Innovation and Technology Studies
Print ISSN: 3007-6919
Online ISSN: 3007-6927
DOI: https://doi.org/10.61784/its3017

LLM AND SOCIAL MEDIA FAKE NEWS DETECTION

ChenYe Zhao
Xi'an Jiaotong-Liverpool University, Suzhou 215000, Jiangsu, China.
Corresponding Email: Chenye.Zhao23@student.xjtlu.edu.cn

Abstract: The proliferation of fake news on social media has made automatic detection methods necessary. Traditional approaches often rely on dynamic or unavailable social context, while this study explores text-based classification methods. We empirically evaluated the performance of several classic machine learning models (Logistic Regression, Naive Bayes, Random Forest, Support Vector Machine, XGBoost) and the advanced large language model DeBERTa on a real-world dataset for fake news detection. For the classic models, we used the TF-IDF vectorizer to extract features. The results show that tree-based ensemble methods, especially Random Forest and XGBoost, performed exceptionally well, with accuracy and F1 scores exceeding 99%. In contrast, the DeBERTa model's performance was only slightly better than random guessing (accuracy of 50.32%), which is attributed to catastrophic overfitting due to its large number of parameters (184 million) when trained on a relatively small dataset. This highlights a key challenge in applying powerful LLMs to specialized tasks: their performance is highly dependent on a large amount of high-quality training data. The research findings suggest that in tasks with limited data, robust classic models may be more effective than complex models that require a large amount of data.
Keywords: Fake news detection; Text classification; Machine Learning; Large Language Models

1 INTRODUCTION

The advancement of media technology has brought people lots of benefits. When browsing social media platforms, people usually get massive amounts of information. Some of them can provide useful suggestions and guidance, while some will simply bring falsehood and deception, this is often called fake news. According to Klyuev, fake news can be traced back to 1439 when the printing process was invented [1]. Since then, fake news has been continuously produced in different forms. Currently the proliferation of fake news on the Internet has become a great crisis, and will cause significantly negative social impacts, like social panic or economic losses. Public services like healthcare and education, or even environmental protection will be affected too. Therefore, it is very necessary to identify fake news to prevent them from being spread. However, the number of fake news is so dramatic that it is costly and time-consuming to identify them, which is absolutely impossible with manpower alone, so people have begun to try to develop different detection methods.

Mehta and Ma state that existing fake news detection methods mostly realize identification by analyzing the information dissemination process or social structure [2,3]. Although these methods do show excellent detection performance, because social networks have dynamic changes, and some data privacy cannot be authorized, the practicality of these methods will also be constrained, when the social context is incomplete. Another drawback of traditional methods is that they often lack explanation of predicted results, which leads to public distrust of results, according to Wang [4]. Since these methods have shortcomings, the direction of improvement should be to extract semantic features at different levels, convert them into plain text classification tasks, and only use text content in social media for analysis and processing. Thota suggests that, Large Language Models, also called LLMs, which have extraordinary ability in understanding language and context modeling, will play an important role in the approach [5]. In addition, LLMs have already displayed their ability in explanation generation, so left to do is combining it with the fake news information. Another advantage of LLMs is that it can generate Dynamic Data Augmentation, which means that it can generate diverse and realistic synthetic training samples. For example, if one report indicates that "Vaccine A passed the phase 2 clinical trial". The LLMs-generated fake variants would be "Vaccine A caused severe side effects during phase 2 trial" and "WHO specialist claimed Vaccine A failed the test". And with these automatically generated fake news, better adaptability with this field will be enabled.

2 LITERATURE REVIEW

The revolution of LLMs has not been a sudden disruption, but a gradual and iterative process of innovation. This evolution could be categorized into five distinct yet connected phases, while each is built on previous outcomes and overcome different limitations.

Pre-Transformer Era: The roots of LLMs can be traced back to neural language models based on Recurrent Neural Networks (RNNs) and Long Short-Term Memory networks (LSTMs). Models like the Seq2Seq model with an encoder-decoder could tackle with variable-length sequences, which laid the ground for machine translation and text classification. However, because of their inherent sequential nature – processing words one after another, the training process would be notoriously slow and restricted the application of parallel hardware like GPUs. Meanwhile, it would

be difficult to capture long-range dependencies, even if for advanced LSTMs.

The Transformer Revolution: "Attention Is All You Need" by Vaswani et al. was published in 2017[6], was a watershed moment, which openly introduced the Transformer architecture. Its core innovation, the self-attention mechanism, allowed the model to process all words in a sequence in parallel and weigh their relationships simultaneously, regardless of their positional distances. This strategy significantly improved the training efficiency and enabled the model to build a more holistic understanding of context. This parallelizable design of Transformers meant it could leverage modern computing infrastructure to an unprecedented degree, turning out to be the undeniable turning point of LLM development.

