A Progressive Step Towards Automated Fact-Checking by Detecting Context in Diverse Languages: A Prototype for Bangla Facebook Posts

Mahbuba Shefa[1], Tasnuva Ferdous[1], Afzal Azeem Chowdhary[2][a], Md Jannatul Joy[2], Tanjila Kanij[2][b] and Md Al Mamun[2]

[1]*Rajshahi University of Engineering & Technology, Rajshahi, Bangladesh*

[2]*Swinburne University of Technology, Melbourne, Australia*

Keywords: Fact-Checking, Facebook, Large Language Models, Context Detection.

Abstract: Fact-Checking has become a critical tool in combating misinformation, particularly on platforms like Facebook, where the rapid spread of false information poses significant challenges. Much work has been done on languages like English but not on low-resource languages like Bangla. To address this gap, we explored the application of classic ML models, RNNs, and BanglaBERT on a small dataset of Bangla Facebook textual posts to understand its context. Surprisingly, BanglaBERT underperformed compared to traditional approaches like models based on TF-IDF embeddings, highlighting the challenges of working with limited data and insufficient fine-tuning. To support fact-checkers, we developed the "Automated Context Detector," which is developed with NLP and machine learning that automates repetitive tasks, allowing experts to focus on critical decisions. Our results demonstrate the feasibility of using machine learning for context detection in Bangla social media posts, providing a framework adaptable to similar linguistic and cultural settings.

1 INTRODUCTION

Social media has firmly established itself as a platform for social interaction and information dissemination in the daily lives of billions of people. Facebook is one of the most popular social media worldwide. Bangladesh ranks as the 8th largest country in terms of Facebook audience size (Dixon, 2024).

With social media as a primary news source, misinformation spreads easily, posing severe societal risks. Lower literacy rates and limited access to reliable sources in underdeveloped countries like Bangladesh make misinformation especially harmful.

Bangladesh has had to endure unexpected and life-threatening acts of violence caused by the spread of misinformation through Facebook (Ali, 2020) (Minar and Naher, 2018) (Naher and Minar, 2018). A detailed analysis of the reported incidents in the literature shows that they all originated from Facebook posts. To mitigate such unforeseen circumstances, it is important to check the credibility of every piece of information, referred to as fact-checking (Chan et al., 2017). However, fact-checking is a multi-step and time-consuming process requiring heavy manual intervention. Automating Fact-checking, or parts of it, could significantly benefit journalism and assist the public in verifying the credibility of various media. Developing a robust automated fact-checking system requires establishing effective methods for evaluating its performance. While publicly available datasets exist for English to support this evaluation, no systematic research has been conducted for other underrepresented low-resource languages, such as Bangla.

We developed a prototype tool to classify Bangla Facebook posts by context, extendable to other low-resource languages. Fact-checking requires external knowledge and contextual understanding, demanding significant manual effort from fact-checkers in Bangladesh (reference removed for anonymity). This study categorizes manually scraped posts into four topics—Health, Religion, Politics, and Miscellaneous—and automates classification to ease fact-checkers' workload. It also evaluates machine learning models' effectiveness in context detection for Bangla, a non-English language.

The overall contribution of this research work is-

[a] https://orcid.org/0000-0003-2722-6424

[b] https://orcid.org/0000-0002-5293-1718

Shefa, M., Ferdous, T., Chowdhary, A. A., Joy, M. J., Kanij, T. and Al Mamun, M.
A Progressive Step Towards Automated Fact-Checking by Detecting Context in Diverse Languages: A Prototype for Bangla Facebook Posts.
DOI: 10.5220/0013430000003928
In *Proceedings of the 20th International Conference on Evaluation of Novel Approaches to Software Engineering (ENASE 2025)*, pages 684-691
ISBN: 978-989-758-742-9; ISSN: 2184-4895

1. This study introduces a prototype for automated context detection in low-resource languages like Bangla.
2. We propose a scalable framework for fact-checking, focusing on context detection, using Bangla as a case study.
3. A growing dataset of Bangla Facebook posts with extracted features is provided for future research.
4. Machine learning models were applied to analyze the dataset, demonstrating the effectiveness of the proposed tool in identifying context.
5. This work lays the foundation for developing tools for underrepresented languages, showcasing the potential of advanced methodologies.
6. This study evaluates classic ML models and advanced approaches like RNN and BanglaBERT, highlighting their strengths, limitations, and best-use cases in Bangla processing.

