Vol. 06, No. 01, pp. 43 –49 (2025)
ISSN: 2708-0757

JOURNAL OF APPLIED SCIENCE AND TECHNOLOGY TRENDS

www.jastt.org

Text Classification for Clickbait Detection: A Model-Driven Approach Using CountVectorizer and ML Classifiers

Nay Oo Lwin[1], Rituraj Jain*, Rabnaz Dal, Htet Oo Yan, Kaung Khant Thaw, Saw Yan Naung

[1] *Department of Information Technology, Marwadi University, Rajkot, Gujarat, India,*
nayoolwin.115714@marwadiuniversity.ac.in, jainrituraj@yahoo.com, rabnaz.dal118547@marwadiuniversity.ac.in, htetooyan.118918@marwadiuniversity.ac.in, kaungkhantthaw.115775@marwadiuniversity.ac.in, sawyannaung.115689@marwadiuniversity.ac.in
*Correspondence: *jainrituraj@yahoo.com*

Abstract

In both digital and mainstream media, clickbait headlines are rampant forms of receiving and distributing attention, designed to maximize click through rates which in turn delegitimize content and help spread misinformation. In this paper we propose an efficient and interpretable machine learning framework for the binary classification of clickbait vs non-clickbait headlines using traditional models. Our pipeline involves the preprocessing step, followed by n-gram feature extraction via CountVectorizer and the classification done using Multinomial Naive Bayes, Logistic Regression and XGBoost. The models were trained and evaluated on accuracy, precision, recall, F1 score and ROC AUC, using a publicly available dataset. Our results indicate that Naive Bayes and Logistic Regression models enjoyed better performance with an accuracy of 95.88% and an F1-score of 95.88% and an AUC of 0.99, performing better than the more complex XGBoost classifier. Confirming the ability of lightweight models for real time clickbait detection, we further show that traditional machine learning is also interpretable and scalable.

Keywords: *Clickbait Detection, Text Classification, CountVectorizer, Machine Learning, Naive Bayes Classifier*
Received: April 16th, 2025 / Revised: June 16th, 2025 / Accepted: June 23rd, 2025 / Online: June 24th, 2025

I. INTRODUCTION

The sensationalized and misleading content is spreading at a higher rate just in the form of a clickbait headline. Unrealistic clickbait headlines contain driving forces designed to play on people's curiosity and drum up engagement, but very often do not correlate to content, erode user trust and incite misinformation propagation [1–3]. Social platforms and online news outlets need to particularly address this problem; accuracy and integrity of information matter [4]. Deep learning-based approaches for text classification tasks have shown promising performance, but with much increased computational overhead, reduced interpretability and limited scalability for use in real–time or resource limited contexts [5].

This study aims at creating an interpretable, robust and lightweight framework for the binary classification of clickbait and non—clickbait headlines through traditional machine learning algorithms. A multi stage pipeline was practiced from data acquisition, preprocessing, count vectorizer based feature extraction to Multinomial Naïve Bayes, logistic regression and XGBoost model development. We ran our 6,400 sample held out test set through the models and did accuracy, precision, recall, F1 score, ROC-AUC and confusion matrix analysis.

Experimental results reveal that both Naive Bayes and Logistic Regression models achieved an accuracy and F1-score of 95.88% and 0.96, respectively, outperforming XGBoost, which recorded an F1-score of 0.87. Furthermore, ROC analysis confirmed near-perfect AUC values of 0.99 for the linear models, highlighting their suitability for high-dimensional, sparse text data. This research demonstrates the potential of classical algorithms, enhanced by n-gram features, to deliver effective and interpretable clickbait detection in real-world applications.

II. LITERATURE REVIEW

Text classification techniques have been extensively utilized to identify deceptive or manipulative content such as clickbait, fake news, and cyberbullying across digital and social media platforms. In [6] authors explored transformer-based models—BERT, RoBERTa, XLNet, DistilBERT, and GPT-2.0—for

doi: 10.38094/jastt61237

cyberbullying detection, with BERT achieving an F1-score of 95% and RoBERTa reaching the highest accuracy of 96% on the Tweeteval dataset. Their work highlighted the critical trade-off between performance metrics and inference efficiency, especially in real-time deployment scenarios. In the field of clickbait detection, authors of [7] benchmarked several classifiers including Support Vector Machines (SVM), Random Forest (RF), and BERT variants. Among these, the CKIP-BERT model, pre-trained by Yang and Ma, recorded superior performance with F1-scores of 0.887 for binary and 0.918 for multi-class classification in the context of Taiwan news headlines. Several researchers have explored topic modeling to enhance clickbait detection. Authors in [8] integrated Latent Dirichlet Allocation (LDA), Latent Semantic Analysis (LSA), and BERTopic into feature engineering pipelines, boosting classification performance across social platforms like YouTube and Instagram. Similarly, authors of [9] employed these topic modeling techniques to extract latent thematic features and reported recall values up to 90%, demonstrating their utility in identifying clickbait beyond surface-level patterns.