The Rise of Pre-training: The Transformer architecture provided the perfect substrate for a new paradigm: pre-training on a massive corpus of text followed by fine-tuning on specific downstream tasks. OpenAI's Generative Pre-trained Transformer (GPT-1, 2018) demonstrated the power of a left-to-right, autoregressive pre-training objective, where the model learned to predict the next word in a sequence. By contrast, Google's BERT (2018) introduced Masked Language Modeling (MLM) for bidirectional pre-training, which randomly masked words in the input and trained the model to predict them using bidirectional context. It achieved breakthrough results on various understanding tasks. And it has become a promising way to align large language models (LLMs) with a human's intent [7].

The Era of Scaling: From 2019-2020, empirical research, notably form OpenAI, indicated that scaling up model size, data size, and computational resources would lead to predictable and continuous improvements in model performance. In sequence, Power Laws were observed, describing the mathematical relationship between scale and capability. OpenAI's GPT-2 and, more dramatically, the 175-billion-parameter GPT-3 were epitomes of this trend. GPT-3 demonstrated remarkable few-shot and zero-shot learning abilities, meaning it could perform new tasks simply from a natural language description or a few examples provided in its prompt, without the need for task-specific fine-tuning [8]. This suggested that scale itself could unlock qualitatively new emergent abilities.

The Era of Refinement and Alignment: Nowadays, the focus has shifted from pure scaling to refining output quality, ensuring safety and aligning model behaviors with complex human values, emotions and intent. Key techniques driving this phase include Reinforcement Learning from Human Feedback (RLHF), where models are fine-tuned based on preferences expressed by human raters, making them more helpful, honest, and harmless [9]. Furthermore, LLMs are evolving beyond pure text, some models are now being cultivated with visual understanding and generation capabilities, hoping to create more powerful multi-modal systems. The application scope is also deepening into specialized domains like medicine, with models like DrugGPT [10], which assisted with clinical decision-making.

Since its inception, the Transformer architecture has proven to be the most effective foundation for neural language modeling. By overcoming the sequential computation limits of recurrent neural networks (RNNs) via self-attention, it has brought revolutionary progress to NLP and AI as a whole. LLMs have permeated countless fields, with key applications including: Content Creation & Generation, Intelligent responding services, Code generation, and Information Retrieval. It could also enable researchers to transform unstructured clinical text into insights that can improve patient care and advance medical science [11]. As we look to the future, the trajectory of LLMs points toward greater specialization, improved reasoning, tighter human-AI collaboration, and a relentless focus on reliability and safety. It is obvious that these models will remain at the forefront of technological innovation, becoming increasingly integral to our digital lives, for the days to come.

3 MODEL EXPLANATION

3.1 Basic Concept

And what is going to be discussed in this part, DeBERTa, which means Decoding-enhanced BERT with Disentangled Attention, is an advanced BERT variant by Microsoft, improving language understanding via disentangled attention and enhanced mask decoder [12].

Disentangled attention: Usually, BERT applies each word as a single vector, combining its content embedding and position embedding. However, DeBERTa poses a great difference, while each word is encoded using two distinct vectors—one for content and another for position. Then the attention weights between words are computed using disentangled matrices, considering both their semantic meaning and relative positions. This matters a lot since the relationship between words should not only depends on their basic meaning, but also on how close they are. For instance, when considering two words, they tend to be linked more closely when they are in one specific sentence, rather than appearing in separate sentences.

Enhanced mask decoder: Similar but different, DeBERTa enhances BERT's MLM pretraining by introducing absolute position embeddings at the softmax decoding stage. While its disentangled attention tackles with relative positions, the enhanced decoder clearly incorporates absolute positional cues critical for syntactic understanding. For example, suppose a sentence telling '*The cat chased the mouse because it was hungry.'* If we mask the words 'it' and 'mouse', a model without absolute position awareness might incorrectly predict that 'it' refers to 'mouse', since 'mouse' is closer to the pronoun. However, in reality, 'it' refers to 'cat' (the subject of the sentence). Yet DeBERTa's Enhanced Mask Decoder resolves this ambiguity by incorporating absolute position information before prediction. Since word 'cat' appears earlier in the sentence (absolute position = 1), the model recognizes its higher syntactic importance as the subject, leading to correct interpretation.