The paper is structured as follows: Section 2 covers fact-checking background, Section 3 details our prototype, Section 4 evaluates it on a small dataset, Section 5 discusses findings, Section 6 outlines limitations, and Section 7 concludes with future directions.

2 BACKGROUND

2.1 Fact Checking

In today's digital age, the abundance of readily available information has increased the use of the term "Fact-checking". As misinformation spreads easily, fact-checking has become essential to mitigate its impact, as explored in detail in the following sections.

2.1.1 Defining Fact Checking

The primary form of fact-checking is debunking, which involves "presenting a corrective message that clarifies the previous message as misinformation" (Chan et al., 2017). Fact-checking entails verifying the accuracy of statements, news, and informative content in media, including social media, by thoroughly investigating reliable sources and evidence.

Sensationalist newspapers in the 1850s fueled a demand for factual media, driving the evolution of fact-checking (Dickey, 2019). Milestones include Time magazine (Fabry, 2017), the Associated Press, Pulitzer's Bureau of Accuracy and Fair Play (1912), and The New Yorker's fact-checking department (Dickey, 2019). Strengthening fact-checking as seen in The Washington Post and *PolitiFact* remains crucial today.

2.1.2 Why Fact Checking Is Needed

The rise of social media as a primary news source has simultaneously made it a platform for disseminating harmful fake information, which impacts individuals and society. Misinformation disrupts the authenticity of the information ecosystem, misleading consumers, promoting biased narratives, and enabling the exploitation of social media for financial or political gain which threatens social stability and security.

While many incidents worldwide highlight the importance of fact-checking, COVID-19 and the 2020 US Presidential election have brought it into the spotlight. With 68% of US adults getting news from social media, (Hitlin and Olmstead, 2018), misinformation led to an infodemic, shaking public trust in the COVID-19 vaccine (Carey et al., 2022) (Eysenbach et al., 2020) (Kreps and Kriner, 2022). Similarly, President Trump and Republican officials spread false claims of election fraud, fueling the 2021 Capitol riot. Years of misinformation allowed conspiracy theories to move from obscure online spaces into mainstream media and politics (Roose, 2021) (Tollefson, 2021).

Countries like Bangladesh face greater challenges due to low literacy and socio-economic factors, making fact-checking vital to prevent severe consequences.

2.2 Fact-Checking Worldwide

With disinformation spreading rapidly, fact-checking organizations have expanded globally (Haque et al., 2020). Duke Reporters' Lab recorded 149 projects in 53 countries in 2018, up from 114 in 2017, but growth slowed to 341 active projects in 2021 (Stencel et al., 2021). While 87% of US fact-checkers are linked to major news outlets, only 53% outside the US have such affiliations (Haque et al., 2018). Most rely on manual fact-checking, while others explore automation to keep pace. However, full automation raises concerns over AI's ethics, safety, and geopolitical risks. While automation advances in high-resource languages, low-resource languages lag, highlighting the need for inclusive fact-checking solutions.

2.2.1 Facebook Fact-Checking

Facebook employs independent third-party fact-checkers to carry out fact-checking on its platform. These fact-checkers review and assess the accuracy of content posted on Facebook to identify false or misleading information. When content is flagged as misinformation, Facebook reduces its distribution and displays warning labels to alert users about the inaccuracies. Collaborating with independent fact-checkers helps Facebook combat the spread of fake news and misinformation and tries to ensure a more trustworthy and reliable platform for its users.

2.2.2 Facebook Fact-Checking in Bangladesh

Fact-checking organizations in Bangladesh include BD FactCheck, Rumor Scanner, FactWatch, Boom Bangladesh, and AFP Fact Check (Hossain et al., 2022); the last three are linked to Meta. FactWatch operates solely in Bangladesh, BOOM Bangladesh in India, Bangladesh, and Myanmar, and AFP operates globally. Several IFCN-certified fact-checkers also work to combat misinformation in Bangladesh. Based on interviews with eight Bangladeshi fact-checkers, our previous research found that most fact-checking remains manual (reference removed for anonymity). While "critical thinking" requires human intervention, automating tasks like context detection could accelerate and enhance the process.

2.3 Automated Fact-Checking

The growing spread of misinformation makes manual fact-checking labor-intensive and difficult for organizations to keep up with whereas automated approaches offer the potential for sustainable solutions. This subsection briefly overviews automated fact-checking using machine learning and NLP primarily for the English language. A comprehensive overview is available in (Thorne and Vlachos, 2018), with additional studies discussed here.