Efforts in low-resource languages have yielded notable progress. In [10] authors developed a CNN-based model using fastText embeddings for Amharic clickbait detection, achieving an accuracy of 94.27% and an F1-score of 94.24%. Authors of [11] addressed Bangla clickbait using an ensemble of transformer models enriched with linguistic, sentiment, and semantic features, reaching an accuracy of 96%. Further innovations include prompt-based strategies. A two-stage summarization-enhanced prompt-tuning method was introduced [12] to bridge semantic gaps between headlines and articles, achieving state-of-the-art results. A prompt-tuning model proposed in [13] trained only on few-shot labeled titles, attaining 95% accuracy across multiple benchmarks.

BERT-based ensemble approaches were also explored by [14] in the Indonesian context, where a BERT + CNN model achieved a precision of 0.91 and an F1-score of 0.89. Hebrew headlines examined using machine learning models and linguistic analysis, yielding an accuracy of 0.87 and emphasizing the importance of language-specific feature tuning [15].

Several datasets have recently enriched this domain. BaitBuster-Bangla, a multimodal Bangla dataset of over 253,000 samples across 18 feature categories introduced in [16]. Authors in [17] released RoCliCo, a Romanian dataset of 8,313 annotated samples, and proposed a contrastive Ro-BERT model that achieved an F1-score of 0.8852. In [18] authors compiled a 20,896-article Chinese dataset to train the CA-CD model using contextual representation and part-of-speech tags. Verbo-visual elements in Arabic YouTube thumbnails was examined in [19], using visual grammar and meta-discourse frameworks to analyze multimodal clickbait cues.

III. Methodology

This section elaborates on the complete pipeline designed for the classification of clickbait headlines using traditional machine learning approaches. The methodology consists of five major stages: dataset acquisition, preprocessing, feature representation, model development with architectural specifications, and model evaluation. A comprehensive system architecture has been developed to streamline the entire process from raw data ingestion to final prediction and visualization. Figure 1 show the system architecture for the proposed system.

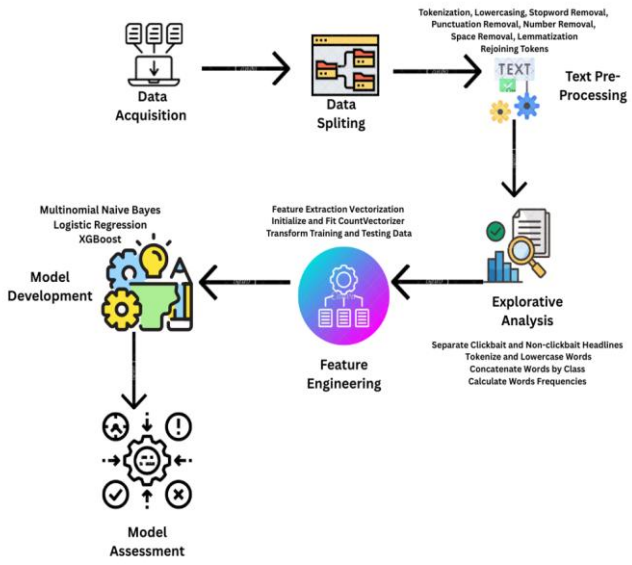


Fig. 1. System Flow Architecture of the Proposed System.

A. Dataset Description

The dataset used in the present research is the file called clickbait_data.csv, which contains 32,000 headlines in the English language, with two columns: 1 denotes clickbait, and 0 denotes a non-clickbait [20]. This ideally organized database was obtained as a publicly shared submission that is quite common and extended in the study of benchmarks related to headline classes of duties. It includes two major variables, the headline with the text and the clickbait label, which indicates the binary classification target. By using the statistical analysis of the classes distribution, we see that the data is as near to having perfect balance as possible because it contains 16,001 non-clickbait headlines or 50.003% of the total records, in comparison to 15,999 clickbait headlines or 49.997%. This unbiasedness character makes the classification models be trained without necessarily using resampling methods and helps in making learning unbiased. Considering the large size of the dataset, its clean format and an equal distribution of the classes contained in it, the dataset proves to be optimal to carrying out supervised machine learning and makes the proposed classification structure robust and generalizable.

