Sarcasm Detection in Telugu Language Text using Distinct Machine Learning Classification Algorithms

B. Ravikiran
Research Scholar, Osmania University
Department of CSE, UCE(A), Hyderabad
Telangana

Srinivasu Badugu
Stanley College of Engineering &Technology for Women(A) Abids, Hyderabad
Telangana

ABSTRACT
Sarcasm detection is a growing field in Natural Language Processing (NLP). Sarcasm is identified using positive or more increased positive words, often with a negative connotation, to insult or mock others. In sentiment analysis, detecting sarcasm in the text has become critical. They reviewed numerous relevant research articles, but due to the telugu language's limited resources, detecting sarcasm in telugu language texts remains challenging. As a result, the sentiment detection model struggles to accurately identify the exact sentiment of a sarcastic statement, necessitating the development of an automated sarcasm detection system. Many researchers have trained and tested various machine learning classification algorithms to identify sarcasm, but these algorithms require a dataset as input, which often contains noise.

The dataset undergoes various preprocessing techniques to eliminate noise. Gathered a Telugu conversational dataset from the Kaggle repository, developed their dataset called the Telugu News Headline dataset, labeled the statements as sarcastic or non-sarcastic by the annotators, and then input them into the proposed model. Built the proposed model using SVM (Support Vector Machine), NB (Naive Bayes), and LR (Logistic Regression) and utilized One Hot Encoding (OHE) to transform the dataset into vectors, then fed to the Sarcasm Detection Model to determine the model accuracy. It is trained and tested the Sarcasm detection model on positive or even more positive sentences with 60:40, 70:30, 80:20, and 90:10 splitting ratios to enhance the model performance. By considering the base 70:30 split ratio the best of three algorithms, Logistic Regression resulted in accuracy rates of 65.89% on the imbalanced Telugu conversational dataset and 67.01% on the balanced Telugu conversational dataset. Logistic Regression resulted in accuracy rates of 90.07% on the imbalanced Telugu news headline dataset, and SVM resulted in an accuracy of 98.35% on the balanced Telugu conversational dataset. It is observed that Logistic Regression had better accuracy on the imbalanced and balanced Telugu conversational dataset and the imbalanced Telugu news headline dataset, whereas on the balanced Telugu news headline dataset, SVM had good accuracy. In the future, it can be applied deep learning algorithms to detect sarcasm for better accuracy.

General Terms
NLP,Machine Learning Classification Algorithms,Telugu Language Text, SVM, NB, LR

Keywords
Natural Language Processing; Sarcasm Detection; Machine Learning, Low-resource language

1. INTRODUCTION
NLP lies in between Artificial Intelligence (AI), computer science, and linguistics. NLP encompasses sentiment, and sarcasm is a component of it. Sarcasm, a form of sentiment, originates from the Greek term "sarkasmós," meaning "tear flesh" or "grind the teeth." Simply put, Sarcasm refers to the act of speaking with bitterness. Understanding Sarcasm is a challenge for young children, individuals with autism spectrum disorders, and some patients with brain damage. There are three types of sarcasm: verbal, gestural, and textual. Sarcasm always conveys a negative opinion, even when it employs positive or even more positive words within the text. Sentiment analysis models can't detect the exact sentiment of the given text in Telugu due to the underlying sarcasm in the text. A small number of researchers are currently investigating the detection of sarcasm in Telugu texts. The researchers [1] used a knowledge-based approach. This paper investigates the relevant literature, then expounds on the general architecture of the Sarcasm Detection Model and makes a distinction between different machine learning classification algorithms called SVM, NB, and LR.

2. RELATED WORK
This section provides an overview of current techniques used for detecting sarcasm. The majority of research on sarcasm detection has focused on the English language. Research on low-resource languages like Hindi, Telugu, Tamil, Chinese, Arabic, and others remains limited. The study [2]–[8] discusses preprocessing, feature extraction methods, and comparing several machine learning models on a dataset of 1.3 million social media comments, including both sarcastic and non-sarcastic comments. This study [9] aims to conduct a systematic literature review (SLR) to categorize and analyze the identification of sarcasm in textual data. The hybrid Model [10] surpasses advanced techniques by 3.8%, with 80.64% SARC precision and 95.7% news headline precision. The Reference [11] hybrid model excels at 95.7% news headlines and 80.64% SARC datasets.

The article [12] introduces self-training for tweet-level stress detection (SMTSD) as a new semi-supervised method that analyses four Twitter datasets and identifies those with labeled data shortages. The paper [13] explores how the integration of sentiment, emotion, and personality features using deep learning techniques can significantly enhance the performance of sarcasm detection in social media content, particularly on tweets. The paper [14] explores different methodologies for analyzing political text areas. The paper [15] investigates how education uses sentiment analysis, or opinion mining, and considers how NLP can evaluate student feedback and improve instruction. The abusive comment detection model [16] performed well. The study [17] categorizes Hindi sentiment analysis approaches and discusses their effects on SA issues, levels of analysis, and performance evaluation. The paper [18]

provides a decision tree classification method for Twitter emoticons. The deep learning architecture [19] improves sarcasm detection using the COMET model and textual data. Sarcasm identification in computational linguistics [20] is expanding using rule-based, statistical, and deep-learning methods. This study [21] found hyperbole in an unbiased dataset, which improves sarcasm recognition. This study [22] uses the Abu Farah and Misogyny datasets to detect Arabic sarcasm and misogyny. This study [23] offers a new multi-stage deep learning architecture (MSDLA) for Tamil language sentiment analysis. The paper [24] contains Mann Ki Baat transcription and concludes that a fully effective MLP system requires many tools and strategies to improve language processing accuracy and efficiency. The paper [25] discusses irony and sarcasm detection in Italian.

