

An Agentic Email Spam Detection System Based on DeepSeek-R1, Dify, and Fine-Tuned BERT

Zhijun Qu

School of Computer Science and Engineering
Xi'an Technological University
Xi'an, China
E-mail: quzj158@163.com

Zhongsheng Wang

State and Provincial Joint Engineering Lab. of
Advanced Network, Monitoring and Control
Xi'an Technological University
Xi'an, China
E-mail: wzsh1681@163.com

Abstract—Spam email has long threatened communication security and work efficiency. To address this, we design and implement an email spam-detection agent that integrates the DeepSeek large language model, the Dify agent-development platform, and a fine-tuned BERT model. The system uses BERT as the core classifier, leveraging its strengths in semantic understanding and deep feature extraction; it adopts a binary scheme (label 0 = ham, 1 = spam), and fine-tuning enables effective recognition of email text features. Meanwhile, the DeepSeek LLM is introduced to exploit its capabilities in reasoning and generation: for ham messages, DeepSeek produces key-point summaries, and for spam it provides risk explanations and safety recommendations. With Dify's tool orchestration and human-computer interaction interface, the system automates the entire pipeline from parsing email content to intelligent decision-making and interactive feedback, forming an end-to-end agentic framework for spam detection. For experiments, we train and validate on the Kaggle email dataset (33,715 messages: 17,170 spam / 16,545 ham), using a 70%/15%/15% train/validation/test split. On the spam-detection task, the system achieves 99.49% accuracy; precision, recall, and F1 reach 99.42%, 99.57%, and 99.49%, respectively. These results demonstrate excellent detection performance and strong generalization. In summary, the proposed DeepSeek-Dify-BERT integrated agent effectively safeguards user communications, reduces potential information-security risks, and substantially improves the intelligence and automation of the detection workflow.

Keywords-Deepseek; Dify; BERT Model; Email Detection; Agent

I. INTRODUCTION

With the rapid development of information technology, email has become an indispensable medium of communication in modern society,

widely used in personal correspondence, enterprise operations, and government affairs. However, as email communication becomes pervasive, the problem of spam is growing more severe—characterized by surging volume, diverse forms, and strong content obfuscation. Spam not only disrupts normal user experience but also triggers a range of security risks, such as privacy leakage, phishing, malware dissemination, and exploitation of system vulnerabilities. Consequently, building an efficient, intelligent, and scalable spam-detection mechanism has become a core challenge for improving network communication security and user experience[1].

Traditional spam-detection approaches—keyword filtering, rule engines, blacklist/whitelist matching, and Bayesian classifiers—achieved certain results in earlier stages, but they often struggle against today's increasingly complex and semantically flexible spam, which employs diverse evasion tactics. These methods tend to suffer from lower accuracy, higher false alarms, and a lack of contextual understanding and adaptability. As spam continues to evolve and attacks become more covert, traditional techniques face growing pressure, calling for more intelligent and flexible recognition mechanisms. In recent years, with the rapid progress of natural language processing (NLP) and deep learning, the pretrained language model BERT (Bidirectional Encoder Representations from Transformers) has become a key technology for email classification and spam detection; its deep contextual modeling markedly improves accuracy and robustness.

Meanwhile, the rise of agent technology has provided strong momentum for building systems with autonomous decision-making, multi-turn dialogue, and task execution. Agents can perceive and understand user intent, decompose and schedule complex tasks intelligently, and thus enhance overall automation and intelligence. As an agent-development platform integrating large language models, Dify offers advantages in workflow orchestration, multi-tool integration, and human-computer interaction design, enabling modular and intelligent execution of complex detection processes. DeepSeek, a leading domestic large language model, performs well in text generation and understanding and can effectively strengthen grasp of deep semantics and contextual logic in email content. By deeply integrating NLP models with agent capabilities, we can achieve higher-level text analysis and semantic understanding while building interactive, manageable, and evolvable spam-detection workflows[3].

Based on this background, this paper proposes a spam-detection agent that integrates DeepSeek, Dify, and BERT. The agent employs BERT as the base model for feature extraction from email text, leverages DeepSeek to enhance understanding of complex semantics, and uses the Dify platform to construct intelligent interaction and automated processing pipelines, thereby enabling accurate identification and efficient handling of spam.

II. RELATED WORK

Spam detection, as an important branch of text classification, has long attracted extensive attention from both academia and industry. As spam becomes increasingly complex, the technology has evolved from rule-based approaches to machine learning, then to deep learning and agent-based systems.

