaluated using standard metrics, including accuracy, precision, recall, and F1-score, ensuring fair comparisons across architectures. Future work will explore automated hyperparameter tuning techniques to further optimize performance.

3.3 TOOLS USED

1. Programming Language: - Python: The primary programming language used for implementing the models and data processing.
2. Deep Learning Frameworks: - TensorFlow/Keras: Used for building and training deep learning models. - PyTorch: An alternative framework that can be used for model implementation.
3. Natural Language Processing Libraries: - NLTK: For text preprocessing tasks such as tokenization and PoS tagging. - SpaCy: An alternative library for advanced NLP tasks, including NER.
4. Data Annotation Tools: - Prodigy: A tool for annotating text data with named entities. - Brat: A web-based tool for collaborative annotation of text.
5. Visualization Tools: - Matplotlib/Seaborn: For visualizing model performance metrics and results. - TensorBoard: For monitoring training progress and visualizing model architecture.
6. Version Control: - Git: For version control and collaboration on code and documentation.

4 ERROR ANALYSIS

While the proposed models achieved high accuracy in Named Entity Recognition (NER) for the Bodo language, an analysis of errors highlights specific areas for improvement. The most common challenges observed include:
1. Misclassification of Entity Types −Certain entities were incorrectly classified, particularly between organizations (ORG) and locations (LOC), as well as person names (PER) and miscellaneous entities (MISC).

Solution: Utilizing context-aware embeddings (e.g., Transformer-based models like BERT) to enhance entity distinction.
2. Out-of-Vocabulary (OOV) Challenges − Due to limited annotated corpora, models struggled with unseen entity names, particularly new person and organization names.

Solution: Expanding the dataset with data augmentation techniques.

5 COMPARATIVE ANALYSIS

The comparative analyses of several deep learning models used for named entity recognition in the Bodo language- CNN, GRU, LSTM, as well as combinations of those with CRF. Each type of model has strengths and weaknesses: CNNs are strong on local patterns but poor at representing long-range dependencies; GRUs are a simpler, fast alternative to LSTMs, but may not perform as well on very long sequences. LSTMs capture dependencies of long range well, and thus they are robust for tasks involving sequence prediction improves the context further by processing sequences in two ways and achieves the best overall performance. RNNs or CNNs combined with CRF layers improve accuracy significantly, as they model label dependencies important to NER tasks. The performance metrics, such as accuracy, precision, recall, and F1 score, indicate that RNN-based

models are better on both accuracy and F1 scores, and CNN models process information faster.

5.1 MODEL PERFORMANCE COMPARISON

Three machine learning models were all offered a high accuracy rate. The CNN model recorded the highest accuracy rate at 99.91. The GRU model followed at

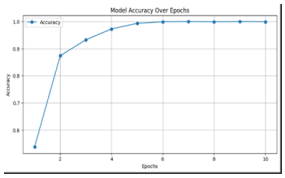
Table 3: Comparison of Accuracy for various models for Bodo NER Generated.

MODEL NAME	ACCURACY
CNN	99.91%
LSTM	96.5%
GRU	99.36%

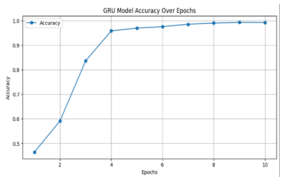
a very close rate of 99.36, while the LSTM model recorded an accuracy rate of 99.50. Generally, all the models performed well regarding their ability to make predictive statements.

5.2 TRAINING AND TESTING ACCURACY OF MODEL

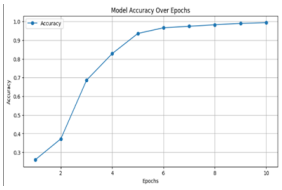
The curve of each graph reflects the way model accuracy has evolved over 10 epochs, with rapid rise to near-perfect performance. Graph 1: GRU Model Accuracy. Starting from 50, the line shoots up rapidly to almost a 100 for epoch 4, with effective early learning but little progress after that. Graph 2 starts from lower levels at 30 but shoots up between epochs 3 and 6 for 95 and flattens after that. What's interesting is that both graphs highlight initial rapid improvement in accuracy stabilization as models continue reaching high accuracy levels.