The researchers compare five deep learning systems and find that the top one achieves a 93% F1-Score. The Chinese translation model AlexNet achieves better semantic recognition accuracy than other neural network algorithms [26]. Popular choices of classification algorithms include Naïve Bayes, support vector machines (SVM), hidden Markov models (HMM), gradient-boosting trees, and random forests [27]. Researchers and developers have used machine learning approaches to detect toxicity [28–30]. The paper [31] presents a hybrid CNN-LSTM model for sentiment analysis on social media data, achieving 91.3% accuracy. The paper [32] proposes a model for extracting product features and sentiment from online reviews using word segmentation, LSTM neural networks, and the LDA topic model. The model proposed in [33] outperforms traditional methods in accurately analyzing sentiments from qualitative data. According to the study [34], factors such as age, English language nativeness, and country have a significant impact on the perception of Sarcasm. These findings suggest that incorporating sociocultural variables can improve the design of social analysis tools to better understand and detect sarcasm online. The paper [35] highlights improvements in preprocessing, segmentation, and POS tagging to enhance the corpus's reliability and usefulness. The paper [36] presents a novel approach called EmoTrans, which uses the transitions between different emotions within a text to detect sarcasm. References [37–38] suggest a model for POS tagging that is based on machine learning and deep learning. They also generate a manually annotated dataset and investigate deep learning architectures such as RNNs. These models deal with low-resource languages' linguistic complexity and achieve higher accuracy and robustness, making NLP capabilities much better.

3. PROPOSED METHODOLOGY

Introduction: It builds the proposed methodology with the One Hot Encoding technique and with machine learning classification algorithms like SVM (Support Vector Machine), NB (Naïve Bayes), and LR (Logistic Regression).

Dataset Collection: This paper has utilized the Telugu language sarcasm dataset, which includes 2422 Telugu conversational datasets and a newly created dataset of 940 Telugu news headlines that manually re-annotated into sarcastic and non-sarcastic classes. The study made use of the conversational dataset, which was available in various formats from the Kaggle repository. Generated the conversational dataset and applied it to kappa coefficient statistics. The Telugu language is the most widely spoken in India. For detecting sarcasm in Telugu-language text, there is only a limited dataset available in online repositories like Kaggle. The proposed methodology architecture describes how the modules will execute the input given to them, and it contains modules as depicted in Figure 1. The proposed system contains modules like preprocessing, vectorization, model creation, and model evaluation. These modules are shown in detail in the following architecture.

Architecture: Architecture describes how the modules will execute the input given to the model, and system modules are depicted in Figure 1.

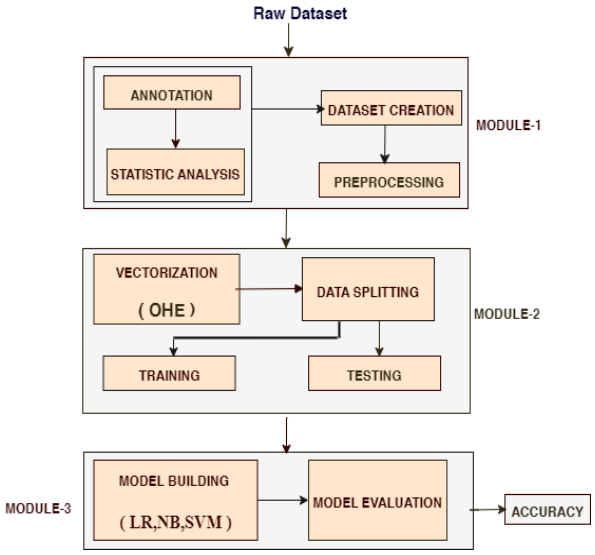

Figure 1: Architecture of the Proposed Methodology

Source : Made by authors

Annotation: Telugu language experts received the collected datasets and manually labeled the sentences to determine whether they were sarcastic or not.

Preprocessing: It is the process of converting text documents into a readable word format. The papers possess numerous characteristics that prepare them for the next stage of text categorization. The process parses a document into a list of tokens after treating it as a string. Since stop words are frequently used, then it must eliminate unimportant terms like punctuation. Data preprocessing may contain other punctuation, but in the dataset, it only has limited punctuation, such as commas, full stops, exclamatory marks, open brackets, and closing brackets. Preprocessing is a crucial module in any language or text. The goal of preprocessing is to eliminate any noise from the corpus. Bag of Words techniques convert textual data into vectors of numbers, enabling machine learning classification algorithms to perform tasks such as classification, semantic analysis, and prediction.