Online misinformation is a significant challenge, and AI-driven fact-checking still requires supervision. The first automated fact-checking process involved manually labelling datasets and defining fact-checking as assigning a Boolean truth value to a claim within a specific context (Vlachos and Riedel, 2014).

A hybrid human-in-the-loop framework combining AI, crowdsourcing, and expert input was proposed for scalable misinformation tackling (Barbera et al., 2023). An Arabic fact-checking corpus integrated tasks like document retrieval, source credibility, stance detection, and rationale extraction (Baly et al., 2018). A pipeline-based approach for fact-checking included document retrieval, stance detection, evidence extraction, and claim validation (Hanselowski, 2020).

3 PROTOTYPE TOOL

This preliminary research develops a prototype tool for context detection to accelerate fact-checking in Bangla. A small-scale trial with various Machine Learning (ML) models assesses practical feasibility. We detail dataset curation, including collection, organization, preprocessing, and refinement. The automation process proposes a Context Detector using different ML algorithms and BanglaBERT, with tailored preprocessing for each approach. For clarity, ML algorithms are categorized as Classic and Advanced.

We outline training/testing methodologies, experimental setup, and hyperparameter tuning. Finally, model performance is evaluated using appropriate metrics. Figure 1 illustrates the research workflow.

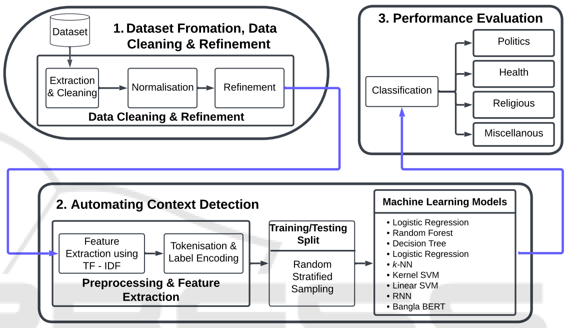
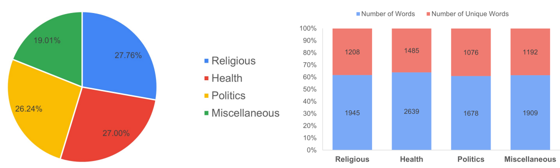


Figure 1: Automated Context Detection Process Flow.

3.1 Dataset Formation, Data Cleaning & Refinement

Dataset Formation: We curated a dataset of 267 Bangla Facebook posts from diverse sources, including Rumour Scanner Bangladesh (Rumour Scanner Bangladesh, 2020) and Jachai (Jachai, 2017). These fact-checked posts were categorized into four contexts: Politics, Religious, Health, and Miscellaneous, ensuring a balanced distribution (Guo et al., 2008), as shown in Figure 2a. Manual labelling with cross-verification ensured accuracy.



(a) Number of Posts by Label (b) Word Distribution

Figure 2: Distribution of Posts According to Category.

We paraphrased posts using ChatGPT to expand the dataset, increasing the sample size to 562. Each paraphrased post was manually reviewed to preserve con-

textual integrity. However, this process was labour-intensive, limiting large-scale expansion.
Data Cleaning: Upon initial cleaning, non-Bangla characters, extraneous text, and punctuation were removed, ensuring uniform Bangla-language content. Spaces replaced new lines for structural consistency.
Data Refinement: Further refinement eliminated special characters, punctuation, and emojis using regular expressions. To address class imbalance (Guo et al., 2008), selective row removal balanced the dataset while maintaining representative class distributions (Figure 2a). This rigorous preprocessing established a robust dataset for subsequent machine learning tasks. Figure 2b illustrates the total and unique word distributions per class after balancing.

3.2 Automating Context Detection

The ML models listed in Table 1 were chosen for our analysis; they are the most commonly used for classification tasks (Kotsiantis et al., 2006).

Linear Regression is valued for its simplicity and effectiveness in classification. Decision Trees aid in feature selection and decision analysis, while Random Forests handle high-dimensional data and resist overfitting. Multinomial Naive Bayes excels in large-scale text classification, particularly in NLP. SVMs perform well in high-dimensional classification with clear margins, and *k*-NN classifies based on similarity. Neural Networks capture complex patterns, and BERT generates contextually accurate text representations for NLP tasks. Together, these models provide a comprehensive set of methodological strengths, ensuring a balanced evaluation of the proposed prototype tool's predictive performance.

