# A Multi-faceted Approach to Rumor Detection: Leveraging Large Language Models for Cross-domain Rumor Verification

Lisu Wang

**Our society encounters various challenges brought by the advent of abundant misinformation on online platforms, especially during worldwide events like COVID-19. Therefore, real-time rumor verification plays an indispensable role in diminishing the spread of misleading narratives. This paper presents a novel, multi-faceted approach to rumor verification that synergistically combines the abilities of large language models (LLMs) with traditional machine learning techniques. We introduce a comprehensive feature engineering framework that utilizes a powerful LLM, GLM-4.5, to extract a rich set of 20 different linguistic and semantic features from the text on social media augmented with Retrieval-Augmented Generation (RAG) to incorporate external evidence. Then, these features, which include analyses of coercive language, logical fallacies, and source credibility, are used to train a number of machine learning models. According to the results of our experiments, which were conducted on a cross-domain dataset of Weibo posts, this approach can improve the performance of the task of rumor detection significantly. An ensemble model achieves an F1 score as high as 0.9009 by combining the predictions of our feature-based machine learning models with the direct output of a separate LLM (GLM-4-plus), and this outperforms the baseline model, demonstrating the efficacy of our integrated methodology.**

***Index Terms*—Rumor Verification, Cross-domain Generalization, Large Language Models, Linguistic and Semantic Feature Engineering, Retrieval-Augmented Generation (RAG), Machine Learning (ML)**

## 1. INTRODUCTION

In today's society, online platforms play a crucial role in our lives. Users are able to publicly comment and express their ideas on various issues and news [1]. However, the information posted by the users can sometimes be unreliable and misleading. These rumors, especially those on social media, can cause severe real-world consequences, ranging from public health crises to social unrest. As a result, it is imperative to evaluate whether a piece of information is a rumor in real-time, as it emerges and evolves constantly and is a critical research problem.

Existing methods of rumor detection and verification often find it challenging to generalize to new and unseen events, and they frequently do not employ the external knowledge that is available at the time the rumor starts to spread. In order to address this gap, we propose an innovative approach that integrates time-sensitive external evidence for improved rumor verification. Our research aims to present a robust and adaptable framework for rumor verification that can effectively deal with the dynamic and challenging nature of online misinformation.

## 2. LITERATURE REVIEW

The task of rumor detection has long been studied by academia with extensive research, including a variety of methodologies. Early research often relied on hand-crafted features and traditional machine learning models. For instance, LSTM-based models were among the first to be applied to this task, demonstrating the potential of recurrent neural networks for rumor detection on the Weibo dataset [2]. Later, subsequent research explored Convolutional Neural Networks (CNNs), which further improved detection performance [3].

Graph-based neural networks have also achieved notable results by effectively capturing the complex relationships inherent in social media data. Bi-directional Graph Convolutional Networks (BiGCN) can model both the propagation and dispersion of rumors [4]. Other advanced models include PLAN [5] and PPA-WAE [6], which leverage attention mechanisms and propagation path aggregation, respectively. Additional models such as UCD-RD [7], ARG [8], and LeRuD [9] have also been proposed to enhance rumor detection performance.

Automatic rumor detection has a long history rooted in engineered linguistic and social features as well. Traditional feature-based methods extract lexical, syntactic, and propagation features and feed them to classical classifiers, while more recent work leverages deep networks to learn representations directly from text. But, both the approaches find it is challenging to generalize their models to perform the task in a new

domain: features that are effective in one context (e.g., political tweets) may lose effectiveness in another (e.g., health- or science-related reporting), and data-driven representations can still fail to incorporate real-time external evidence necessary for rumor identification.

More recently, large language models (LLMs) and retrieval-augmented framework have been proposed as novel and effective paradigms that can effectively improve the robustness and the abilities of the models. Cao et al. construct the SciNews dataset and propose several LLM-centric architectures (including modular designs that combine summarization, explicit evidence retrieval, and inference) to detect misinformation in scientific news without requiring explicit human-written claims [10]. They show that architectures that include an evidence-retrieval component often outperform direct end-to-inference approaches and stress that integrating retrieval can improve factual grounding for LLM decision making. Importantly, their work highlights practical constraints, such as domain specificity, that limit direct generalization to other languages and social-media settings.

Hu et al. empirically explore the role of LLMs in fake-news detection and analyze how LLMs can function as components (or "advisors") within detection pipelines [11]. Their study examines strengths and failure modes of LLMs in the fake-news context and suggests that while LLMs bring strong language understanding and reasoning capabilities, their raw outputs can still show some calibration and domain transfer issues; consequently, integrating external evidence and complementary models is a practical strategy to improve robustness.