Vectorization: The process of converting data into numbers is called vectorization. The machine cannot understand words, so it requires numerical values to make it easier for it to process the data. To apply any type of algorithm to the data, it needs to convert categorical data into numbers. It converts the preprocessed data into vectors through vectorization. Machine learning techniques require the conversion of the string or text into a vector, a set of real numbers, to process natural language text and extract useful information from the given word or sentence. During the classification phase, it will classify the collected statements using algorithms like Naive Bayes, Support Vector Machine, and Logistic Regression to detect the presence of sarcasm in the statement. They will divide the dataset into training and testing sections to train and test the model. To evaluate and interpret natural language data, such as text or speech, NLP requires the integration of machine

learning techniques. The most common evaluation metrics include accuracy, precision, recall, and f1-score. The Telugu language is the most widely spoken in India. For detecting sarcasm in the Telugu language, there are limited datasets available on online repositories like Kaggle.

Table 1. Web Based Translated Sentences

English Sentence	Transliteration	Translation in Telugu	Type
Rulers who are taking development back in their countries like a time machine	Tama desalalo abhivrddhini taim meshin laga venakki tisukeltunna palakulu	తమ దేశాలలో అభివృద్ధిని టైంమెషిన్ లాగ వెనక్కి తీసుకెళ్తున్న పాలకులు	Sarcastic
Osmania University has a hundred years of history	Osmania university vanda endla charitra kaladi	ఉస్మానియా యూనివర్సిటీ వంద ఏండ్ల చరిత్ర కలది	Non-Sarcastic

A web-based translator is used to translate the english language dataset into Telugu. It is a statistical translation system. Existing translation systems are effective at some level, but some words in a sentence do not translate according to context, as shown in Table 1. In this context, postprocessing is required, as shown in Table 2. Preprocessing is a crucial module in any language or text-processing application. In this paper, it used preprocessing to remove noise from the corpus. It includes eliminating special characters and removing unwanted text using the Escaped Unicode Representation.

Table 2. Post Processing for telugu language text

Telugu Sentence	Escaped Unicode	Post Processing	Data set Type
“మీ సలహాలు మేము పాటించలేనంత గొప్పగా ఉన్నాయ్”.	“మీ సలహాలు మేము పాటించలేనంత గొప్పగా ఉన్నాయ్”.	మీ సలహాలు మేము పాటించలేనంత గొప్పగా ఉన్నాయ్	Sarcastic
“చెక్కతో చేసిన ఉపగ్రహాన్ని ప్రయోగించనున్న జపాన్”.	“చెక్కతో చేసిన ఉపగ్రహాన్ని ప్రయోగించనున్న జపాన్”.	చెక్కతో చేసిన ఉపగ్రహాన్ని ప్రయోగించనున్న జపాన్	Non Sarcastic

Computers are symbolic processing machines. Computers process numbers quickly, but they cannot understand any text data. All machine learning and deep learning algorithms work with numbers very efficiently. The algorithms cannot directly process text; instead, they require an encoding system to convert text data into numerical vectors. Vectorization is the process of converting text data into numerical vectors. In this paper, three machine learning classification algorithms were tested for the Telugu language text to identify whether the given input sentence from the dataset is sarcastic or nonsarcastic.The dataset is divided into training and testing segments using varying ratios. The following section provides a detailed analysis of the results.

4. RESULT & ANALYSIS

Collected the Telugu conversational sarcastic and nonsarcastic dataset from the Kaggle repository, and it was imbalanced because sarcastic sentences were 1580 and non-sarcastic sentences were 842. The dataset is balanced using the oversampling method, resulting in 842 sarcastic sentences and 842 non-sarcastic sentences and conducted experiments on both datasets using the Python environment and accounted for the data distribution, also known as training and testing measures. These distribution measures are used to determine the model's accuracy.

Telugu Conversational Dataset: Executing the Imbalance Telugu Conversational Dataset Using the Python environment, it obtained a total of 11,553 unique words, 45,454 words in the corpus, and a vocabulary length of 11514. Furthermore, it is observed that the value of these unique words, the total number of words in the corpus, and the vocabulary length remained consistent. Executing and Balancing the Telugu Conversational dataset Using the Python environment, it can calculate the total number of unique words (9740), the total number of words in the corpus (31,544), and the length of the vocabulary (9740). It also observed that under all four types of splitting ratios, the value of these unique words, the total number of words in the corpus, and the length of the vocabulary remain the same.

Table 3: Confusion Matrix of imbalanced Telugu Conversational dataset using OHE

Table 3(a)

0	395	476
1	370	1183
	0	1

Table 3(a) shows the imbalanced Telugu Conversational dataset confusion matrix.There are 1,183 true negatives,370 false negatives,476 fasle positive,and 395 true positive.