Table 1: ML Models for Automated Context Detection.

Model	Reference
Classic	
Linear Regression (LR)	(Fisher, 1936)
Decision Tree (DT)	(Quinlan, 1986)
Random Forest (RF)	(Ho, 1995)
Multinomial Naive Bayes (MNB)	(McCallum et al., 1998)
k-Nearest Neighbour (k-NN)	(Cover and Hart, 1967)
Linear SVM	(Cortes, 1995)
Kernal SVM	(Cortes, 1995)
Advanced	
Recurrent Neural Network (RNN)	(Rumelhart et al., 1986)
Bangla BERT (BanglaBERT)	(Kowsher et al., 2022)

3.2.1 Classic Machine Learning Models

To assess feasibility, we started trials with the classic ML models from Table 1 to assess feasibility.

Preprocessing & Feature Extraction: After data cleaning, Feature Engineering identified frequent words per class (Health, Religion, Politics, Miscellaneous) using Scikit-learn and Numpy. Tokenisation and n-grams with TensorFlow segmented text and captured context using unigrams, bigrams, trigrams (Abadi et al., 2016). TF-IDF Feature Extraction assigned term importance using TfidfVectorizer (Qader et al., 2019). Finally, Label Encoding & One-hot Encoding converted labels into numerical values via Keras.
Training: The dataset was stratified to maintain class distribution (Neyman, 1992), ensuring generalizability, and then used to train and evaluate classic ML models from Scikit-learn.

3.2.2 Advanced Machine Learning Models

Building on classic ML models, we explored advanced models, specifically RNNs and BanglaBERT, to enhance context detection.
A. Recurrent Neural Network (RNN): RNNs process sequential data while maintaining a hidden state for capturing contextual dependencies.
Preprocessing & Feature Extraction: After initial preprocessing, Expanding Contractions standardized common Bangla contractions using a predefined dictionary for clarity. Train-Test Split ensured balanced class distribution across training and testing sets. Tokenisation with Keras (Chollet et al., 2015) mapped words to a word index. Data Transformation converted tokenised text into a binary matrix (word presence (1) or absence (0)). Lastly, Label Encoding transformed categorical labels into integer values & applied one-hot encoding for model compatibility.
Training: The Adam optimiser (Kingma, 2014) was employed for adaptive learning rate adjustments, ensuring faster convergence.
Model Architecture: A six-layer network was designed to handle the small dataset challenge. The Input Layer had 16 nodes to balance complexity and overfitting risk, while the Output Layer included four nodes representing Politics, Religious, Health, and Miscellaneous. Dropout Layers mitigated overfitting by randomly deactivating nodes. Activation Layers used ReLU in hidden layers for complex pattern learning and Softmax in the output layer for probability distribution (Nwankpa et al., 2018).

B. BanglaBERT: We utilized BanglaBERT (Kowsher et al., 2022), a pre-trained language model, for its superior contextual understanding and fine-tuning capabilities. Label Encoding mapped string labels (Religious, Politics, Health, and Miscellaneous) to numerical values. Text Normalization was applied during initial preprocessing (see Section 3.1) to remove inconsistencies in Bangla text. Tokenisation used the pre-trained BERT tokenizer with truncation, padding (256), and max length (256) for consistency. Data Transformation structured each row as a dictionary containing text (post content) and label (corresponding class label). Additionally, text was tokenized into token IDs and attention masks for model input.

3.3 Hyper-Parameter Tuning

Hyperparameter optimisation is crucial for improving model performance, particularly in small-scale datasets where overfitting and sensitivity to parameter changes pose challenges. We employed random search, outperforming grid search with significantly lower computational cost.

Hyperparameter optimization was essential for improving model performance, especially in small-scale datasets prone to overfitting. We used random search, outperforming grid search with a lower computational cost. The Babysitting method (Elshawi et al., 2019) further refined parameters through iterative expert-guided tuning of learning rates, regularization strength, and tree depths. For robust evaluation, *k*-fold cross-validation (Efron, 1982) ensured validation across multiple iterations, minimizing random success and maximizing data utility. Gradient Boosting (GB) (Friedman, 2001) was explored with various ML models (excluding BanglaBERT) due to its ability to handle sparse data and capture non-linear patterns. XGBoost (Chen and Guestrin, 2016) was not used, as the dataset size made it excessive.

Table 2: Hyperparameters for ML algorithms.