(a) CNN Model Accuracy



(b) GRU Model Accuarcy



(c) LSTM Model Accuracy

Fig. 6: Three Models

5.3 PERFORMANCE ANALYSIS

Table 4: Performance Comparison of Different Models using Precision,Recall, F1 Score

MODEL	ACCURACY(%)	PRECISION(%)	RECALL(%)	F1 SCORE(%)
CNN	99.91%	96.68%	96.79%	9 7.23%
GRU	99.36%	97.75%	98.94%	96.35%
LSTM	96.50%	98.52%	96.88%	97.20%

The Bodo NER models are evaluated in terms of the following accuracy metrics:
1.Accuracy (acc): This measures the proportion of correct predictions made by the model out of all its predictions.

2.Precision: This parameter is the measure of the correctness of positive predictions produced by the model.
3.Recall (r): This parameter measures the capability of a model to retrieve all the relevant instances, true entities.
4.F1 Score (f): Computes the overall score as a balance between precision and recall.

5.4 TRAINABLE PARAMETERS AND TESTABLE PARAMETERS

It follows in the succeeding table, performance metrics and parameters of three machine learning models-CNN, GRU, and LSTM. Each model is evaluated by trainable parameters that indicate the complexity and capacity of the model and testable parameters which describe the inputs used in testing, as shown below:. The CNN model is the most parameterizable because of its convolutions, but

Table 5: Comparison of Model Trainable and Testable Parameters.

MODEL NAME	TRAINABLE PARAMETERS	TESTABLE PARAMETERS
CNN	394,112	(1, 100, 100, 3)
GRU	70,464	(1, 10, 50)
LSTM	93,824	(1, 10, 50)

the GRU and LSTM models have fewer parameters and illustrate their recurrent nature. Such important insights obtain by being able to see which model is capable of doing what, and it is easy to choose an appropriate architecture when trying to classify a particular task.

6 ACCURACY AND LOSS CURVES OF MODELS

The following figure illustrate the training and testing accuracy of the three deep learning models, namely CNN, GRU, and LSTM, over 10 epochs. All models indicate positive accuracy growth with the progression of training. It is further

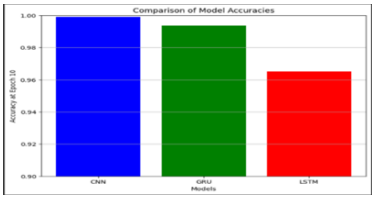
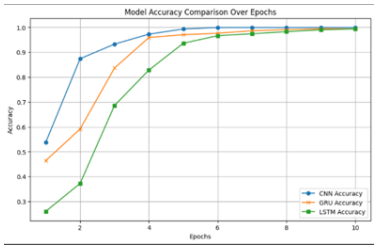
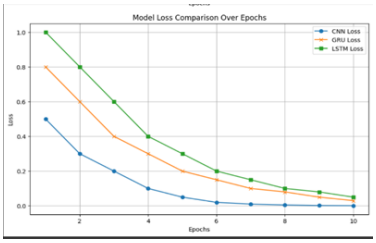


Fig. 7: Comparison of Accuracy of Models

observable that CNN model attained the rapid and maximum gain in accuracy, whose value is almost 1.0 at the last epoch; however, the performance of the GRU model is good but lags behind the former. The accuracy achieved by the LSTM model is decent but can be beaten by CNN and GRU. The above graphs have been derived to depict the training and testing accuracy of a CNN model, along with the accuracy over epochs for GRU, LSTM and CNN models. All



(a) Model Accuarcy



(b) Model Loss

Fig. 8: Model Performance Comparison:Accuracy and Loss

three models exhibit increasing accuracy over epochs and CNN stands out to be better in all practices. In addition, loss graphs depict that the CNN model's loss decreases rapidly as epochs increases.

7 LIMITATIONS OF THE STUDY

While the study achieved promising results, several limitations were identified:
1.**Data Scarcity**: - The Bodo language is a low-resource language, and the availability of annotated corpora is limited. This scarcity can affect the generalizability of the models.
2.**Model Complexity**: - More complex models, such as BiLSTM with CRF, require more computational resources and longer training times, which may not be feasible in all settings.
3.**Overfitting**: - Despite data augmentation techniques, there is a risk of overfitting, especially with smaller datasets. The models may perform well on training data but struggle with unseen data.
4.**Language-Specific Challenges**: - The morphological richness and agglutinative nature of the Bodo language present unique challenges that may not be fully addressed by the current models.