B. Data Preprocessing

Prior to model training, the headline text data underwent a series of preprocessing steps to prepare it for feature extraction. These steps were applied sequentially to both the training and testing sets to prevent data leakage and ensure a consistent transformation pipeline.

1) Tokenization: Splitting headlines into individual words or tokens. I split the text on white space to accomplish this.

2) Lowercasing: The lowercase form of all tokens were performed to ensure words with different capitalization are tokenized as one token (e.g., 'Headline' and 'headline').

3) Stopword Removal: English stopwords were removed using the NLTK list of stopwords. In practice such words are usually assumed to hold little semantic meaning for classification tasks.

4) Punctuation Removal: All tokens had their punctuation characters stripped from them such as periods, commas and question marks, using string.punctuation set.

5) Number Removal: All numerical digit were removed from tokens.

6) Space Removal: Trailing and leading environment from tokens was removed.

7) Lemmatization: Tokens were lemmatized by the means of the WordNetLemmatizer from NLTK. Lemmatization is a reduction of words to a base or dictionary form (e.g., from "running," to "run"), that is useful to reduce vocabulary size and cluster related words.

8) Rejoining Tokens: Then using processed tokens for each headline, these were rejoined back to a single string separated with spaces. Then I used this formatted text to extract features.

Figure 2 visualizes the distribution of headline lengths (in terms of word count) for clickbait and non-clickbait samples. It highlights that clickbait headlines tend to be shorter and more uniform in length, typically peaking around 6–9 words, whereas non-clickbait headlines have a wider spread and longer average length.

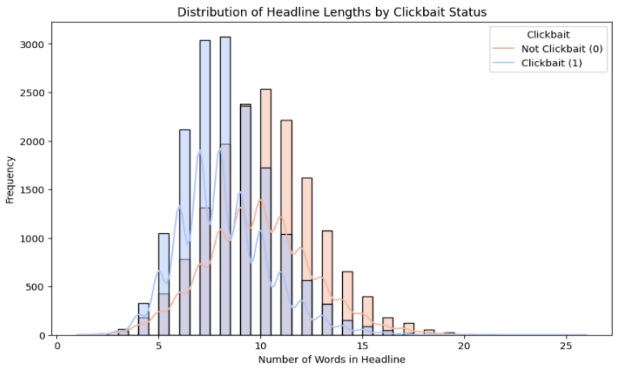


Fig. 2. Distribution of Headline Lengths by Clickbait Status

The data was split into training and testing sets on an 80/20 ratio. A split was performed using a random_state of 42 so as to ensure reproducibility.

C. Feature Representation

The machine learning models were prepared using the preprocessed text data, using the CountVectorizer function from the skicit learn package to prepare it. The ability to transform a film into acclaim and attention is described through this technique. After that, for every document (headline) in the vocabulary, a vector is allocated, each dimension corresponding to a keyword and its frequency in the associated document, this mapping due to the vocabulary.

For different models we have used two different CountVectorizer configurations.

- Naive Bayes and XGBoost: A CountVectorizer were initialized with analyzer = 'word', ngram_range = (1, 2) and max_features = 10000 for Naive Bayes and XGBoost. During testing, this configuration was used to extract both unigrams (single words) and bigrams (sequences of two words) and restrict the vocabulary size to the top 10,000 most frequent terms.
- Logistic Regression: We initialized a separate CountVectorizer with analyzer = 'word', ngram_range = (1, 2), max_features = 22500. This configuration extracted unigrams and bigrams as well, but had a larger vocabulary size of 22,500 features. During experimentation, we used a different feature set for logistic regression that we observed empirically would be useful.

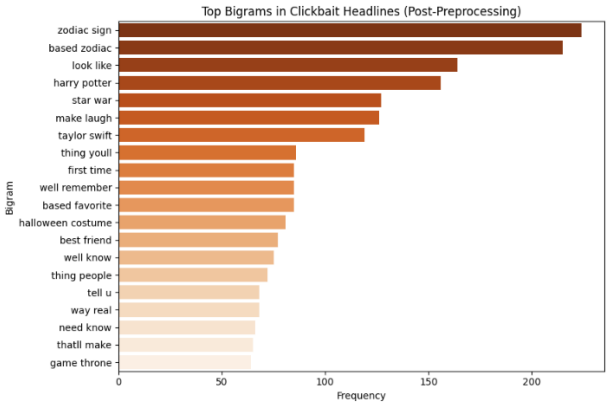
The training data vectorizer was fit and applied to both the training and testing data, transforming them to sparse arrays which were transformed to dense arrays for model input.