Table 3(b)

0	458	413
1	510	1043
	0	1

Table 3(b) shows the imbalanced Telugu Conversational dataset confusion matrix.There are 1,043 true negatives,510 false negatives,413 fasle positive,and 458 true positive.

Table 3(c)

0	327	544
1	288	1265
	0	1

Table 3(c) shows the imbalanced Telugu Conversational dataset confusion matrix.There are 1,265 true negatives,288 false negatives,544 fasle positive,and 327 true positive.

Table 4: Confusion Matrix of balanced Telugu Conversational dataset using OHE

Table 4(d)

0	508	313
1	280	595
	0	1

Table 4(d) shows the balanced Telugu Conversational dataset confusion matrix.There are 595 true negatives,280 false negatives,313 fasle positive,and 508 true positive.

Table 5(e)

0	469	352
1	280	585
	0	1

Table 5(e) shows the balanced Telugu Conversational dataset confusion matrix.There are 585 true negatives,280 false negatives,352 fasle positive,and 469 true positive.

Table 5(f)

0	521	300
1	270	595
	0	1

Table 5(f) shows the balanced Telugu Conversational dataset confusion matrix.There are 595 true negatives,270 false negatives,300 fasle positive,and 521 true positive.

Table 6: Confusion Matrix of imbalanced Telugu News Headline dataset using OHE

Table 6(g)

0	**846**	**6**
1	**86**	**2**
	0	**1**

Table 6(g) shows the imbalanced Telugu news headline dataset confusion matrix.There are 2 true negatives,86 false negatives,6 fasle positive,and 846 true positive.

Table 6(h)

0	686	169
1	50	38
	0	1

Table 6(h) shows the imbalanced Telugu news headline dataset confusion matrix.There are 38 true negatives,50 false negatives,169 fasle positive,and 686 true positive.

Table 6(i)

0	852	0
1	88	0
	0	1

Table 6(i) shows the imbalanced Telugu news headline dataset confusion matrix.There are 0 true negatives,88 false negatives,0 fasle positive,and 852 true positive.

Table 7: Confusion Matrix of balanced Telugu News Headline dataset using OHE

Table 7(j)

0	772	19
1	13	816
	0	1

Table 7(j) shows the balanced Telugu news headline dataset confusion matrix.There are 816 true negatives,13 false negatives,19 fasle positive,and 772 true positive.

Table 7(k)

0	585	206
1	10	819
	0	1

Table 7(k) shows the balanced Telugu news headline dataset confusion matrix.There are 819 true negatives,10 false negatives,206 fasle positive,and 585 true positive.

Table 7(l)

0	781	10
1	19	810
	0	1

Table 7(l) shows the balanced Telugu news headline dataset confusion matrix.There are 810 true negatives,19 false negatives,10 fasle positive,and 781 true positive.

Table 8. Summary of results on imbalanced telugu conversational dataset using OHE

Model	Precision	Recall	F1-Score	Accuracy
SVM	0.70	0.75	0.73	64.23%
NB	0.71	0.67	0.69	62.31%
LR	0.69	0.82	0.75	65.89%

The model is trained with different training and testing percentages like 60:40,70:30,80:20 and 90:10 on imbalanced telugu conversational dataset among three algorithms and observed that at 70:30 base splitting ratio Logistic Regression has given best accuracy with 65.89% when compare to other algorithms as described in the table 8.

Table 9. Summary of results on balanced telugu conversational dataset using OHE

Model	Precision	Recall	F1-Score	Accuracy
SVM	0.67	0.68	0.68	65.61%
NB	0.64	0.69	0.67	63.83%
LR	0.68	0.70	0.69	67.01%

The model is trained with different training and testing percentages like 60:40,70:30,80:20 and 90:10 and among three algorithms and observed that at 70:30 base splitting ratio

Logistic Regressionn has given best accuracy with 67.01% compare to other algorithms as described in the table 9.

Figure 2 and Figure 3 describes SVM,NB and LR algorithms under 60:40,70:30,80:20 and 90:10 training and testing ratios

accuracy of Telugu conversational and Telugu news headline imbalanced and balanced dataset. It's observed when the training percentage is more, accuracy is more and when training percentage is less, accuracy is more or less equal among two algorithms ,because of that it is considered base splitting ratio 70:30 to measure high accuracy among three algorithms.