Taken together, these studies motivate a research path for us: adopt the LLM + retrieval idea to provide evidence-grounded features, and explicitly address three gaps that remain in the previous studies. First, while LLM-based pipelines can produce high-level judgments, they are often less interpretable than feature-based approaches. Second, prior methods struggle to apply their approaches on new and unfamiliar domain of information. Third, most prior work focuses on English-language news or scientific articles rather than on Chinese social-media data; cross-domain transfer remains underexplored. This inspires us to combine LLM-extracted, human-interpretable feature scores with statistical validation and classical classifiers. Building on these observations, our approach (i) adapts the retrieval-augmented LLM pipeline to Chinese data, (ii) uses LLM prompts to produce a structured set of 20 interpretable feature scores for each post, (iii) validates feature discriminability with rigorous hypothesis tests and effect-size calculations, and (iv) ensembles multiple classical classifiers to improve cross-domain robustness. This positions our work as an application and innovative approach to rumor verification, especially in the Chinese language, cross-domain rumor-detection setting, and added emphasis on statistical validation and ensemble robustness.

## 3. METHODOLOGY

There are several key stages of our methodology: (1) data collection and preprocessing, (2) feature engineering with a large language model, (3) statistical feature analysis and selection, and (4) model training and evaluation using both individual machine learning models and an ensemble approach.

### *3.1. Data*

In this study, we utilized a cross-domain experimental setup. Cross-domain refers to the setting in which the training and test sets derive from different data distributions. Specifically, the Weibo dataset served as the training data, and the Weibo-COVID dataset was the test data. While the Weibo dataset represents the social media domain, the latter is classified as the public health domain. This cross-domain approach allowed us to evaluate our models' generalizability and robustness when faced with new and unseen events[10].

The training data contains 4606 posts, and the test data consists of 411 posts. Each entry consists of a source post (the original Weibo blog) and its comments.

### *3.2. Web Search and Retrieval-Augmented Generation (RAG)*

In order to provide our models with external context, we employed a Retrieval-Augmented Generation (RAG) approach. RAG is a method that combines information retrieval and generative modeling. The model can query an external knowledge source to collect relevant documents. After integrating the extracted information into the input, the large language model is able to generate responses that are more accurate, grounded, and context-aware.

We first extracted the blog content from each JSON entry, and then applied this content as the query to the search engine using the Search-Pro API. The search was initially limited to official government sources and expanded to the entire web in later experiments, and, for each entry, we retrieved the top five sources in the search results. For each, the information includes the title, link, content, reference, and publish data.

The content of these extracted search results was then used to augment the input to our language model in order to provide it with additional information to measure the veracity of the original post. By formatting the retrieved results into a prompt, we instruct the model to integrate external knowledge with the evidence provided, analyze both the content and discussion, and then determine whether the blog post is classified as a rumor. The prompt structure we provided to the language models is shown below:

```
classification_prompt = """You are a COVID-19 rumor analyst. Based on the following search page content, your knowledge, and the blog comments, determine step by step whether the blog content is a rumor. If it is a rumor, finally output yes; if it is not a rumor, finally output no.
## Blog content: <<content>>
## Blog comments: <<comments>>
## Search page content: <<search_result>>"""
```

Prompt 1 . web search RAG classification prompt

Then, the processed instances were saved into new JSONL files. The predictions can then be compared with the original labels to calculate standard evaluation metrics.

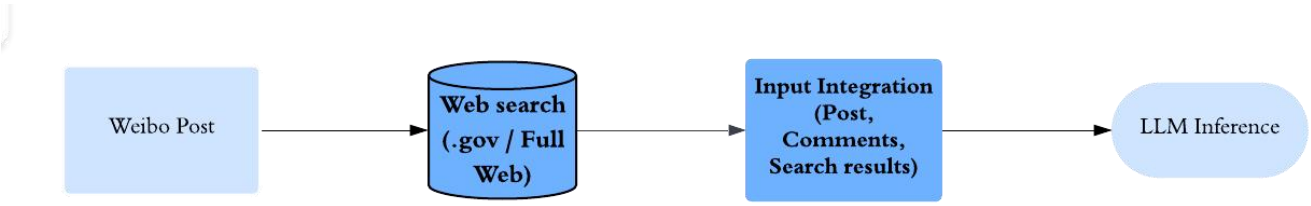


Fig. 1. Flow chart illustrating the Web Search RAG pipeline

### *3.3. Feature Engineering*

A core part of our methodology is using a large language model for feature engineering. We employed the GLM-4.5 model to extract a set of 20 distinct features from each blog post. The features were carefully designed to capture a wide range of semantic and linguistic characteristics relevant to misinformation. The list of features is provided below, along with a brief interpretation of each feature's name (excerpted from the feature extracting prompts used for LLM analysis).":

1 . **Coercive Language Analysis:** analyze whether the text contains coercive phrases and determine if it drives dissemination by inciting panic or anger.

2 . **Divisive Content Identification:** identify content that exploits ethnic, regional, or class differences to create division.

3 . **Manipulative Rhetoric Analysis:** check for manipulative rhetoric and assess the intensity of the emotional manipulation.

4 . **Absolutist Language Detection:** identify absolutist language and evaluate its frequency and context.

5 . **Factual Consistency Verification:** verify factual consistency and compare key claims in the text

against known facts or common sense to identify factual errors.

6 . **Logical Fallacy Identification:** identify logical fallacies and analyze the argumentative structure.

7 . **Attribution and Source Evaluation:** check if information is attributed to vague sources and assess their credibility.

**8 . Conspiracy Theory Narrative Detection:** detect conspiracy theory narratives.

9 . **Emotional Appeal Analysis:** quantifies the proportion and intensity of words appealing to emotions like fear, anger, or hope.