Hyperparameter name	ML	RNN	BanglaBERT
Batch Size	-	8	16
`epochs`	-	50,75	10,20
`max_length`	256	5000 words	512 tokens
`random_state`	0	0	42
`C` (SVMs only)	1	-	-
`gamma` (SVMs only)	scale	-	-
`learning_rate`	2e-5	1e-3	1e-5,3e-5
`Beta_1`	-	0.9	-
`Beta_2`	-	0.999	-
`dropout_rate`	-	0.2	0.3
`decay`	-	-	1e-3,1e-4

By integrating random search for efficiency, Babysitting for domain expertise, cross-validation for reliability, and Boosting for enhanced learning, we ensured the model generalized effectively while avoiding overfitting. Table 2 shows the hyperparameter values used for the analysis of ML algorithms.

3.4 Performance Evaluation

The models' ability to correctly identify the context of the Facebook post is measured in terms of accuracy ($=\frac{(TP+TN)}{(TP+FP+FN+TN)}$), recall ($=\frac{TP}{(TP+FN)}$), precision ($=\frac{TP}{(TP+FP)}$) and F1-score ($= 2 \times \frac{(Precision \times Recall)}{(Precision+Recall)}$).

Where TP are the cases where the model correctly identifies the intended context of the Facebook post, i.e. either it belongs to Health, Politics, Religion or Miscellaneous; TN represents the cases where the model correctly identifies that the input does not belong to a specific context; FN is the cases the model incorrectly identifies an input as belonging to a particular context when it does not, and FP represents the cases where the model fails to identify an input as belonging to the correct context.

4 EXPERIMENTS

All ML models were implemented in Python 3.13 using Google Colab. The dataset was split into training (75%) and testing (25%) using stratified sampling to preserve class distribution. Alternative splits (70:30, 80:20) were tested, but 75:25 yielded better results.

A *5*-fold cross-validation approach was applied to all ML algorithms (except BanglaBERT) to prevent overfitting while ensuring reliable evaluation. N-gram analysis explored unigrams, bigrams, and trigrams, with unigrams performing best; all reported results are based on unigram features. Gradient Boosting (GB) was tested with and without all ML models (except BanglaBERT) to assess its impact, with parameters `n_estimators` = [100, 200, 300], `learning_rate` = [0.01, 0.1, 0.2] and `max_depth` = [3, 5, 7]. Feature representation was evaluated using TF-IDF vectorisation and BanglaBERT embeddings for non-BanglaBERT models. BanglaBERT's hyperparameters are detailed in Table 2, and it was also tested with an 80:20 split for comparative analysis.

5 RESULTS & DISCUSSION

The ML models (Table 1) were evaluated based on average accuracy and other performance metrics. Figure 3 presents accuracy comparisons for ML models (excluding BanglaBERT) with and without Gradient Boosting (GB) using TF-IDF embeddings, while Table 3 provides additional performance metrics. The error bars in Figures 3 and 4 indicate 95% confidence intervals over five-fold cross-validation. Approximate values are used for result interpretation.

RNNs achieved the best results at 50 and 75 epochs (denoted as RNN 50 and RNN 75) after testing with various epochs. Beyond this, the models risked overfitting due to the small dataset. To address this, we also considered additional performance metrics beyond accuracy. Higher accuracy indicates better differentiation of the context across four classes.

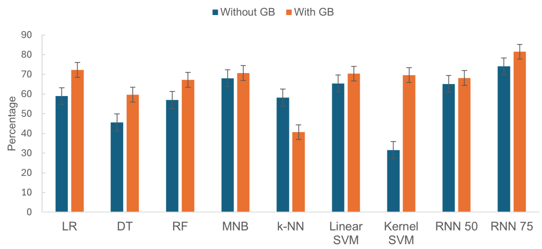


Figure 3: Average accuracy of ML models w/o and with GB.

Without Gradient Boosting (GB), RNN 75 achieved the highest accuracy (74%), followed by MNB (68%) and Linear SVM / RNN 50 (65%). LR, RF, and *k*-NN reached 60%, while DT (45%) and Kernel SVM (32%) performed the worst. With GB, Accuracy improved across models, with RNN 75 reaching 81%, LR 72%, and MNB, RF, and Linear SVM 70%. RNN 50 and DT rose to 68%, while Kernel SVM jumped from 32% to 70%, benefiting from enhanced feature separability. However, *k*-NN dropped from 58% to 41%, likely due to GB disrupting distance-based transformations, impacting classification effectiveness.