3.2 Formulas Explanation

(a) DeBERTa Architecture: Disentangled Attention DeBERTa
Each token is represented by two vectors: H_i Encodes content, and $P_{i|j}$ Encodes relative position of token i.
The cross-attention score between tokens i and j is decomposed into four components:

$$A_{i,j}=\{H_i,P_{i|j}\}\times\{H_i,P_{j|i}\}^T=H_iH_j^T+H_iP_{j|i}^T+P_{i|j}H_j^T+P_{i|j}P_{j|i}^T \quad (1)$$

The attention weight between a pair of words is calculated by summing four distinct attention components derived from disentangled content and position representations: content-to-content, content-to-position, position-to-content, and position-to-position interactions[12].
(b) For a maximum relative distance k, the relative distance $\delta(i,j)$ between tokens =i and j is:

$$\delta(i,j)=\begin{cases}0 & if\ i-j\le -k\\ 2k-1 & if\ i-j\ge k\\ i-j+k & otherwise\end{cases} \quad (2)$$

The formula $\delta(i,j)$ is a clipping function that maps the raw position difference (i - j) between token i and token j to an index between [0, 2k), which is used to lookup the corresponding vector from the relative position embedding matrix P.
(c) Disentangled Self-Attention
DeBERTa extends standard attention by disentangling content and position into separate vectors:
(1) Content Projections:

$$Q_c=HW_{q,c}\quad K_c=HW_{k,c}\quad V_c=HW_{v,c}$$

This is identical to the standard Transformer. The content vectors H are linearly projected into Query (Q_c), Key (K_c), and Value (V_c) spaces.
(2) Relative Position Projections:
$Q_r=PW_{q,r}\quad K_r=PW_{k,r}$, where $P\in R^{2k\times d}$ is a fixed relative position embedding matrix.
The **relative position embedding matrix P** (a learnable parameter) is also linearly transformed to obtain relative position Query (**Q_r**) and Key (**K_r**) vectors. Note there is no Value (**V_r**) projection.
(3) Attention Scores:

$$A_{i,j}=Q_i^cK_j^{cT}+Q_i^cK_{\delta(i,j)}^{rT}+K_j^cQ_{\delta(i,j)}^{rT} \quad (3)$$

(4) Scaling and output:

$$H_o=softmax(\frac{A}{\sqrt{3d}})V_c \quad (4)$$

There are some key differences between BERT. DeBERTa adds Position-to-Content term and uses fixed relative position embeddings. It scales by $\sqrt{3d}$ (vs. $\sqrt{d}$ in BERT) to balance three attention terms.

4 EXPERIMENT

4.1 Experiment Process

The empirical analysis was conducted using a custom-built Python framework leveraging the Scikit-learn and XGBoost machine learning libraries. A real-world dataset composed of news articles labeled as either "True" or "Fake" was utilized. The dataset was constructed by combing two separate CSV files, one for each class, and then shuffling the combined entries in random to ensure that there is no inherent ordering bias. Then the dataset was partitioned into two distinct subsets: 80% of the data for training set and 20% of the data for test set, adopting a fixed random state for reproducibility. This process would ensure that the model is trained on one portion of the data and eventually evaluated on a completely unseen set to provide an unbiased assessment of its generalization capabilities. The TF-IDF vectorizer was configured to remove common English stop words and to restrict the vocabulary to the top 5,000 most significant features, converting the raw text data into a numerical matrix suitable for machine learning algorithms.
The core of the feature extraction process was a Term Frequency-Inverse Document Frequency (TF-IDF) vectorizer [13]. The text from two columns, ‘title’ and ‘text’, of each news article was concatenated to form a single input feature. The TF-IDF vectorizer was configured to remove common English stop words and to restrict the vocabulary to the top 5,000 most significant features, converting the raw text data into a numerical matrix suitable for machine learning algorithms.
Then, an evaluation of five distinct classical machine learning models was performed, including Logistic Regression, Naive Bayes, Random Forest, Support Vector Machine, and XGBoost. Each model was instantiated with its default or commonly recommended parameters, with specific adjustments made to ensure convergence and reproducibility (e.g., max_iter=1000 for Logistic Regression, random_state=42 for Random Forest). Another attempt was also conducted with an advanced pre-trained model, DeBERTa , from the Hugging Face Transformers library. A robust error-handling routine was implemented for this complicated model, which included a complete re-installation of its dependencies to resolve potential environment conflicts. In case that it might failed, a stub model was created to maintain the workflow integrity.
All models were trained on the training data and their performance was evaluated on the transformed test set. The evaluation protocol was standardized across all models, calculating key performance metrics including accuracy, a detailed classification report (precision, recall, F1-score), and a confusion matrix. And to better view the results, they

are visualized, including a composite figure of confusion matrices and a bar chart plotting the accuracies to see the differences.