10 . **Pseudoscientific Language Identification:** identify pseudoscientific language and check for the misuse of scientific terms to package claims.

11 . **Call to Action Assessment:** analyze whether the text contains explicit action instructions and assess their urgency.

12 . **Authority Impersonation Detection:** identify if the text impersonates government agencies, official media, or well-known organizations to enhance credibility.

13 . **Bot Activity Sign Detection: l**ook for signs of bot amplification or coordinated inauthentic behavior.

14 . **User Reaction Assessment:** calculate the proportion of user comments that request evidence to quantify the rate of fact-based inquiries.

15 . **Dissemination Modification Tracking:** track modifications during the dissemination process. Identify and record the types of modifications that may occur as the text spreads.

16 . **Source Credibility Assessment:** assesses the authority, independence, and reliability of sources.

17 . **Factual Accuracy Verification:** verifies factual claims against trusted databases.

18 . **Information Completeness Check:** checks for omitted context that can distort understanding.

19 . **External Consistency Analysis:** compares claims with established knowledge.

20 . **Expert Consensus Alignment:** evaluates alignment with expert consensus.

We constructed a feature-specific prompt. For each data instance, the blog content, comments, and retrieved search results were substituted into the template, allowing the model to analyze the text and output feature-specific scores.

Part of the prompt is shown below:

```
You are a professional misinformation analysis expert. Your task is to receive a text and conduct a thorough, multi-dimensional analysis based on the following 15 carefully designed questions. Your goal is to identify potential misinformation
```

```
characteristics, manipulative strategies, and inherent risks within the text.
[List of Analysis Instructions]
Please analyze the input text strictly according to the following 15 questions.
For each question, you must:
1. Perform a deep analysis: Understand the core of the question and search for
relevant evidence within the text.
2. Provide a 0-1 score: Quantify the risk or characteristic intensity of the text
on this dimension.
      Scoring Reference:
         0.0 - 0.3: Very low risk or characteristic is almost non-existent. The
text is very neutral and objective.
         0.4 - 0.6: Moderate risk or characteristic is partially present. The text
contains some ambiguous or questionable statements, but they are not explicit or
strong.
         0.7 - 1.0: High risk or characteristic is clearly present. The text contains
clear, strong, and potentially misleading language or logic.
3. Provide a detailed explanation: Elaborate on the reasons for your score, citing
specific content from the text as evidence.
4. List key evidence: Extract and list the exact phrases, sentences, or structural
elements from the text that support your analysis.
...
```

Prompt 2 Excerpt of the feature-specific scoring prompt.

Before feature extraction, each data instance was checked against previously processed entries to avoid redundant computations. For each feature, the GLM-4.5 model was prompted to provide a score ranging from 0.0 to 1.0, indicating the intensity of the characteristic, with a detailed explanation and the corresponding supporting evidence from the text (to ensure data consistency, missing or malformed responses were defaulted to 0.0).

Then, the enriched dataset was stored in a JSONL file and contained the following fields: source, comments, search_result, label, and 20 numerical feature columns. This dataset served as the input for feature ranking (Section 3.3.3) and subsequent machine learning model training (Section 3.5).

For the test data, the same procedure was applied.

The model training process is shown in Fig. 2, while Fig. 3 illustrates the testing workflow:


Fig. 2. The model training process

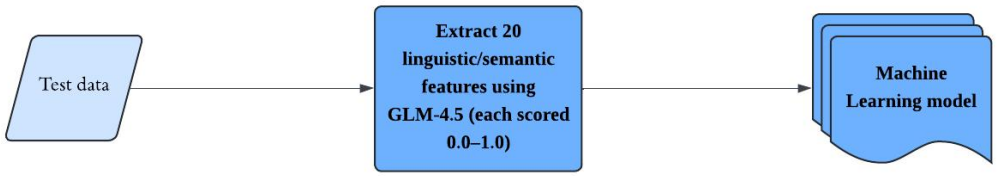


Fig. 3. The model testing process

### *3.4. Statistical Analysis and Feature Selection*

To validate the efficacy of the extracted 20 features, we conducted a rigorous and thorough statistical analysis. We used hypothesis testing for each feature, determining if there was a statistically significant difference between the feature scores of rumor and non-rumor posts. Specifically, the null hypothesis assumed that the two groups have equal means; in practice, this means that the observed difference is small enough to be perceived as statistically insignificant. The alternative hypothesis stated that the means are not equal. From each test, we obtained a corresponding P-value, the probability of observing a difference at least as extreme as the one measured, if the null hypothesis is true [13]. A small p-value (typically less than 0.05) suggests that the observed difference is unlikely to occur by chance, indicating a statistically significant difference between rumor and non-rumor posts. Conversely, a large p-value indicates that the observed difference could plausibly be due to random variation [14]. This setup directly tests whether each feature we proposed is useful for rumor detection.