Table 3: Performance Metrics for ML models with & w/o GB.

ML Models	Precision		Recall		F1 Score	
	W/o GB	GB	W/o GB	GB	W/o GB	GB
LR	53.84	68.61	58.93	72.23	52.69	66.23
DT	48.65	68.59	45.58	59.67	44.66	57.45
RF	63.42	81.67	57.03	67.25	53.74	65.04
MNB	71.64	68.33	68.06	70.71	65.76	64.84
k-NN	61.34	55.88	58.17	40.64	57.64	34.02
Lin. SVM	61.59	64.72	65.41	70.34	59.35	64.4
Ker. SVM	50.68	74.59	31.56	69.58	19.36	70.30
RNN 50	73.00	81.00	65.00	81.00	63.00	81.00
RNN 75	68.00	75.00	68.00	74.00	66.00	73.00

Independent GB and XGBoost Testing showed GB achieved 63% accuracy, while XGBoost performed better at 69%, highlighting the effectiveness of ensemble techniques for small datasets.
RNN Performance Breakdown: While RNN 75 achieved the highest accuracy, RNN 50 exhibited better class balance across precision, recall, and F1 score. This suggests that RNN 75 prioritized overall correctness, while RNN 50 minimized misclassification in underrepresented classes. The complete comparison of precision, recall, and F1 scores is shown in Table 3. RNN 75 excelled by capturing sequential dependencies, reinforcing the advantages of deep learning for text classification in small datasets.

Figure 4 compares BanglaBERT's accuracy across 75:25 and 80:20 splits. Surprisingly, BanglaBERT underperformed compared to other ML models despite being pre-trained on Bangla text. For

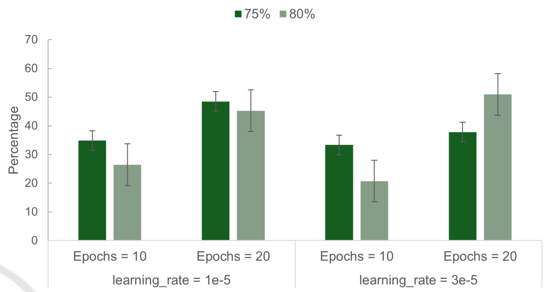


Figure 4: BanglaBERT average accuracy across two splits.

the 75:25 split, a learning rate (`lr`) of $1e{-}5$ resulted 35% accuracy (10 epochs) and 49% (20 epochs). Increasing `lr` to $3e{-}5$ dropped accuracy to 33% (10 epochs) and 38% (20 epochs), suggesting that a higher `lr` disrupted fine-tuning on a small dataset. For the 80:20 split, accuracy declined to 10 epochs from 27% (at $1e{-}5$) to 21% (at $3e{-}5$), but at 20 epochs, it improved from 45% to 51%, indicating that BanglaBERT benefits from training on a small dataset. Given its inconsistencies, Gradient Boosting was not applied, as underfitting limited its contribution to the meta-learner.

Table 4: BanglaBERT performance across two splits.

Parameters	Precision		Recall		F1-Score	
	75%	80%	75%	80%	75%	80%
`learning_rate=1e-5`						
Epochs = 10	58.04	6.60	32.50	25.00	31.00	10.45
Epochs = 20	65.13	40.49	55.47	42.14	46.27	39.40
`learning_rate=3e-5`						
Epochs = 10	20.09	10.67	28.32	26.07	19.40	12.68
Epochs = 20	69.55	39.70	36.35	50.36	30.30	43.32

Comparison with Related Work- A previous study by (Chakma and Hasan, 2023) focused on sentiment analysis (32k samples, three-class classification) and achieved an F1 score of 72%. Our study, however, focuses on fact-checking, making direct comparisons difficult. Despite our significantly smaller dataset, we achieved an F1 score of 66% with RNN 75, demonstrating its potential in this domain.

To evaluate different feature representation techniques, we tested TF-IDF vectorization and BanglaBERT embeddings with Gradient Boosting (Table 5). Overall, TF-IDF outperformed BanglaBERT embeddings across all models except k-NN. The underperformance of BanglaBERT embeddings is likely due to insufficient fine-tuning of a small dataset. However, k-NN benefited from BanglaBERT embeddings, as dense semantic representations improved its nearest-neighbor calculations.

Table 5: Accuracy - TF-IDF & BanglaBERT Embeddings.