In order to gain a clearer insight into the lexical pattern contained in dataset, Figure 3 shows the largest 20 bigrams in both clickbait and non-clickbait headlines following preprocessing. The clickbait genre stands out as having first the emotionally charged or curiosity-generating phrases like zodiac sign or look like and commonly using factual-oriented phrases like New York or prime minister in non- clickbait headlines.

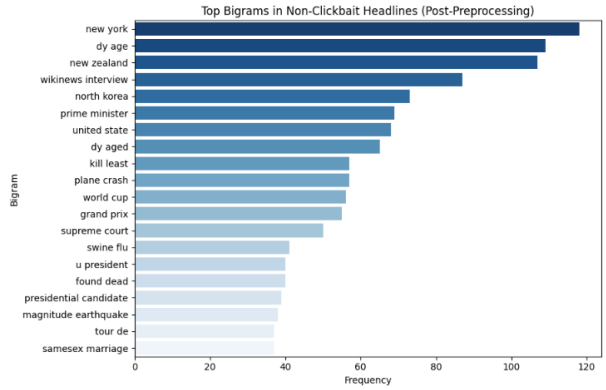
Along with the visual frequency analysis, we carried out the additional research on interdependence feature and feature correlation in order to make the models more interpretable. Pearson correlation matrices have been computed between target label (clickbait) and individual features. The features which were positively correlated were highly correlated to clickbait headlines, and the negatively correlated were prevalent in the non-clickbait content. Also, in order to measure features redundancy, the correlations among features in the high-dimensional vector space were tested. Although models such as XGBoost are robust to multicollinearity, other models like Logistic Regression may be affected by the redundancies that exist, which may indeed interfere with interpretability and performance. Even though dimensional constraints preclude visualizing the entire correlation matrix, this analysis confirmed the ability to discriminate as well as consistency of the lexical features that appeared in the models.

D. Model Architectures

Three traditional yet powerful machine learning models were deployed to classify the headlines as either 'clickbait' or 'non-clickbait'. The selection was based on their demonstrated effectiveness in various text classification tasks, computational efficiency, and ability to provide insights into the classification process. Each model was trained independently on the prepared feature sets. Table I show all the details for these models.



(a)



(b)

Fig. 3. Top Bigrams in (a) Clickbait vs (b) Non-Clickbait Headlines

1. Multinomial Naive Bayes (MNB) is a probabilistic classification algorithm based on the Naive Bayes theorem. Classification with discrete features is particularly well fit by XGBoost, as it is the CountVectorizer vector which creates word counts or frequencies. Instead the model uses the probabilities of individual words appearing in headlines belonging to that class ('clickbait' or 'non-clickbait') to calculate the probability of a given headline belonging to one of these classes. It also presents the assumption that the presence of a word in a headline is independent of other words in the headline, given the class label (which leads to the 'naive' component of the approach). We used implementation from sklearn.naive_bayes.MultinomialNB with default parameters.

2. The Logistic Regression (LR) is a widely used linear model for binary classifier. Given a linear combination of the input features, it predicts the probability of a binary outcome (in that case of clickbait or not). A sigmoid function is then applied once again to the output of the linear combination, ensuring the output sits between 0 and 1 — which is interpreted as indicating a probability. In classification, these probabilities are class assigned using a threshold (typically 0.5). We used the implementation from sklearn.linear_model.LogisticRegression with default parameters. The feature set used were based on CountVectorizer with max_features = 22500 and ngram_range = (1,2), this model was trained on.

3. Gradient boosting framework with a nice implementation called Extreme Gradient Boosting (XGBoost) being highly efficient and flexible. It is a method of ensemble learning which constructs multiple decision trees sequentially. Each new tree tries to correct prediction errors in the already build trees. It does this regularization to prevent overfit and works with sparse data (which is what we see often with text vectorization). For this part, we used xgboost.XGBClassifier implementation with n_estimators = 100, eval_metric = 'logloss' and most other hyperparameters set to default. Then we trained this model with a feature set produced using a CountVectorizer with max_features = 10000 and ngram_range = (1,2).