Before applying a t-test that can directly prove the effectiveness of our features, we first need to use Levene's test for the homogeneity of variances. This is because the standard Student's t-test assumes equal variances, while Welch's t-test is used when variances are unequal, so we need to select an appropriate two-sample t-test based on Levene's test. In Levene's test, we set up another null hypothesis that the data have equal variances with an alternative hypothesis that the variances are unequal. The calculated results of all features suggest that the assumption of equal variances failed. The results is shown below (for reproting purposes, we denote extremely small values, when $p < 1 \times 10^{-300}$ , as 0):

| Feature | Levene_statistic | Levene_p_value |
|---|---|---|
| Coercive Language Analysis | 1055.578979 | $1.167773\times10^{-208}$ |
| Divisive Content Identification | 1098.316056 | $3.464952\times10^{-216}$ |
| Manipulative Rhetoric Analysis | 1654.515357 | $2.683953\times10^{-309}$ |
| Absolutist Language Detection | 297.841184 | $1.018695\times10^{-64}$ |
| Factual Consistency Verification | 59.096268 | $1.823622\times10^{-14}$ |
| Logical Fallacy Identification | 698.313687 | $2.138527\times10^{-143}$ |
| Attribution And Source Evaluation | 25.942845 | $3.657308\times10^{-07}$ |
| Conspiracy Theory | 3259.779694 | 0.000000 |
| Emotional Appeal Analysis | 168.384189 | $7.634375\times10^{-38}$ |
| Pseudoscientific Language Identification | 135.144044 | $8.242163\times10^{-31}$ |
| Call Action Assessment | 715.307933 | $1.338356\times10^{-146}$ |
| Authority Impersonation Detection | 564.783760 | $7.399861\times10^{-118}$ |
| Bot Activity Sign Detection | 245.835356 | $5.133163\times10^{-54}$ |
| User Reaction Assessment | 403.176851 | $4.916363\times10^{-86}$ |
| Dissemination Modification Tracking | 907.508687 | $3.960184\times10^{-182}$ |
| Source Credibility Assessment | 55.774005 | $9.670968\times10^{-14}$ |
| Factual Accuracy Verification | 10.185302 | $1.425164\times10^{-03}$ |
| Information Completeness Check | 140.812660 | $5.157067\times10^{-32}$ |
| External Consistency Analysis | 68.978981 | $1.294898\times10^{-16}$ |
| Expert Consensus Alignment | 580.076637 | $8.139063\times10^{-121}$ |

Table 1 The calculated results of Levene’s test

Therefore, we leveraged Welch's t-test, which can be applied when the variances are unequal, to compare the means of the two groups. The t-statistic is calculated as:

$$t=\frac{\overline{X_1}-\overline{X_2}}{\sqrt{\frac{s_1^2}{n_1}+\frac{s_2^2}{n_2}}} \quad \text{Eq. 1}$$

Where $\overline{X}$ is the sample mean, $s^2$ is the sample variance, and n is the sample size.

The P-value was used to assess statistical significance. The resulted P-values of all the 20 features is shown below:

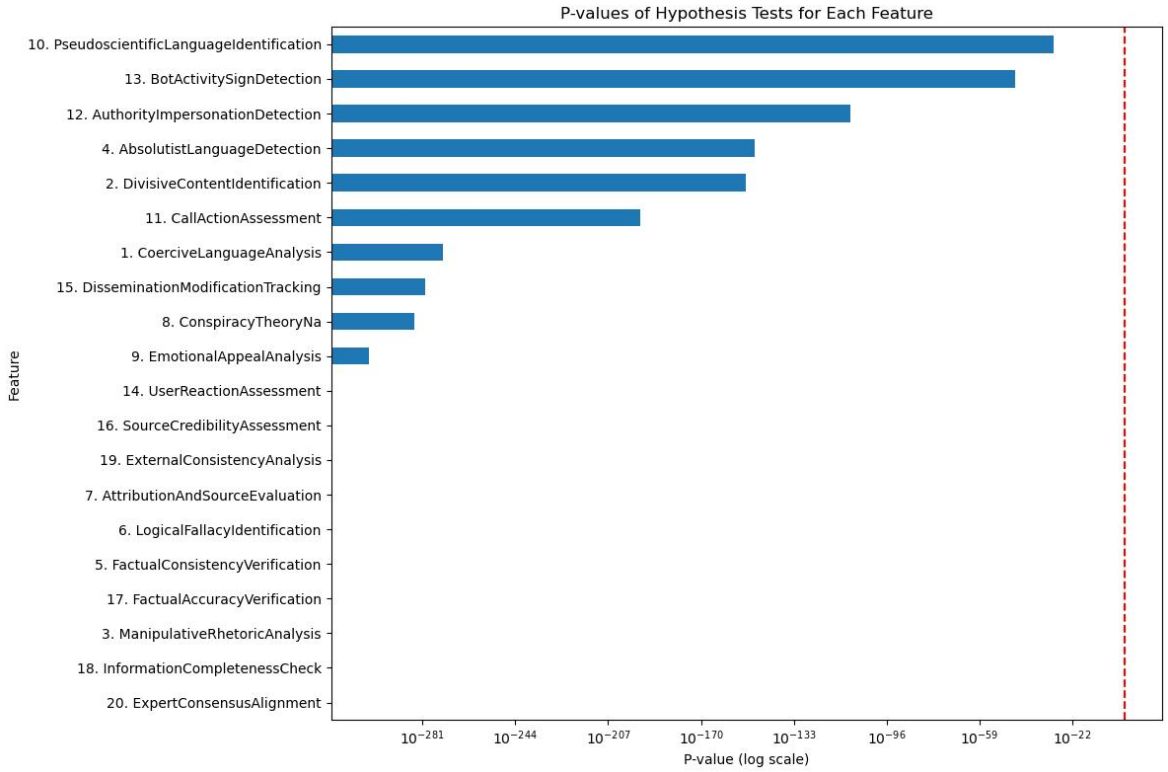


Fig. 4. P-values of Hypothesis Tests for Each Feature

As shown in Fig. 4, our analysis revealed that all 20 features demonstrate a statistically significant difference between the two classes ($p<0.05$), confirming their relevance.

We also calculated the effect size for each feature using Cohen's d, which is given by:

$$d=\frac{\overline{X_1}-\overline{X_2}}{s_p} \quad \text{Eq. 2}$$

Where the $s_p$ is the pooled standard deviation.

The resulted values are shown below:

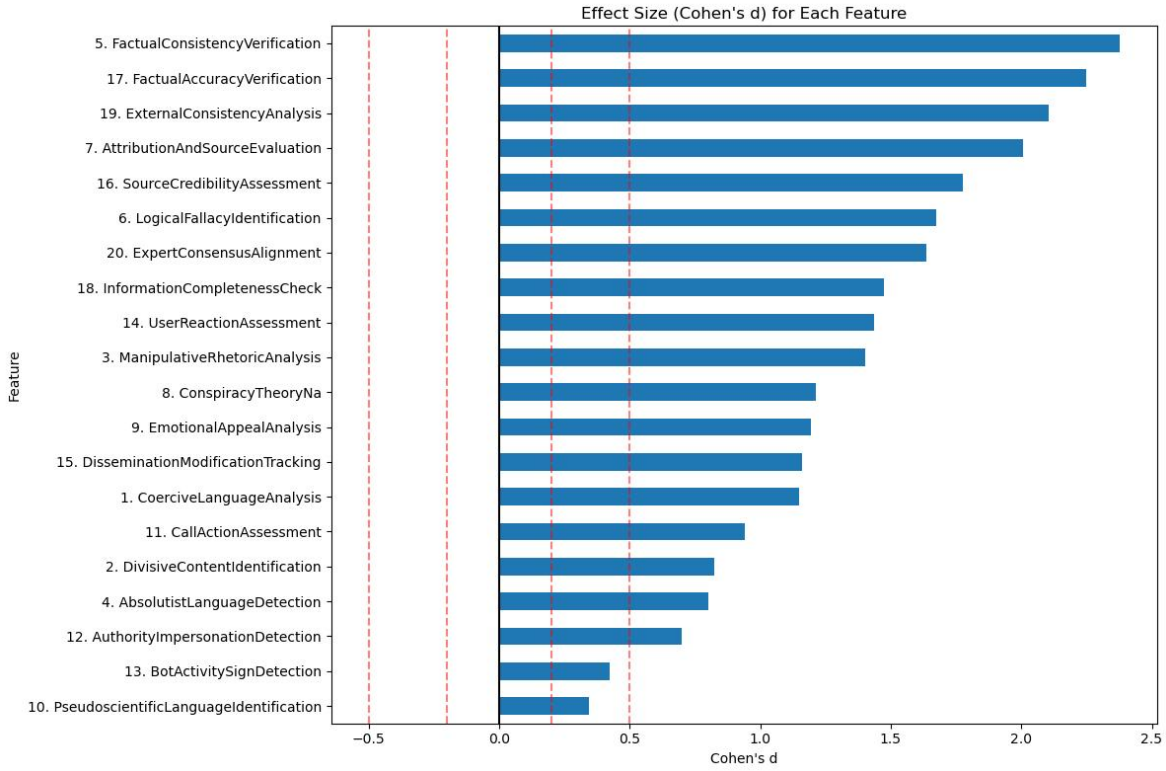


Fig. 5. Effect Size (Cohen d) for Each Feature

The results verified that the majority of the features, such as “5. FactualConsistencyVerification” (d=2.3775) and “17. FactualAccuracyVerification” (d=2.2473), have a large effect size. However, two features, “13. Bot Activity Sign Detection” and “10. Pseudoscientific Language Detection”, have effect sizes below 0.5. According to conventional benchmarks (0.2 = small, 0.5 = medium, 0.8 = large [15]), these values are classified into the medium range. This finding does not necessarily indicate that these features are ineffective. On the contrary, both features were still statistically significant, according to the previous hypothesis test. The medium effect size suggests that the magnitude of the separation is smaller than that of other features, but still moderately meaningful.

This statistical analysis supports the selection of the 20 features for subsequent model evaluation and ensure the reliability of our methodology.

### *3.5. Machine Learning Models*

After rigorous statistical analysis, we trained and evaluated several traditional machine learning models, implemented with the scikit-learn (sklearn) library, using the extracted feature scores mentioned before,

which serve as numerical representations of linguistic and social signals identified by the LLM (GLM-4.5). The models are Logistic Regression, which captures linear decision boundaries [16]; a Decision Tree, which classifies by hierarchical, non-linear decision rules [17]; and a Support Vector Machine (SVM), which finds an optimal hyperplane that maximizes the margin between rumor and non-rumor samples [18].