A. Rule-Based and Traditional Machine-Learning Methods

Early spam filters primarily relied on rule-based systems such as keyword matching, regular expressions, black/white lists, and sender-reputation mechanisms. These methods are simple to implement and computationally efficient, but

they depend heavily on manually crafted rules and struggle with rapidly changing content and semantic obfuscation. Adversaries can bypass filters by altering spellings, adding noise tokens, or mimicking legitimate emails, which undermines precision and adaptability. With the rise of machine learning, statistical methods were introduced into spam detection. Classic algorithms such as Naive Bayes (NB), Support Vector Machines (SVM), and k-Nearest Neighbors (KNN) have been widely used. By modeling class probabilities or decision boundaries for spam vs. ham, they provide a degree of generalization. NB, in particular, has seen widespread deployment for its efficiency and interpretability. However, these approaches typically rely on hand-engineered features (e.g., TF-IDF matrices, bag-of-words) and struggle to capture contextual and semantic relationships. Their performance often degrades in multilingual, cross-domain, or structurally complex email content[4].

B. Deep Learning and Pretrained Language Models

With the rapid progress of deep learning—especially in NLP—neural approaches have increasingly replaced traditional algorithms. Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and Long Short-Term Memory (LSTM) models learn features directly from text, reducing manual feature engineering and improving understanding of linguistic structure and context. Building on this, BERT (Bidirectional Encoder Representations from Transformers) leverages a bidirectional Transformer architecture to capture rich contextual semantics. Pretrained on large corpora and then fine-tuned for downstream tasks, BERT achieves strong classification performance with minimal task-specific adjustments. In spam detection, BERT improves both accuracy and recall and handles semantically complex or obfuscated content, with advantages demonstrated across multiple studies[10].

C. Emerging Agentic Approaches to Spam Detection

As large language models and task-oriented

agent technologies mature, spam detection is moving toward more intelligent and automated paradigms. Agent systems incorporate environment perception, autonomous decision-making, multi-turn interaction, and task execution, and have become a promising direction for building intelligent email systems. Dify, an agent-development platform integrating LLM capabilities, supports workflow orchestration and agent construction, offering high flexibility, modularity, and interactivity; with users in the loop, filtering strategies and behaviors can be continuously refined. DeepSeek, one of the leading large language models, shows strong performance in semantic understanding, complex reasoning, and long-text modeling. In spam detection, at the semantic understanding and generation layer, DeepSeek can serve as an explanation-and-summarization module: generating key points for ham and providing risk cues and safety advice for suspected spam, effectively complementing traditional language models' limitations in generalization and complex-context understanding[2].

In summary, spam-detection technology has evolved from early rule- and statistics-based methods to deep-learning-centered solutions delivered within agent frameworks. Although pretrained models such as BERT achieve strong classification accuracy, real-world deployments still require user-facing interaction and causal/chain reasoning mechanisms to produce interpretable conclusions and actionable recommendations. To this end, building on related research, this paper proposes an integrated DeepSeek + Dify + BERT architecture (see Figure. 1): BERT provides robust semantic representations and binary classification; DeepSeek generates user-oriented key-point summaries, risk explanations, and remediation suggestions, and under complex contexts supplies traceable reasoning chains; Dify offers visual orchestration to enable tool coordination, branch control, and result aggregation, among other engineering capabilities. Together, these components form a “feature extraction—semantic enhancement—automated handling—feedback” closed loop that preserves interpretability while

improving accuracy, generalization, and engineering scalability, thereby providing a practical pathway toward next-generation intelligent email-security systems oriented to enterprise deployment and compliance requirements [9].