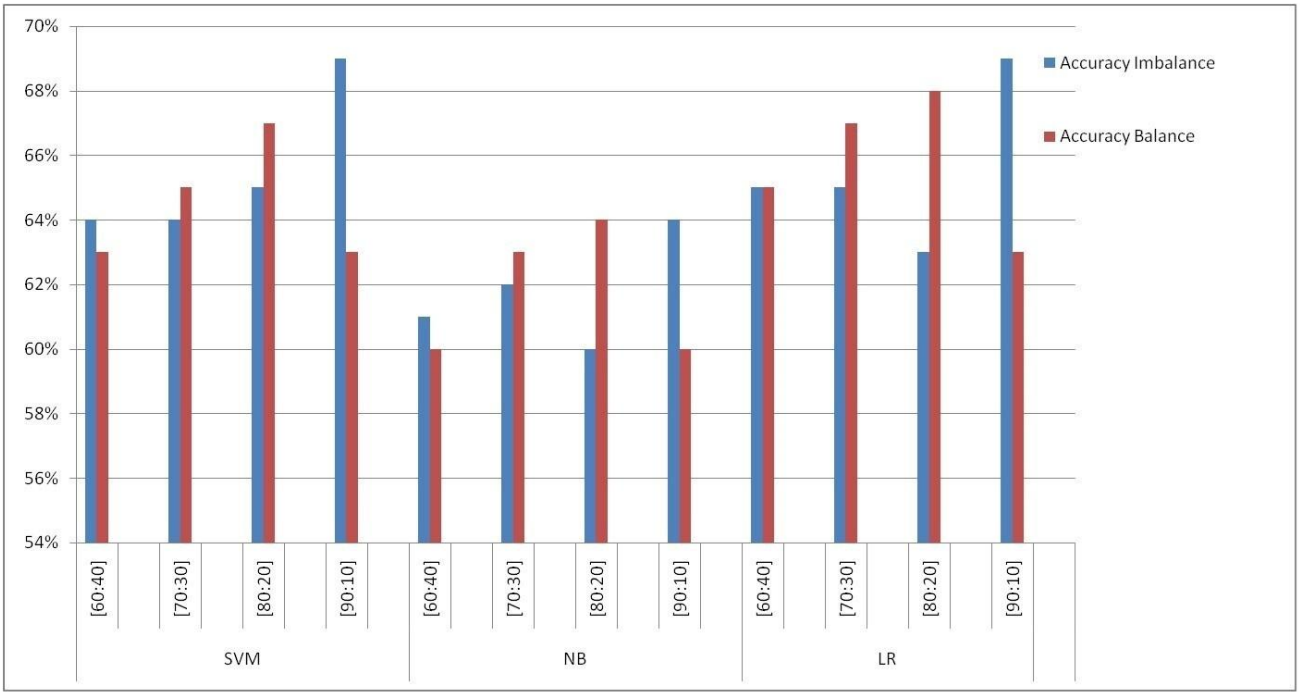

Fig 2: Accuracy analysis of Telugu Conversational Imbalanced and Balanced Dataset

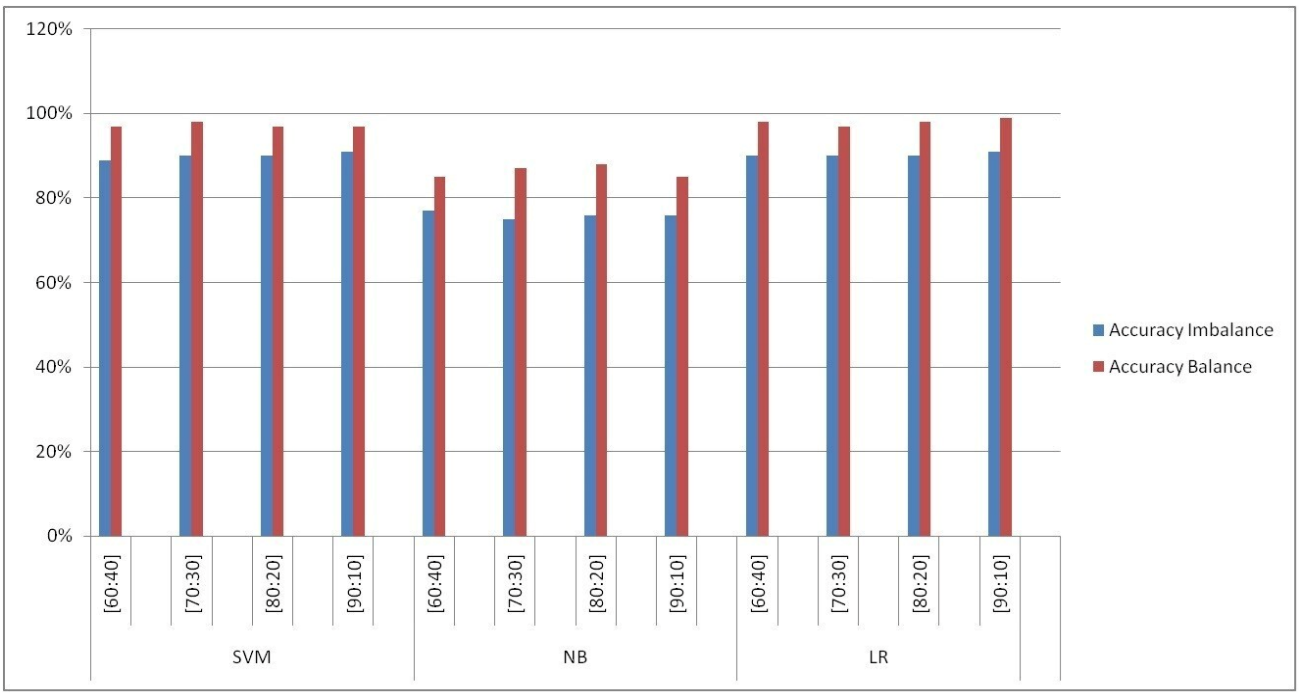

Fig 3: Accuracy analysis of Telugu News Headline Imbalanced and Balanced Dataset

Figure 2 and Figure 3 describes SVM,NB and LR algorithms under 60:40,70:30,80:20 and 90:10 training and testing ratios accuracy of Telugu conversational and Telugu news headline imbalanced and balanced dataset. It is observed when training percentage is more accuracy is more and when training percentage is less, accuracy is more or less equal among two algorithms ,because of that it considered base splitting ratio 70:30 to measure high accuracy among three algorithms.