TABLE I. MODEL ARCHITECTURE SUMMARY

Model	Type	Implementation	Core Mechanism
Multinomial Naive Bayes	Probabilistic Classifier	sklearn.naive_bayes	Multinomial Distribution
Logistic Regression	Linear Classifier	sklearn.linear_model	Sigmoid Activation
XGBoost Classifier	Ensemble /Boosting	xgboost.XGBClassifier	Gradient Boosted Decision Trees

E. Cross-Validation Strategy

In order to ensure the strength of our model evaluation, we applied 5-fold cross-validation to both our Naive Bayes and logistic regression classifiers through training data. This is a more general strategy to overcome the risk of overfitting of a model to a particular partition of the data and to give a better estimate of the performance of the model. Means of the F1-scores obtained during the five folds were 0.9581 and 0.9495, over Naive Bayes and Logistic Regression, respectively. These findings prove that the two models were stable and consistent in their classification of the provided data.

F. Hyperparameter Configuration

To initialize each model, the following configuration was selected, as shown in Table II. This study had not implemented more advanced hyperparameter optimization mechanisms like grid search or a random search. Rather, we used standard or empirical values derived in typical best practices to prevent confusion and assure the ability to reproduce the results during baseline analysis. Naive Bayes (MultinomialNB) was set with the default additive smoothing parameter alpha = 1.0, that works reasonably well in text classification with count-based features. The Logistic Regression model was initialized with penalty = l2, C = 1.0 and solver = lbfgs. Whereas the max_iter parameter was pushed to convergence in the process of cross-validation, the setting was kept at the default during the main model used in evaluation. XGBoost was defined using the parameter settings of n_estimators = 100 and eval_metric = logloss but other parameters of mount depth and learning rate were kept on default as provided by the xgboost library. The configuration protocol enabled us to make a reasonable comparison of the inbuilt learning ability in each model using a standardized configuration without bias of tuning locally available in the dataset. Although we realize that hyperparameter tuning can be

of significant importance to the performance, especially in models, such as XGBoost or Logistic Regression, this sort of optimization was not considered in the framework of our initial investigation as it requires too much time to be calculated. It shall be improved in future work that uses more systematic tuning procedures to optimize models and extend validity of generalizability.

TABLE II. HYPERPARAMETER SETTINGS

Model	Key Hyperparameters
MultinomialNB	alpha=1.0, fit_prior=True (default)
Logistic Regression	penalty='l2', solver='lbfgs', max_iter=100
XGBoost Classifier	n_estimators=100, eval_metric='logloss', use_label_encoder=False

G. Evaluation Metrics

The performance of the suggested classifying models was thoroughly tested by use of the conventional assessment measures as applied in previous investigations. Accuracy, the total correctness of the model, was determined by taking the ratio of the correct predicted samples to the total number of samples. In addition, we used the F1 score (a harmonic mean of recall and precision) as an actual performance ranking metric for cases where the class distribution changes or the cost of a false positive is greater than the cost of a false negative. Finally, in another section, we examined the confusion matrix to see how distributed are each of the four types of misclassification errors – namely, true positive, false positive, true negative and buried pattern. It gave us a better view of patterns. In particular, we built Receiver Operating Characteristic (ROC) curves for each of the models and measured their 'ability to discriminate performance' across varying thresholds through the Area Under the Curve (AUC). Consequently, the model was tested after on a simulated dataset and results were used to generalize and validate the model. We used a single graph containing the ROC curves of all of the models to compare them visually to see how discriminative they are.

IV. RESULT AND DISCUSSION

This paper evaluates and compares the three machine learning models used for clickbait headline classification (Multinomial Naive Bayes, Logistic Regression and XGBoost Classifier) in detail. Each model was tested using a held-out test set comprising 6,400 samples. The performance of models were measured with multiple standard metrics: accuracy, precision, recall, F1-score, confusion matrices and ROC-AUC curves.

A. Quantitative Performance Comparison

The primary classification performance metrics of each model are summarized in Table III. As can be seen, MNB and LR obtained the same accuracy of 95.88%, while XGBoost was just a little behind at 87.72%. Also, MNB and LR had the same macroaveraged precision, recall and F1 score (0.96) which indicated a matchless ability to trap the tradeoff between false positives and false negatives. Looking at the other hand, XGBoost scored slightly lower precision (0.89), recall (0.88) and F1-score (0.87) which showing poorer generalization on this particular text classification task.

TABLE III. COMPARATIVE EVALUATION OF CLASSIFICATION MODELS

Model	Accuracy (%)	Precision	Recall	F1-Score
Multinomial Naive Bayes	95.88	0.96	0.96	0.96
Logistic Regression	95.88	0.96	0.96	0.96
XGBoost Classifier	87.72	0.89	0.88	0.87

B. Statistical Significance Testing

In order to further prove the gap in performance between models, we ran the paired-t-test on the F1-scores of results of cross-validation of Naive Bayes and Logistic Regression. The t-statistic value thus obtained was 6.4533 and the corresponding p-value was 0.0030 which is statistically significant as $p < 0.05$. This conclusion implies that Naive Bayes has a more optimized and stable effect of being better than Logistic Regression in the assessed experiment environment.