4.2 Results

To better compare different models, one table and two figures will be shown. Table1 lists the six models, indicating their accuracy, precision, recall, F1-score and support in this experiment. In Table1, ‘Accuracy’ means the proportion of all news articles (both fake and real) that the model predicted correctly, Random Forest and XGBoost reach the highest. ‘Precision’ means that, among all news articles that the model flagged as "Fake", the proportion that were truly fake, while Random Forest, SVM and XGBoost reach the highest. ‘Recall’ means that, among all truly fake news articles, the proportion that the model successfully caught, still the three reach the highest. ‘F1 score’ means that a single metric that balances Precision and Recall. ‘Support’ means the number of actual fake and real news samples in the test set used for evaluation, here referring to 8980. Figure 1 uses Confusion Matrix to better display the outcomes, counting their numbers of true results false results, while the larger the number is, the darker the color will be, and the comparison of colors will directly indicate the result. Figure 2 uses Bar Chart to compare the effects of different models, while the comparison will be obvious through comparing the height, Random Forest, SVM and XGBoost is the highest.

Table 1 Classification Performance

Model	Accuracy	Precision	Recall	F1-score	Support
Logistic Regression	0.9875	0.99	0.99	0.99	8980
Naive Bayes	0.9338	0.93	0.93	0.93	8980
Random Forest	0.9984	1.00	1.00	1.00	8980
SVM	0.9953	1.00	1.00	1.00	8980
XGBoost	0.9984	1.00	1.00	1.00	8980
DeBERTa	0.5032	0.50	0.50	0.50	8980

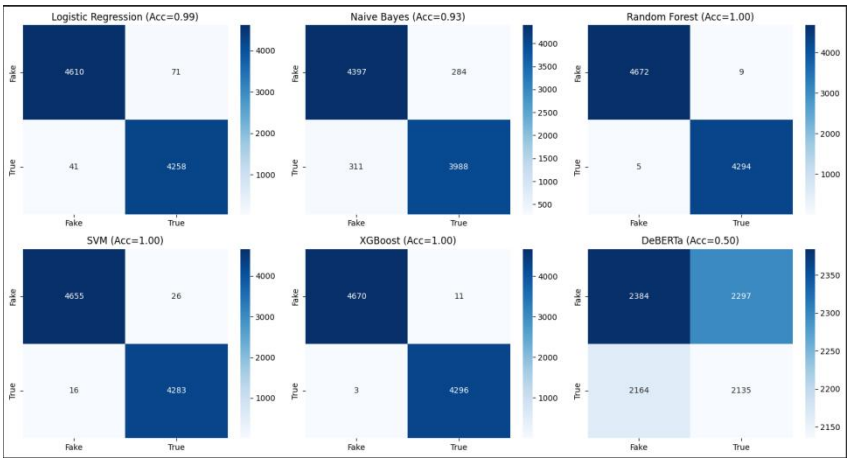

Figure 1 Confusion Matrix for the Proposed Model

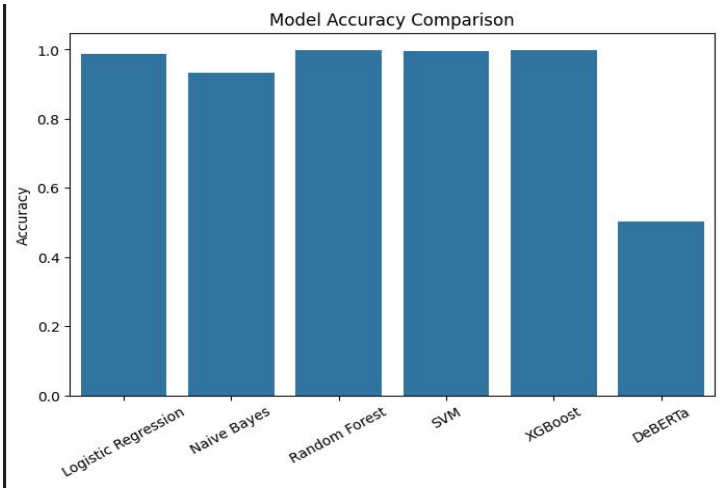

Figure 2 Bar Chart Comparing the Accuracy of All Evaluated Models

4.3 Analysis

The charms above indicate that Random Forest and XGBoost reach the highest accuracy among all of the models, meaning that they have the best performance in such tasks. Tree-based models automatically learn which features are most important, without the need for manual feature selection or heavy pre-processing. Meanwhile, they have great robustness to irrelevant features, thus performing well in practical tasks. For instance, Dieterrich indicated that random split selection did incredibly better [14].