Telugu News Headline Dataset: Experimenting on imbalace Telugu News Headline Dataset Using python environment and it gets total number of uniqe words 5240 and total number of

words in corpus 8485 and length of vocabulary 5240 and it is also observed under all four kinds of splitting ratios these Unique words , total number of words in the corpus and length of vocabulary are same in value.Experimenting on balaced Telugu News Headline Dataset Using python environment for 60:30 training and testing ratio and it gets total number of uniqe words 5240 and total number of words in corpus 13936 and length of vocabulary 5239.for 70:30 training and testing ratio and it gets total number of uniqe words 5240 and total number of words in corpus 14336 and length of vocabulary 5237.for 80:20 training and testing ratio and it gets total number of uniqe words 5240 and total number of words in corpus 13793 and length of vocabulary 5239. for 90:10 training and testing ratio and it gets total number of uniqe words 5240 and total number of words in corpus 13869 and length of vocabulary 5239.

Table 10. Summary of results on imbalanced telugu news headline dataset using OHE

Model	Precision	Recall	F1-Score	Accuracy
SVM	0.50	0.04	0.07	90.07%
NB	0.18	0.43	0.26	75.53%
LR	0.20	0.45	0.28	80.07%

The model is trained with different training and testing percentages like 60:40,70:30,80:20 and 90:10 on imbalanced telugu headline dataset among three algorithms and observed that at 70:30 base splitting ratio SVM has given best accuracy with 90.07% when compare to other algorithms as described in the table 10.

Table 11. Summary of results on balanced telugu news headline dataset using OHE

Model	Precision	Recall	F1-Score	Accuracy
SVM	0.98	0.98	0.98	98.35%
NB	0.81	0.98	0.89	87.45%
LR	0.99	0.96	0.98	97.53%

The model is trained with different training and testing percentages like 60:40,70:30,80:20 and 90:10 on balanced telugu headline dataset among three algorithms and observed that at 70:30 base splitting ratio SVM has given best accuracy with 98.35% when compare to other algorithms as described in the table 11.

5. CONCLUSION

The objective of this paper is to design a Machine Learning Classification Model for detecting Sarcasm in Telugu Language text. The model is used One Hot Encoding word embedding method and multiple Machine Learning Classification Algorithms on two telugu language text datasets and it is observed that the majority of researchers worked on Western languages, and very few researchers are working on low resource language called Telugu.To demonstrate the potential of this approach, it is conducted experiments on two telugu language text datasets of different genres and sizes and applied one hot encoding word embedding method with different training and testing ratios. The vectors are tested using Machine Learning Classification Algorithms with one hot encoding and observed the performance measures like precision,recall,f1-score and accuracy. The model is trained and tested using three classification algorithms and obtained the accuracy 67% with SVM, 64% with Naive Bayes, 68% with logistic regression on the Telugu Conversational Dataset, and 98% with SVM, 88% with Naive Bayes, and 99% with logistic regression on Telugu News Headline dataset. Logistic Regression brings a high accuracy rate with the both datasets. In the future, the breadth of this Model can be applied to detect Sarcasm automatically using deep learning approaches with different word embedding methods and with different dataset sizes.This classification could be used as a language independent classification Model.

6. REFERENCES

[1] Joshi, A., Bhattacharyya, P., & Carman, M. J. (2017). "Automatic Sarcasm Detection : A Survey". *ACM ComputingSurveys*, *50*(5),1–22. https://doi.org/10.1145/3124420

[2] Misra, R., & Arora, P. (2023). "Sarcasm Detection using news headlines dataset". *AI Open*. https://doi.org/10.1016/j.aiopen.2023.01.001

[3] Otter, D. W., Medina, J. R., & Kalita, J. K. (2020). "A Survey of the Usages of Deep Learning for Natural Language Processing". *IEEE Transactions on Neural Networks and Learning Systems*, *32*(2), 1–21. https://doi.org/10.1109/TNNLS.2020.2979670

[4] Šandor, D. and Bagić Babac, M. (2024), "Sarcasm Detection in online comments using machine learning", *Information Discovery and Delivery*, Vol. 52 No. 2, pp. 213-226. https://doi.org/10.1108/IDD-01-2023-0002

[5] Rahma, A., Azab, S. S., & Mohammed, A. (2023). "A Comprehensive Survey on Arabic Sarcasm Detection: Approaches, Challenges and Future Trends". *IEEE Access*, *11*,18261–18280. https://doi.org/10.1109/access.2023.3247427