C. Confusion Matrix Analysis

Figure 4 illustrates the confusion matrices for all three models, enabling an in-depth analysis of the type of errors committed.

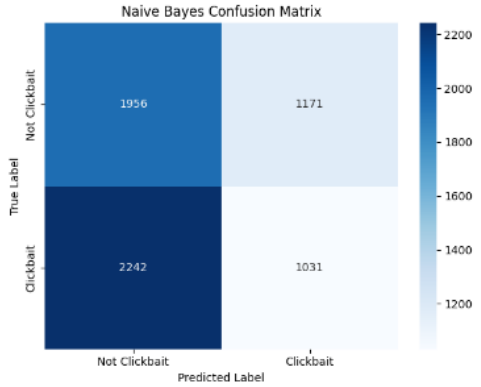
- Naive Bayes struggled with distinguishing clickbait from non-clickbait headlines, especially in cases with subtle semantic differences. It misclassified 1,171 non-clickbait samples as clickbait and 1,031 clickbait samples as non-clickbait.
- Logistic Regression demonstrated the most balanced classification performance, with very low false positives and false negatives: only 108 non-clickbait headlines were misclassified as clickbait, and 185 clickbait headlines were misclassified as non-clickbait.
- XGBoost, despite its complexity, underperformed in identifying clickbait headlines, misclassifying 656 clickbait instances, though it retained decent accuracy in recognizing non-clickbait headlines.

D. ROC Curve and AUC Analysis

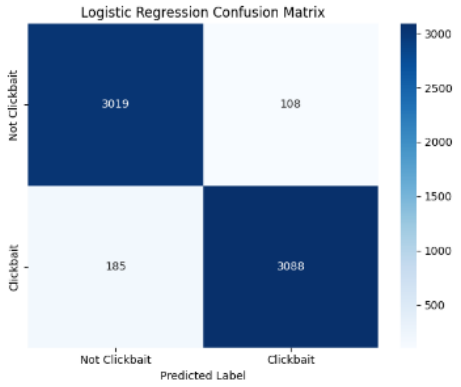
The Receiver Operating Characteristic (ROC) curves and Area Under the Curve values of the three models are shown in Figure 5.

- Naive Bayes and Logistic Regression both attained near perfect AUC of 0.99 and both demonstrate excellent discriminatory ability.
- XGBoost, in contrast, recorded an AUC of 0.96, which is respectable but slightly lower than the other two models.

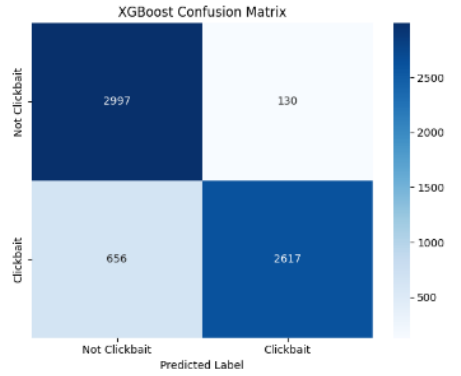
These results suggest that although XGBoost is typically strong in structured data, it may not be optimally suited for sparse, high-dimensional representations derived from text data without careful feature engineering or hyperparameter tuning.



(a)



(b)



(c)

Fig. 4. Confusion Matrices of (a) Naive Bayes, (b) Logistic Regression, and (c) XGBoost

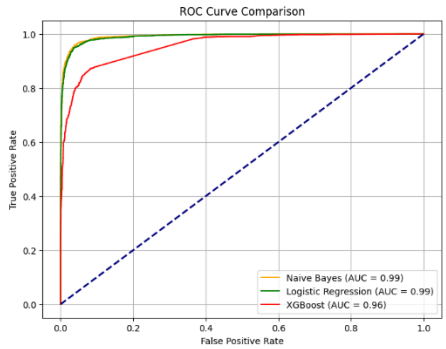


Fig. 5. ROC Curve Comparison of All Models

E. Comparative Evaluation with Existing Approaches

To benchmark the proposed system against existing approaches, we compared its performance with state-of-the-art methods as given in Table IV. While transformer-based models such as BERT, RoBERTa, and CKIP-BERT demonstrate strong performance in various languages and contexts, our Naive Bayes and Logistic Regression models achieve a competitive F1-score of 0.96 on English clickbait data. These results underscore the efficiency and applicability of traditional machine learning models when combined with effective preprocessing and feature engineering, particularly for real-time or resource-constrained environments.