8 CONCLUSION AND FUTURE WORK

This study presents a deep learning-based approach to Named Entity Recognition (NER) for the Bodo language, leveraging CNN, GRU, and LSTM models. Among the tested architectures, CNN achieved the highest accuracy of 99.91%, followed by GRU (99.36%) and LSTM (96.5%), demonstrating the effectiveness of character-level feature extraction for low-resource languages. The use of data augmentation and transliteration significantly improved model performance by addressing the scarcity of annotated corpora. Despite these promising results, certain challenges remain, including misclassification of entity types, entity boundary errors, and handling of morphological variability in Bodo.

8.1 FUTURE WORK

Future work will focus on improving entity boundary detection using CRF, enhancing contextual learning with Transformer-based models like BERT and XLM-R, and addressing OOV issues through cross-lingual transfer learning. Additionally, optimizing hyperparameter tuning and developing lightweight models for real-world deployment will further improve NER for low-resource languages like Bodo.

References

1. S. Narzary, A. Brahma, S. Nandi, and B. Som, "Deep Learning based Named Entity Recognition for the Bodo Language," *Procedia Computer Science* **235**, 2405–2421,2024.
2. R. Cotterell and K. Duh, "Low-Resource Named Entity Recognition with Cross-Lingual, Character-Level Neural Conditional Random Fields," *arXiv preprint*, arXiv:2404.09383, 2024.
3. R. Nath, and B. Mahanta," Data Augmentation for Bodo Named Entity Recognition," *International Journal of Data Science and Analysis* **9**(2),2023.
4. M. Sireesha, Srikanth Vemuru, and S. N. TirumalaRao, "Coalesce based binary table: an enhanced algorithm for mining frequent patterns," *International Journal of Engineering and Technology*, vol. 7, no. 1.5, pp. 51–55, 2018.
5. S. Torge, A. Politov, C. Lehmann, B. Saffar, and Z. Tao, "Named Entity Recognition for Low-Resource Languages - Profiting from Language Families," in *Proceedings of the 9th Workshop on Slavic Natural Language Processing (SlavicNLP 2023)*, Association for Computational Linguistics, pp. 1–10, 2023.
6. F. Ullah, A. Gelbukh, M. T. Zamir, E. M. Felipe Riverón, and G. Sidorov, "Enhancement of Named Entity Recognition in Low-Resource Languages with Data Augmentation and BERT Models: A Case Study on Urdu," *Computers*, vol. 13, no. 10, p. 258, 2023.
7. M. Sireesha, S. N. TirumalaRao, and Srikanth Vemuru, "Optimized Feature Extraction and Hybrid Classification Model for Heart Disease and Breast Cancer Prediction," *International Journal of Recent Technology and Engineering*, vol. 7, no. 6, pp. 1754–1772, Mar. 2019, ISSN 2277-3878.
8. M. Sabane, A. Ranade, O. Litake, P. Patil, R. Joshi, and D. Kadam, "Enhancing Low Resource NER Using Assisting Language and Transfer Learning," *arXiv preprint*, arXiv:2306.06477, 2023.
9. Sireesha Moturi, S. N. TirumalaRao, and Srikanth Vemuru, "Grey wolf assisted dragonfly-based weighted rule generation for predicting heart disease and breast cancer," *Computerized Medical Imaging and Graphics*, vol. 91, p. 101936, 2021, ISSN 0895-6111. doi: 10.1016/j.compmedimag.2021.101936.
10. A. Jain, D. Yadav, A. Arora, and D. K. Tayal, "Named entity recognition for Hindi language using context pattern-based maximum entropy," *International Journal on Natural Language Computing (IJNLC)*, vol. 23, no. 1, 2022.
11. G. Prasad, K. K. Fousiya, M. A. Kumar, and K. P. Soman, "Named entity recognition for Malayalam language: A CRF-based approach," *Journal of Ambient Intelligence and Humanized Computing*, vol. 13, pp. 1953–1963, 2022.
12. M. Aparna and S. Srinivasa, "Active learning for named entity recognition in Kannada," in *Proceedings of the 16th International Conference on Natural Language Processing (ICON)*, 2021.
13. R. Brahma and D. Hazarika, "Named entity recognition in Bodo language using conditional random fields," *International Journal of Engineering Research and Technology (IJERT)*, vol. 10, no. 5, 2021.
14. C. N. Subalalitha and R. Srinivasan, "Automated named entity recognition from Tamil documents," *International Journal of Computer Applications*, vol. 178, no. 4, 2019.