[6] Razali, M. S., Halin, A. A., Ye, L., Doraisamy, S., & Norowi, N. M. (2021). "Sarcasm Detection Using Deep Learning With Contextual Features". *IEEE Access*, *9*, 68609–68618. https://doi.org/10.1109/ACCESS.2021.3076789

[7] Ravi Teja Gedela, Ujwala Baruah, & Soni, B. (2023). "Deep Contextualised Text Representation and Learning for Sarcasm Detection". *Arabian Journal for Science and Engineering*, *49*(3), 3719–3734. https://doi.org/10.1007/s13369-023-08170-4

[8] Kumar, A., & Garg, G. (2019). "Empirical study of shallow and deep learning Models for Sarcasm Detection using context in benchmark datasets". *Journal of Ambient Intelligence and Humanized Computing*. https://doi.org/10.1007/s12652-019-01419-7

[9] Eke, C. I., Norman, A. A., Shuib, L., & Nweke, H. F. (2019). "Sarcasm identification in textual data: systematic review, research challenges and open directions". *Artificial Intelligence Review*, *53*(6), 4215–4258. https://doi.org/10.1007/s10462-019-09791-8

[10] Ravi Teja Gedela, Pavani Meesala, Ujwala Baruah, & Soni, B. (2023). "Identifying Sarcasm using heterogeneous word embeddings: a hybrid and ensemble perspective". *SoftComputing*. https://doi.org/10.1007/s00500-023-08368-6

[11] Xiong, T., Zhang, P., Zhu, H., & Yang, Y. (2019). "Sarcasm Detection with Self-matching Networks and Low-rank Bilinear Pooling". *The World Wide Web*

Conference on - WWW '19. https://doi.org/10.1145/3308558.3313735

[12] Prashanth KVTKN, & Tene Ramakrishnudu. (2023). "Semi-supervised approach for tweet-level stress detection". *Natural Language Processing Journal*, 100019–100019. https://doi.org/10.1016/j.nlp.2023.100019

[13] Poria, S., Cambria, E., Hazarika, D., & Vij, P. (2016). "*A Deeper Look into Sarcastic Tweets Using Deep Convolutional Neural Networks"*. ArXiv.org. https://arxiv.org/abs/1610.08815

[14] Doan, T. M., & Gulla, J. A. (2022). "A Survey on Political Viewpoints Identification". *Online Social Networks and Media*, *30*, 100208. https://doi.org/10.1016/j.osnem.2022.100208

[15] Shaik, T., Tao, X., Dann, C., Xie, H., Li, Y., & Galligan, L. (2023). "Sentiment analysis and opinion mining on educational data: A survey". *Natural Language Processing Journal*, *2*, 100003. https://doi.org/10.1016/j.nlp.2022.100003.

[16] Chakravarthi, B. R., Priyadharshini, R., Banerjee, S., Jagadeeshan, M. B., Kumaresan, P. K., Ponnusamy, R., Benhur, S., & McCrae, J. P. (2023). "Detecting abusive comments at a fine-grained level in a low-resource language". *Natural Language Processing Journal*, *3*, 100006. https://doi.org/10.1016/j.nlp.2023.100006.

[17] Kulkarni, D. S., & Rodd, S. S. (2022). "Sentiment Analysis in Hindi—A Survey on the State-of-the-art Techniques". *ACM Transactions on Asian and Low-Resource Language Information Processing*, *21*(1), 1–46. https://doi.org/10.1145/3469722

[18] M. Nirmala, Gandomi, A. H., Madda Rajasekhara Babu, Babu, D., & Rizwan Patan. (2024). "An Emoticon-Based Novel Sarcasm Pattern Detection Strategy to Identify Sarcasm in Microblogging Social Networks". *IEEE Transactions on Computational Social Systems*, 1–8. https://doi.org/10.1109/tcss.2023.3306908

[19] Li, J., Pan, H., Lin, Z., Fu, P., & Wang, W. (2021). "Sarcasm Detection with Commonsense Knowledge". *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *29*, 3192–3201. https://doi.org/10.1109/taslp.2021.3120601

[20] He, S., Guo, F., & Qin, S. (2020). "*Sarcasm Detection Using Graph Convolutional Networks with Bidirectional LSTM"*. https://doi.org/10.1145/3422713.3422722

[21] Govindan, V., & Balakrishnan, V. (2022). "A machine learning approach in analysing the effect of hyperboles using negative sentiment tweets for Sarcasm Detection". *Journal of King Saud University - Computer andInformationSciences*. https://doi.org/10.1016/j.jksuci.2022.01.008

[22] Muaad, A. Y., Jayappa Davanagere, H., Benifa, J. V. B., Alabrah, A., Naji Saif, M. A., Pushpa, D., Al-antari, M. A., & Alfakih, T. M. (2022). "Artificial Intelligence-Based Approach for Misogyny and Sarcasm Detection from Arabic Texts". *Computational Intelligence and Neuroscience*, *2022*,e7937667. https://doi.org/10.1155/2022/7937667