TABLE IV. COMPARISON WITH EXISTING STATE-OF-THE-ART APPROACHES

Ref	Language / Context	Model / Technique	Accuracy (%)	F1-Score
[6]	English (Cyberbullying)	BERT, RoBERTa	96	0.95
[7]	Chinese (Taiwan Headlines)	CKIP-BERT (Binary), CKIP-BERT (Multi-class)	—	0.887, 0.918
[10]	Amharic	CNN + fastText embeddings	94.27	0.942
[11]	Bangla	Transformer ensemble + enriched features	96	—
[13]	English	Few-shot prompt tuning	95	—
[14]	Indonesian	BERT + CNN Ensemble	—	0.89
[15]	Hebrew	ML + linguistic tuning	87	—
[17]	Romanian	Contrastive Ro-BERT	—	0.885
Proposed System	English (General Headlines)	Naive Bayes + CountVectorizer	95.88	0.96
		Logistic Regression + CountVectorizer	95.88	0.96
		XGBoost + CountVectorizer	87.72	0.87

F. Discussion

The results demonstrate that traditional linear models—specifically, Multinomial Naive Bayes and Logistic Regression—are well-suited for text classification tasks such as clickbait detection, especially when coupled with n-gram-based frequency features. Their consistent and robust performance indicates that simpler models can often outperform complex ensemble methods like XGBoost in natural language processing tasks when the feature representation is appropriate.

Furthermore, Logistic Regression offers the added advantage of interpretability through its model coefficients, while Naive Bayes remains computationally efficient for real-time applications. On the other hand, XGBoost may require more extensive tuning and possibly the integration of TF-IDF or deep learning-based embeddings to reach competitive performance levels in text classification.

V. CONCLUSION

In this paper we address the increasing concern with the proliferation of clickbait in the digital media space by proposing a model driven approach to headline classification using classic machine learning algorithms. The proposed system integrates CountVectorizer based n-gram feature extraction with classical classifiers Multinomial Naive Bayes, Logistic Regression and XGBoost, showing strong performance but low computation complexity and, for the case of the classical classifiers, model transparency. Logistic Regression (as well as Naive Bayes) achieved an identical and excellent result with an accuracy of 95.88% and an F1 score of 95.88%, confirming high performance for high dimensional, sparse text classification tasks.

Simple, interpretable models such as our logistic regression model as well as naive models are experimentally found to work well for clickbait detection — better than more complex ensembles such as XGBoost — in the setting of vectorized text features. Although the models are easy to deploy in real time, we learn from using them how to moderate by inspecting the decision boundaries and how to prioritize different features in media forensics.

We will continue that research by extending the model to multilingual datasets, incorporating semantic embeddings (Word2Vec, BERT, etc.) and using parameter optimization to optimize for performance. Alongside, qualitative error analysis and adversarial robustness testing can improve the reliability of the model against different varying and emerging clickbait strategies. Based on the proposed framework, a practical basis for scalable and explainable clickbait detection in real world digital ecosystems is provided.

REFERENCES

[1] Y.-W. Ma, L.-D. Chen, Y.-M. Huang, and J.-L. Chen, "Intelligent Clickbait News Detection System Based on Artificial Intelligence and Feature Engineering," IEEE Transactions on Engineering Management, vol. 71, pp. 12509–12518, Jan. 2024, doi: 10.1109/tem.2022.3215709.

[2] A. Diez-Gracia, D. Palau-Sampio, P. Sánchez-García, and I. Sánchez-Sobradillo, "Clickbait Contagion in International Quality Media: Tabloidisation and Information Gap to Attract Audiences," Social Sciences, vol. 13, no. 8, p. 430, Aug. 2024, doi: 10.3390/socsci13080430.

[3] C. Reuter, A. Lee Hughes, and C. Buntain, "Combating information warfare: state and trends in user-centred countermeasures against fake news and misinformation," Behaviour & Information Technology, vol. ahead-of-print, no. ahead-of-print, pp. 1–14, Dec. 2024, doi: 10.1080/0144929x.2024.2442486.

[4] D. Surjatmodjo, A. A. Unde, A. F. Sonni, and H. Cangara, "Information Pandemic: A Critical Review of Disinformation Spread on Social Media and Its Implications for State Resilience," Social Sciences, vol. 13, no. 8, p. 418, Aug. 2024, doi: 10.3390/socsci13080418.