The feature scores served as the input to the models. Each model was trained on the training dataset and evaluated on the test dataset. Predictions were stored as new columns in the dataframe. These predictions were subsequently compared with the labels to calculate evaluation metrics.

### *3.6. Ensemble Model*

In order to improve the performance to a greater extent and reduce variance from single models, we developed an ensemble model that amalgamates the predictions of our best-performing machine learning models with the direct output of a separate LLM. Specifically, we applied the voting-based ensemble approach, where the ultimate label for each instance is determined by the majority of predictions from the three models. The models are the Decision Tree model, the SVM model, and the GLM-4-plus model [from Experiment 3]. During the inference, each model, trained independently, produces a binary label for each test sample. Then, we saved these predictions into separate columns. After that, we compute a majority-vote ensemble. That is, we counted how many base classifiers predicted the sample as “rumor” (1), and if two or more classifiers (out of three) identified the sample as 1, the ensemble prediction was set to 1. Otherwise, the ensemble model will output “non-rumor” (0). Because the number of base models employed here was odd, there was no ambiguity from ties. In other words, the final prediction of the ensemble model is determined by the winner of the vote among the three models. This hybrid ensemble paradigm is novel, as it integrates feature-based machine learning predictions with direct evidence-informed LLM outputs, enhancing cross-domain generalization.

The ensemble model used in this study is illustrated in Fig. 6.

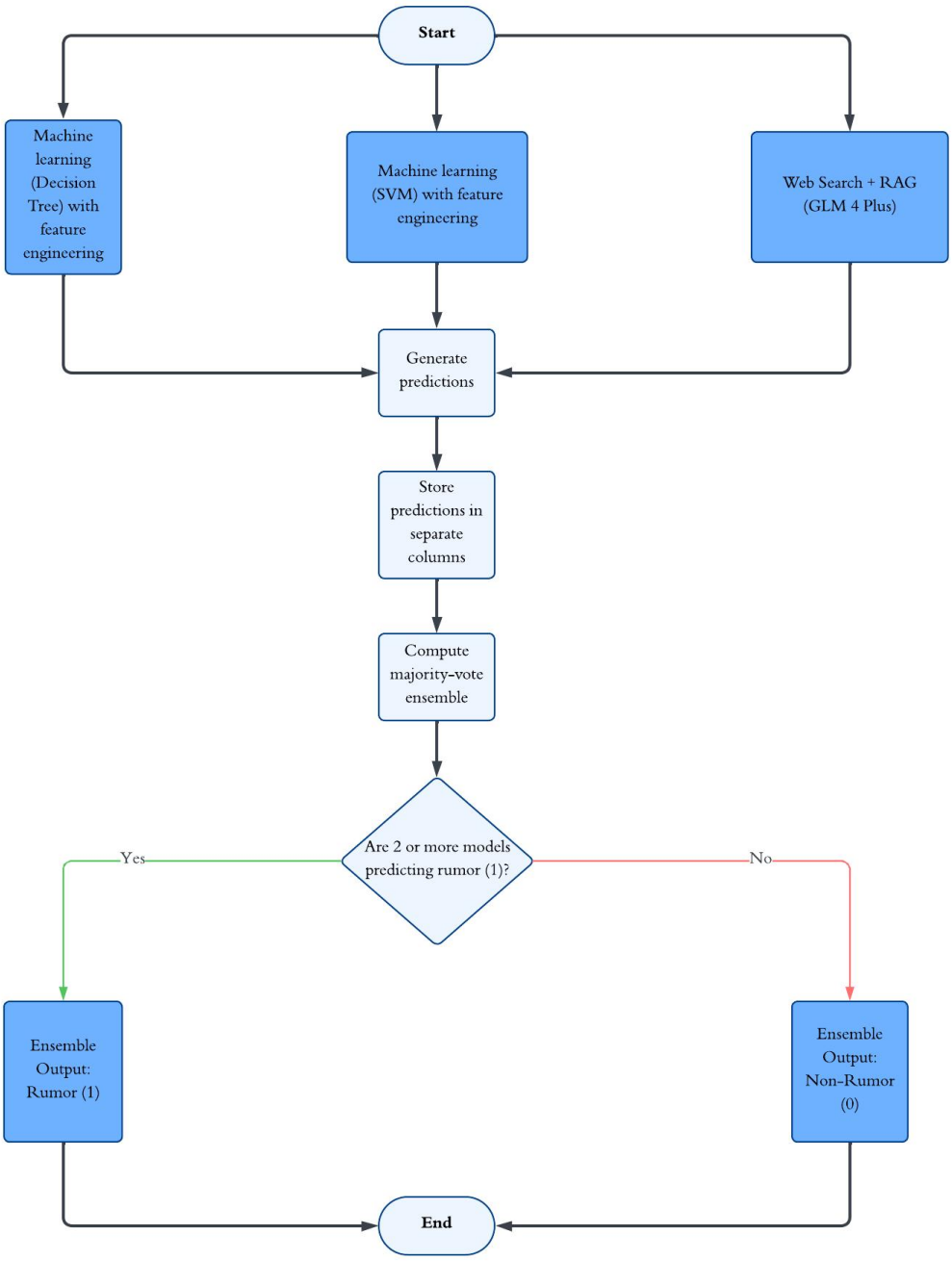


Fig. 6 Overview of the proposed ensemble model

## 4. Experiments and Results

In our study, we conducted a series of experiments to validate our proposed methodology. A LoRA fine-tuned LLama3 8B model, also trained on the Weibo dataset and tested on the separate Weibo-COVID dataset, is our baseline model.