[23] Jothi Prakash V, & Vijay, A. (2023). "Cross-lingual Sentiment Analysis of Tamil Language Using a Multi-stage Deep Learning Architecture. *ACM Transactions on Asian and Low-Resource Language Information Processing*, *22*(12),1–28. https://doi.org/10.1145/3631391

[24] Lahoti, P., Mittal, N., & Singh, G. (2022). A Survey on NLP resources, tools and techniques for Marathi Language Processing. *ACM Transactions on Asian and Low-Resource Language Information Processing*. https://doi.org/10.1145/3548457

[25] Alcamo, T., Cuzzocrea, A., Bosco, G. L., Pilato, G., & Schicchi, D. (2020). Analysis and Comparison of Deep Learning Networks for Supporting Sentiment Mining in Text Corpora. *Proceedings of the 22nd International Conference on Information Integration and Web-Based Applications&Services*. https://doi.org/10.1145/3428757.3429144

[26] Feng, H., Xie, S., Wei, W., Haibin, L., & Zhihan, L. (2022). Deep Learning in Computational Linguistics for Chinese Language Translation. *ACM Transactions on Asian and Low-Resource Language Information Processing*. https://doi.org/10.1145/3519386

[27] Minaee, S., Kalchbrenner, N., Cambria, E., Nikzad, N., Chenaghlu, M., & Gao, J. (2021). Deep Learning--based Text Classification. *ACM Computing Surveys*, *54*(3), 1–40. https://doi.org/10.1145/3439726

[28] Poeller, S., Dechant, M., Klarkowski, M., & Mandryk, R. L. (2023). Suspecting Sarcasm: How League of Legends Players Dismiss Positive Communication in Toxic Environments. *Proceedings of the ACM on Human-Computer Interaction*, *7*(CHI PLAY), 1–26. https://doi.org/10.1145/3611020

[29] Son, L. H., Kumar, A., Sangwan, S. R., Arora, A., Nayyar, A., & Abdel-Basset, M. (2019). Sarcasm Detection Using Soft Attention-Based Bidirectional Long Short-Term Memory Model With Convolution Network. *IEEE Access*, *7*, 23319–23328. https://doi.org/10.1109/access.2019.2899260

[30] Zhang, Y., Yu, Y., Wang, M., Huang, M., & M. Shamim Hossain. (2023). Self-Adaptive Representation Learning Model for Multi-Modal Sentiment and Sarcasm Joint Analysis. *ACM Transactions on Multimedia Computing, Communications and Applications/ACM Transactions on Multimedia Computing Communications and Applications*. https://doi.org/10.1145/3635311

[31] Jain, P. K., Saravanan, V., & Pamula, R. (2021). A Hybrid CNN-LSTM: A Deep Learning Approach for Consumer Sentiment Analysis Using Qualitative User-Generated Contents. *ACM Transactions on Asian and Low-Resource Language Information Processing*, *20*(5), 1–15. https://doi.org/10.1145/3457206

[32] Cao, J., Li, J., Yin, M., & Wang, Y. (2022). Online reviews sentiment analysis and product feature improvement with deep learning. *ACM Transactions on Asian and Low-Resource Language Information Processing*. https://doi.org/10.1145/3522575

[33] Jothi Prakash V, & Vijay, A. (2023). Cross-lingual Sentiment Analysis of Tamil Language Using a Multi-stage Deep Learning Architecture. *ACM Transactions on Asian and Low-Resource Language Information Processing*, *22*(12), 1–28. https://doi.org/10.1145/3631391

[34] Oprea, S. V., & Magdy, W. (2020). The Effect of Sociocultural Variables on Sarcasm Communication Online. *Proceedings of the ACM on Human-Computer Interaction*, *4*(CSCW1), 1–22. https://doi.org/10.1145/3392834

[35] Meelen, M., Roux, É., & Hill, N. (2021). Optimisation of the Largest Annotated Tibetan Corpus Combining Rule-based, Memory-based, and Deep-learning Methods. *ACM Transactions on Asian and Low-Resource Language Information Processing*, *20*(1), 1–11. https://doi.org/10.1145/3409488

[36] Agrawal, A., An, A., & Manos Papagelis. (2020). *Leveraging Transitions of Emotions for Sarcasm Detection*. https://doi.org/10.1145/3397271.3401183

[37] Tusarkanta Dalai, Tapas Kumar Mishra, & Sa, P. K. (2024). Deep Learning-based POS Tagger and Chunker for Odia Language Using Pre-trained Transformers. *ACM Transactions on Asian and Low-Resource Language Information Processing*, *23*(2), 1–23. https://doi.org/10.1145/3637877.

[38] R Prasanna Kumar, G Bharathi Mohan, Yamani Kakarla, L, J. S., Kolla Gnapika Sindhu, Sai, V., Ganesh, B., & Nunna Hasmitha Krishna. (2023). *Sarcasm Detection in Telugu and Tamil: An Exploration of Machine Learning and Deep Neural Networks*. https://doi.org/10.1109/icccnt56998.2023.10306775