Hyperparameter	TF-IDF	BanglaBERT
LR	72.23	63.90
DT	59.67	47.51
RF	67.25	63.88
MNB	70.71	62.72
k-NN	40.64	54.35
Linear SVM	70.34	63.88
Kernel SVM	69.58	65.02
RNN 50	68.18	53.16
RNN 75	81.48	51.90

6 LIMITATIONS

The study faced two key challenges. First, reliance on LLMs (ChatGPT) for paraphrasing introduced hallucinations, generating irrelevant or inaccurate content that sometimes compromised contextual integrity.

Second, dataset expansion was constrained by limited human resources for categorization and validation, affecting scalability. Addressing them requires better LLM accuracy, automated quality checks, and larger annotation teams to enhance dataset reliability.

Despite these challenges, preprocessing and refinement improved consistency and readability, ensuring the dataset's suitability for analysis.

7 CONCLUSION

Our research addresses a critical gap in fact-checking Bangla Facebook posts by developing an automated context detection tool for this low-resource language. Unlike previous studies on sentiment analysis, our approach is tailored for context detection, a key step in fact-checking. Using a small Bangla dataset, we demonstrate that TF-IDF embeddings outperform BanglaBERT embeddings, highlighting the need for larger, diverse datasets to improve performance.

Future work will incorporate opinion detection, category classification, priority ranking, and advanced prompt engineering. Collaboration with professional fact-checkers will assess real-world effectiveness, while further research will explore automated solutions for class imbalance and data validation to enhance scalability and robustness.

ACKNOWLEDGEMENTS

Kanij is supported by ARC Laureate Fellowship FL190100035.

REFERENCES

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., et al. (2016). Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*.

Ali, M. S. (2020). Uses of facebook to accelerate violence and its impact in bangladesh. *Global Media Journal*, 18(36):1–5.

Baly, R., Mohtarami, M., Glass, J., Màrquez, L., Moschitti, A., and Nakov, P. (2018). Integrating stance detection and fact checking in a unified corpus. *arXiv preprint arXiv:1804.08012*.

Barbera, D. L., Roitero, K., Mizzaro, S., Nozza, D., Passaro, L., and Polignano, M. (2023). Combining human intelligence and machine learning for fact-checking: Towards a hybrid human-in-the-loop framework. *Intelligenza Artificiale*, 17(2):163–172.

Carey, J. M., Guess, A. M., Loewen, P. J., Merkley, E., Nyhan, B., Phillips, J. B., and Reifler, J. (2022). The ephemeral effects of fact-checks on covid-19 misperceptions in the united states, great britain and canada. *Nature Human Behaviour*, 6(2):236–243.

Chakma, A. and Hasan, M. (2023). Lowresource at blp-2023 task 2: Leveraging banglabert for low resource sentiment analysis of bangla language. *arXiv preprint arXiv:2311.12735*.

Chan, M.-p. S., Jones, C. R., Hall Jamieson, K., and Albarracín, D. (2017). Debunking: A meta-analysis of the psychological efficacy of messages countering misinformation. *Psychological science*, 28(11):1531–1546.

Chen, T. and Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794.

Chollet, F. et al. (2015). Keras. https://github.com/fchollet/keras.

Cortes, C. (1995). Support-vector networks. *Machine Learning*.

Cover, T. and Hart, P. (1967). Nearest neighbor pattern classification. *IEEE transactions on information theory*, 13(1):21–27.

Dickey, C. (2019). The rise and fall of facts. *Columbia Journalism Review*, 23.

Dixon, S. J. (2024). Leading countries based on facebook audience size as of december 2024. *Statista*. Accessed: 2024-12-31.

Efron, B. (1982). *The jackknife, the bootstrap and other resampling plans*. SIAM.

Elshawi, R., Maher, M., and Sakr, S. (2019). Automated machine learning: State-of-the-art and open challenges. *arXiv preprint arXiv:1906.02287*.

Eysenbach, G. et al. (2020). How to fight an infodemic: the four pillars of infodemic management. *Journal of Medical Internet Research*, 22(6):e21820.

Fabry, M. (2017). Here's how the first fact-checkers were able to do their jobs before the internet. *Time (24 de agosto). Recuperado de https://bit. ly/35FEdnA*.

Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. *Annals of eugenics*, 7(2):179–188.

Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232.

Guo, X., Yin, Y., Dong, C., Yang, G., and Zhou, G. (2008). On the class imbalance problem. *Fourth International Conference on Natural Computation, ICNC '08*.