| Model/Method | Accuracy | Precision | Recall | Negative F1 | F1-score |
|---|---|---|---|---|---|
| **Precious Studies** | | | | | |
| LSTM [2] | 0.4185 | – | – | 0.4156 | 0.4213 |
| CNN [3] | 0.4161 | – | – | 0.4059 | 0.4258 |
| BiGCN [4] | 0.6058 | – | – | **0.4938** | 0.6773 |
| UCD-RD [12] | **0.6788** | – | – | 0.4677 | **0.7700** |
| **Our study** | | | | | |
| Fine-Tuning Llama3 (Baseline) | 0.7737 | 0.8834 | 0.7462 | 0.7224 | 0.8090 |
| Web Search RAG (GLM4 + gov) | 0.7689 | 0.7553 | **0.9470** | 0.5815 | 0.8403 |
| Web Search RAG (DS + gov) | 0.8200 | 0.8065 | **0.9470** | 0.7016 | 0.8711 |
| Web Search RAG (GLM4 + all) | 0.8418 | 0.8419 | 0.9280 | 0.7566 | 0.8829 |
| Web Search RAG (FT Llama3 + all) | 0.8297 | 0.8849 | 0.8447 | 0.7712 | 0.8643 |
| Logistic Regression (LLM Features) | 0.8200 | 0.8545 | 0.8674 | 0.7448 | 0.8609 |
| Decision Tree (LLM Features) | 0.8248 | 0.8721 | 0.8523 | 0.7600 | 0.8621 |
| SVM (LLM Features) | 0.8345 | 0.8769 | 0.8636 | 0.7718 | 0.8702 |
| **Ensemble Model (Best)** | **0.8710** | **0.8893** | 0.9129 | **0.8153** | **0.9009** |

Table 2 Performance Comparison of Our Models and Previous Work under the Cross-Domain Setting (Weibo → Weibo-COVID)

As shown in Table 2, our proposed ensemble model remarkably outperforms all prior models on the task of identifying rumors in the same cross-domain setting. Our model achieves an accuracy score of 0.8710, compared to the previous best score of 0.6788 from UCD-RD [10]. The F1-score reaches 0.9009, representing a substantial improvement of over 13 percentage points. Moreover, with a negative F1-score of 0.8153, which demonstrates a significant improvement over previous works, our model excels at recognizing non-rumors.

## 5. Comparative Analysis and Discussion

The noticeable and superior performance of our method can be attributed to several key innovations and advantages listed below.

### *5.1. Innovative Multi-Dimensional Feature Engineering*

Compared to traditional feature-extracting techniques that rely on handcrafted or fundamental linguistic features, we applied a large language model (GLM-4.5) as a sophisticated feature extractor using a novel framework. This allows us to capture 20 deeper semantic and pragmatic features in the text from distinct and innovative dimensions. By systematically applying the LLM on structured feature engineering, we shift the application of an LLM beyond the traditional "black-box" classifier to a more interpretable model whose feature contributions can be quantified.

### *5.2. Statistically-Validated Feature Selection*

Our methodology is based on complete and rigorous statistical analysis. With each of the 20 LLM-generated features clearly validated through hypothesis testing, we can confirm their statistical significance in the task of differentiating rumors from non-rumors. Additionally, by calculating Cohen's d for effect size, we were able to identify core features with high discriminatory ability. We proved that all of the features are relevant to the task and examined their usefulness as well. This data-driven approach ensures our models are built on a concrete foundation with meaningful and impactful features.

### *5.3. Hybrid Ensemble Methodology*

Combining the predictions of two feature-based traditional ML models, Decision Tree and SVM, with the inference of a RAG-enhanced LLM [GLM-4-plus], the hybrid ensemble model outperformed all other models. This inventive and novel paradigm inherited the robustness of traditional classifiers and leveraged the contextual reasoning and fact-verifying abilities of LLMs. This amalgamation not only boosts overall performance with an F1-score of 0.9009 but also improves the model's generalization and robustness abilities, which are crucial for challenging cross-domain applications in real-life settings.

### *5.4. Effective Use of Real-Time External Evidence*

All of our experiments integrated Retrieval-Augmented Generation (RAG) by deploying real-time searches to collect external evidence. This helps us to overcome many challenges and limitations that many prior methods, which could not utilize external information available at the time of a rumor's emergence, encountered. We provided rich context to the LLM for both direct classification and feature extraction. Therefore, we are able to improve the model's fact-checking skill effectively and enhance its adaptability to fit into the dynamic shift of events.