[5] A. Shrestha, A. Behfar, and M. N. Al-Ameen, "'It is Luring You to Click on the Link With False Advertising' - Mental Models of Clickbait and Its Impact on User's Perceptions and Behavior Towards Clickbait Warnings," International Journal of Human–Computer Interaction, vol. 41, no. 4, pp. 2352–2370, Mar. 2024, doi: 10.1080/10447318.2024.2323248.

[6] A. G. Philipo, D. S. Sarwatt, J. Ding, M. Daneshmand, and H. Ning, "Assessing text classification methods for cyberbullying detection on social media platforms," *arXiv (Cornell University)*, Dec. 2024, doi: 10.48550/arxiv.2412.19928.

[7] Y.-L. Lin, S.-Y. Lu, and L.-C. Yu, *Benchmarking Clickbait Detection from News Headlines*. 2024. doi: 10.1109/o-cocosda64382.2024.10800145.

[8] S. Ghosh, S. Jhalani, and C. Oshiro, Topic Modeling-Driven Feature Engineering to Enhance Clickbait Detection in Social Networks. 2024, pp. 1–8. doi: 10.1109/iisa62523.2024.10786672.

[9] S. Ghosh, D. Sen, and S. Ghosh, *Enhancing Clickbait Detection with Cross-Modal Topic Modeling in Social Networks*. 2024, pp. 289–294. doi: 10.1109/icict62343.2024.00053.

[10] R. S. R, A. Sungheetha, M. A. Haile, A. H. Kedir, R. A, and C. B. G, "Clickbait Detection for Amharic Language using Deep Learning Techniques," *Journal of Machine and Computing*, pp. 603–615, Jul. 2024, doi: 10.53759/7669/jmc202404058.

[11] Y. Arfat and S. C. Tista, *Bangla Misleading Clickbait Detection Using Ensemble Learning Approach*. 2024, pp. 184–189. doi: 10.1109/iceeict62016.2024.10534333.

[12] H. Deng *et al.*, "Prompt-tuning for clickbait detection via text summarization," *arXiv (Cornell University)*, Apr. 2024, doi: 10.48550/arxiv.2404.11206.

[13] Y. Wang, Y. Zhu, Y. Li, J. Qiang, Y. Yuan, and X. Wu, "Clickbait detection via Prompt-Tuning with titles only," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 9, no. 1, pp. 695–705, Jun. 2024, doi: 10.1109/tetci.2024.3406418.

[14] M. E. Syahputra, A. P. Kemala, F. A. Tjan, and R. Susanto, "Clickbait detection in Indonesia Headline news using BERT ensemble models," *2021 4th International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)*, pp. 475–479, Dec. 2023, doi: 10.1109/isriti60336.2023.10467417.

[15] T. Natanya and C. Liebeskind, "Clickbait detection in Hebrew," *Lodz Papers in Pragmatics*, vol. 19, no. 2, pp. 427–446, Dec. 2023, doi: 10.1515/lpp-2023-0021.

[16] A. A. Imran, M. S. H. Shovon, and M. F. Mridha, "BaitBuster-Bangla: A comprehensive dataset for clickbait detection in Bangla with multi-feature and multi-modal analysis," *Data in Brief*, vol. 53, p. 110239, Feb. 2024, doi: 10.1016/j.dib.2024.110239.

[17] D.-M. Broscoteanu and R. T. Ionescu, "A Novel Contrastive Learning Method for Clickbait Detection on ROCLICO: A Romanian Clickbait Corpus of news articles," *arXiv (Cornell University)*, Jan. 2023, doi: 10.48550/arxiv.2310.06540.

[18] H.-C. Wang, M. Maslim, and H.-Y. Liu, "CA-CD: context-aware clickbait detection using new Chinese clickbait dataset with transfer learning method," *Data Technologies and Applications*, vol. 58, no. 2, pp. 243–266, Aug. 2023, doi: 10.1108/dta-03-2023-0072.

[19] M. N. Al-Ali and S. M. Hamzeh, "Extra cues extra views: A multimodal detection of Arabic clickbait thumbnail verbo-visual cues," *Discourse & Communication*, vol. 18, no. 1, pp. 3–27, Aug. 2023, doi: 10.1177/17504813231190332

[20] Kaustubh0201, "Clickbait-Classification/clickbait_data.csv at main · kaustubh0201/Clickbait-Classification," *GitHub*,2025. https://github.com/kaustubh0201/Clickbait-Classification/blob/main/clickbait_data.csv