Hanselowski, A. (2020). *A machine-learning-based pipeline approach to automated fact-checking*. Phd thesis, Technische Universität Darmstadt, Darmstadt, Germany. Available at https://doi.org/10.25534/tuprints-00014136.

Haque, M. M., Yousuf, M., Alam, A. S., Saha, P., Ahmed, S. I., and Hassan, N. (2020). Combating misinformation in bangladesh: Roles and responsibilities as perceived by journalists, fact-checkers, and users. *Proceedings of the ACM on Human-Computer Interaction*, 4(CSCW2):1–32.

Haque, M. M., Yousuf, M., Arman, Z., Rony, M. M. U., Alam, A. S., Hasan, K. M., Islam, M. K., and Hassan, N. (2018). Fact-checking initiatives in bangladesh, india, and nepal: a study of user engagement and challenges. *arXiv preprint arXiv:1811.01806*.

Hitlin, P. and Olmstead, K. (2018). The science people see on social media. pew research center. *Science & Society. https://pewrsr*.

Ho, T. K. (1995). Random decision forests. In *Proceedings of 3rd international conference on document analysis and recognition*, volume 1, pages 278–282. IEEE.

Hossain, B., Muzykant, V., and Md, N. (2022). Roles of fact-checking organizations in bangladesh to tackle fake news. *Law and Authority*, (5):3–8.

Jachai, O. (2017). Jachai organisation. https://www.jachai.org/ [Accessed: (November, 2024)].

Kingma, D. P. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kotsiantis, S. B., Zaharakis, I. D., and Pintelas, P. E. (2006). Machine learning: a review of classification and combining techniques. *Artificial Intelligence Review*, 26:159–190.

Kowsher, M., Sami, A. A., Prottasha, N. J., Arefin, M. S., Dhar, P. K., and Koshiba, T. (2022). Bangla-bert: transformer-based efficient model for transfer learning and language understanding. *IEEE Access*, 10:91855–91870.

Kreps, S. E. and Kriner, D. L. (2022). The covid-19 infodemic and the efficacy of interventions intended to reduce misinformation. *Public Opinion Quarterly*, 86(1):162–175.

McCallum, A., Nigam, K., et al. (1998). A comparison of event models for naive bayes text classification. In *AAAI-98 workshop on learning for text categorization*, volume 752, pages 41–48. Madison, WI.

Minar, M. R. and Naher, J. (2018). Violence originated from facebook: A case study in bangladesh. *arXiv preprint arXiv:1804.11241*.

Naher, J. and Minar, M. R. (2018). Impact of social media posts in real life violence: A case study in bangladesh. *arXiv preprint arXiv:1812.08660*.

Neyman, J. (1992). On the two different aspects of the representative method: the method of stratified sampling and the method of purposive selection. In *Breakthroughs in statistics: Methodology and distribution*, pages 123–150. Springer.

Nwankpa, C., Ijomah, W., Gachagan, A., and Marshall, S. (2018). Activation functions: Comparison of trends in practice and research for deep learning. *arXiv preprint arXiv:1811.03378*.

Qader, W. A., Ameen, M. M., and Ahmed, B. I. (2019). An overview of bag of words; importance, implementation, applications, and challenges. In *2019 international engineering conference (IEC)*, pages 200–204. IEEE.

Quinlan, J. R. (1986). Induction of decision trees. *Machine learning*, 1:81–106.

Roose, K. (2021). What is qanon, the viral pro-trump conspiracy theory. *The New York Times*, 3.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, 323(6088):533–536.

Rumour Scanner Bangladesh, O. (2020). Rumour scanner bangladesh. https://rumorscanner.com/ [Accessed: (September, 2023 and November, 2024)].

Stencel, M., Luther, J., and Ryan, E. R. (2021). Fact-checking census shows slower growth. *Poynter. https://www.poynter.org/fact-checking/2021/fact-checking-census-shows-slower-growth/*.

Thorne, J. and Vlachos, A. (2018). Automated fact checking: Task formulations, methods and future directions. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3346–3359.

Tollefson, J. (2021). How trump turned conspiracy theory research upside down. *Nature*, 590(11):192–193.

Vlachos, A. and Riedel, S. (2014). Fact checking: Task definition and dataset construction. In *Proceedings of the ACL 2014 workshop on language technologies and computational social science*, pages 18–22.