## 6. CONCLUSIONS

In this paper, we have presented a novel and effective approach to rumor verification that leverages the ability of large language models and the robustness of traditional machine learning. Our comprehensive feature engineering framework, based on the GLM-4.5 model, offers a nuanced and exhaustive representation of social media text, allowing our models to achieve outstanding performance. In addition, the statistical validation of our features and the strong performance of our ensemble model show the value of a multi-faceted and data-driven method for the task of identifying rumors. This work sets a new performance benchmark for cross-domain rumor verification and also provides a more interpretable, robust, and effective paradigm for solving the problem of online misinformation.

## REFERENCES

[1] T. Bian, "Rumor Detection on Social Media with Bi-Directional Graph Convolutional Networks", AAAI, vol. 34, no. 01, pp. 549-556, Apr. 2020.

[2] J. Ma, W. Gao, P. Mitra, S. Kwon, B. J. Jansen, K.-F. Wong, and M. Cha, "Detecting rumors from microblogs with recurrent neural networks," in Proc. 25th Int. Joint Conf. Artif. Intell. (IJCAI), New York, NY, USA, July 2016, pp. 3818–3824.

[3] F. Yu, Q. Liu, S. Wu, L. Wang, and T. Tan, "CAMI: A convolutional approach for misinformation identification," in Proc. 26th Int. Joint Conf. Artificial Intelligence (IJCAI), Melbourne, VIC, Australia, 2017, pp. 3905–3911.

[4] T. Bian, X. Xiao, T. Xu, P. Zhao, W. Huang, Y. Rong, and J. Huang, "Rumor detection on social media with bi-directional graph convolutional networks," in Proc. 34th AAAI Conf. Artificial Intelligence (AAAI), New York, NY, USA, 2020, pp. 549–556.

[5] L. M. S. Khoo, H. L. Chieu, Z. Qian, and J. Jiang, "Interpretable rumor detection in microblogs by attending to user interactions (PLAN)," in Proc. 34th AAAI Conf. Artificial Intelligence (AAAI), New York, NY, USA, 2020, pp. 6367–6374.

[6] P. Zhang, H. Ran, C. Jia, X. Li, and X. Han, "A lightweight propagation path aggregating network with neural topic model for rumor detection (PPA-WAE)," Neurocomputing, vol. 458, pp. 468–477, 2021.

[7] H. Zhang, J. Dong, and C. Yu, "UCD-RD: Uncertainty-aware contrastive learning for debunking rumors," in Proc. 59th Annual Meeting of the Association for Computational Linguistics (ACL), Online, 2021, pp. 234–245.

[8] K. Shu, A. Sliva, S. Wang, J. Tang, and H. Liu, "Fake news detection on social media: A data mining perspective," SIGKDD Explor. Newsl., vol. 19, no. 1, pp. 22–36, 2017.

[9] Y. Yuan, J. Ma, and H. Cao, "LeRuD: Leveraging representation learning for rumor detection on social media," in Proc. 29th Int. Joint Conf. Artificial Intelligence (IJCAI), Yokohama, Japan, 2020, pp. 4751–4757.

[10] Y. Cao, A. M. Nair, E. Eyimife, N. Jamalipour Soofi, K. P. Subbalakshmi, J. R. Wullert II, C. Basu, and D. Shallcross, "Can Large Language Models Detect Misinformation in Scientific News Reporting?," arXiv preprint arXiv:2402.14268, Feb. 2024.

[11] B. Hu, Q. Sheng, J. Cao, Y. Shi, Y. Li, D. Wang, and P. Qi, "Bad Actor, Good Advisor: Exploring the Role of Large Language Models in Fake News Detection," Proceedings of the AAAI Conference on Artificial Intelligence, vol. 38, no. 20, pp. 22105–22113, Mar. 2024, doi: 10.1609/aaai.v38i20.30214.

[12] H. Ran and C. Jia, "Unsupervised Cross-Domain Rumor Detection with Contrastive Learning and Cross-Attention", AAAI, vol. 37, no. 11, pp. 13510-13518, Jun. 2023.

[13] J. C. Travers, B. G. Cook, and L. Cook, "Null hypothesis significance testing and p values," Learn. Disabil. Res. Pract., vol. 32, no. 4, pp. 208–215, 2017, doi: 10.1111/ldrp.12147.

[14] D. Curran-Everett, "Explorations in statistics: hypothesis tests and P values," Advances in Physiology Education, vol. 33, no. 2, pp. 81-86, 2009.

[15] Kallogjeri D, Piccirillo JF. A Simple Guide to Effect Size Measures. JAMA Otolaryngol Head Neck Surg. 2023;149(5):447–451. doi:10.1001/jamaoto.2023.0159

[16] M. P. LaValley, "Logistic regression," Circulation, vol. 117, no. 18, pp. 2395–2399, 2008.

[17] Y.-Y. Song and Y. Lu, "Decision tree methods: applications for classification and prediction," Shanghai Arch. Psychiatry, vol. 27, no. 2, pp. 130–135, 2015.

[18] V. Jakkula, "Tutorial on support vector machine (SVM)," School of EECS, Washington State Univ., Pullman, Tech. Rep. 37.2.5, 2006.