However, our DeBERTa only shows an accuracy of 0.5032, simply over 0.50, why this happens? This result stems from catastrophic overfitting. Given that the Deberta-V3 base architecture contains approximately 184 million parameters, and our training set consists of only 8,980 support instances, limited training data may not be enough to enable models of this scale to learn generalizable patterns. Therefore, the model may simply remember surface-level noise and training set-specific statistical artifacts rather than meaningful semantic features that distinguish real text. This memory strategy will completely fail when the model encounters a test set, which disparately distributes the pattern of memory makes the memory useless, resulting in performance equivalent to random guessing.

5 CONCLUSION

In this article, we conducted an empirical evaluation of several machine learning models and a state-of-the-art large language model, DeBERTa, for the task of fake news detection on social media. The results indicated that traditional machine learning models, particularly tree-based ensemble methods, like Random Forest and XGBoost, achieved exceptionally high accuracy, precision, recall, and F1-scores—exceeding 99% in most metrics. In contrast, the DeBERTa model, despite its advanced architecture and strong theoretical foundation, performed only slightly above random chance, with an accuracy of approximately 50.32%. This stark underperformance is attributed to catastrophic overfitting, resulting from the model's large number of parameters (184 million) being trained on a relatively small dataset of only 8,980 samples. Consequently, DeBERTa memorized noise and dataset-specific artifacts rather than learning generalizable semantic features.

These findings highlight the critical importance of dataset scale and quality when deploying large language models for some specialized tasks like fake news detection. While LLMs like DeBERTa offer promising capabilities in semantic understanding and explainability, their practical application requires substantial labeled data and possibly task-specific adaptations such as prompt-based fine-tuning or few-shot learning，otherwise they might not show a good performance.

Future work should focus on collecting larger and more diverse datasets, exploring lightweight fine-tuning strategies, and integrating contextual and social features to enhance model robustness and real-world applicability. There will still be a long journey to explore, but it is certainly that the power of DeBERTa will be out of imagination, not only cooperating with generative AI, but in a promising way in various aspects of lives [15].

COMPETING INTERESTS

The authors have no relevant financial or non-financial interests to disclose.

REFERENCES

[1] Klyuev V. Fake news filtering: Semantic approaches. 2018 7th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), 2018: 9-15. DOI: 10.1109/icrito.2018.8748506.

[2] Mehta N, Pacheco M L, Goldwasser D. Tackling fake news detection by continually improving social context representations using graph neural networks. Proceedings of the ACL, 2022.

[3] Ma X, Wu J, Xue S, et al. A comprehensive survey on graph anomaly detection with deep learning. IEEE Transactions on Knowledge and Data Engineering, 2021.

[4] Wang B, Ma J, Lin H, et al. Explainable fake news detection with large language model via defense among competing wisdom. Proceedings of the ACM on Web Conference 2024, 2024: 2452-2463.

[5] Thota A, Tilak P, Ahluwalia S, et al. Fake news detection: a deep learning approach. SMU Data Science Review, 2018.

[6] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need. Advances in Neural Information Processing Systems (NeurIPS), 2017.

[7] Ouyang L, Wu J, Jiang X, et al. Training language models to follow instructions with human feedback. arXiv preprint arXiv:2203.02155, 2022.

[8] Thurner S, Hanel R, Klimek P. Introduction to the Theory of Complex Systems. Oxford University Press, 2018.

[9] Christiano P F, Leike J, Brown T, et al. Deep reinforcement learning from human preferences. Advances in Neural Information Processing Systems, 2017: 4299-4307.

[10] Zhou H, Liu F, Wu J, et al. DrugGPT: A collaborative large language model for drug analysis. Nature Biomedical Engineering, 2025, 9(1): 1-15. DOI: 10.1038/s41551-025-01471-z.

[11] Truveta Staff. Advancing clinical information extraction with LLM-Augmenter. Truveta, 2025, Oct 14.

[12] He P, Liu X, Gao J, et al. DEBERTA: Decoding-enhanced BERT with disentangled attention. Microsoft, 2020.

[13] de Sales M. TF-IDF in Hadoop Part 1: Word frequency in doc. 2009, Dec 31. https://marcellodesales.wordpress.com/2009/12/31/tf-idf-in-hadoop-part-1-word-frequency-in-doc/

[14] Dietterich T. An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting and randomization. Machine Learning, 2000, 40: 139-157. DOI: 10.1023/A:1007607513941.
[15] Dilmegani C, Palazoğlu M. The future of large language models. AIMultiple, 2025, Oct 19.