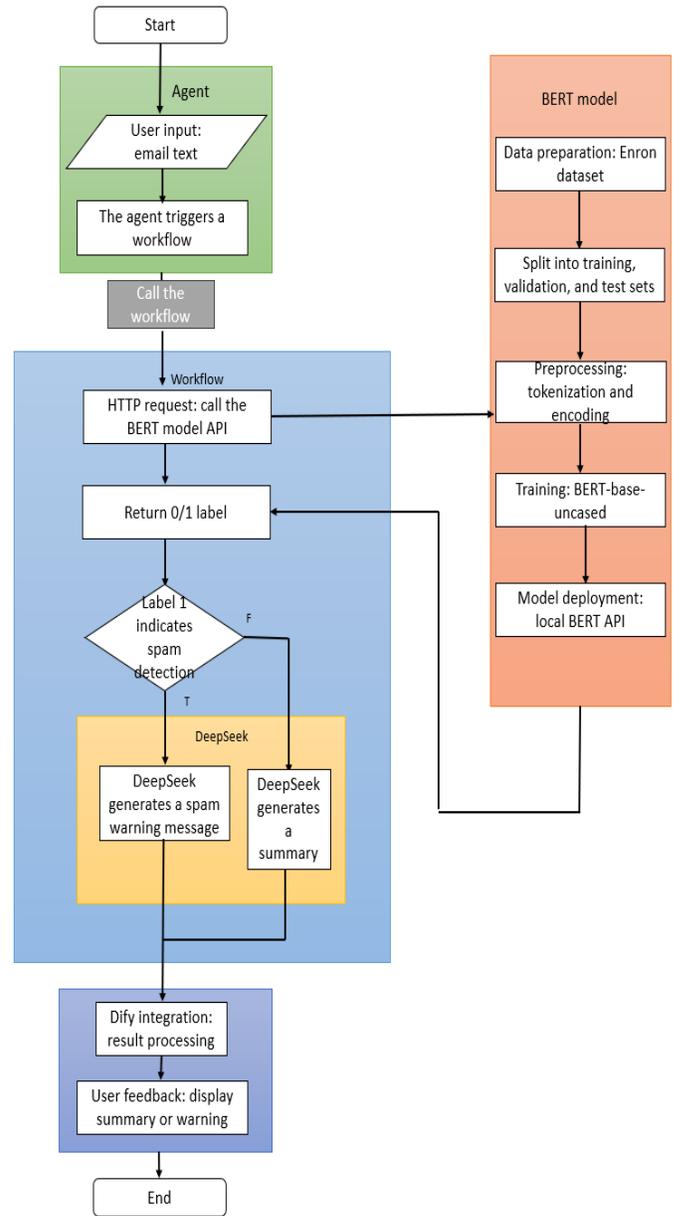


Figure 1. Agentic workflow architecture

III. TECHNICAL MODEL

The proposed spam-detection agent integrates the strengths of DeepSeek, Dify, and BERT. It leverages DeepSeek for semantic understanding

and natural-language generation, Dify for workflow orchestration and agent construction, and BERT for feature extraction and classification, forming an efficient and flexible email detection agent. The system first classifies emails into spam or ham using BERT, then uses the Dify platform together with the DeepSeek model to analyze and summarize the content, ultimately delivering precise feedback.

A. BERT Model

As the core text-classification component, BERT benefits from bidirectional contextual modeling: it captures not only direct word relationships but also longer-range contextual semantics in email text. Trained on large volumes of email data, BERT learns deep discriminative features between spam and ham and performs accurate classification based on semantic content. For each email, BERT outputs a label, where 0 denotes ham and 1 denotes spam. These labels serve as the key decision basis for the downstream agent workflow[5].

B. DeepSeek Model

DeepSeek is a large language model that, in this work, serves as the module for content summarization and feedback generation, with strengths in handling complex context and producing natural-language responses.

- Ham analysis: For legitimate (ham) emails, DeepSeek understands the context and produces a concise summary so users can quickly grasp the core information.
- Spam feedback: For spam emails, DeepSeek is responsible for promptly confirming the spam nature and issuing warning feedback so that users can immediately identify risks and take action.

DeepSeek reasons based on BERT's classification results to provide more intelligent and accurate feedback, rather than merely repeating the class label.

C. Dify Platform

Dify is an agent-development platform that simplifies development and deployment through

visual workflow orchestration and modular design. As one of the core technologies in this study, Dify integrates the BERT and DeepSeek models and enables full automation and intelligence across the spam-detection process. Through Dify's visual workflow builder, users can define the agent's task flow by dragging and connecting nodes. With Dify, the system efficiently schedules BERT and DeepSeek to achieve end-to-end processing—from email input and text analysis to classification and feedback generation.

In this work, the Dify workflow implements automatic parsing of email content, intelligent decision-making, and interactive feedback. Concretely, the system invokes BERT for classification, then invokes DeepSeek for semantic analysis and reasoning, and finally generates user-facing feedback according to the detection result. This workflow design not only simplifies development but also improves flexibility and scalability, making spam detection more intelligent.

The overall workflow, built on the tight collaboration among BERT, DeepSeek, and Dify, proceeds as follows:

- 1) *The user submits the email text.*
- 2) *Via Dify's HTTP request node, the email content is sent to the BERT model for spam classification.*
 - If BERT outputs label 0 (ham), the content is passed to DeepSeek for summarization.
 - If BERT outputs label 1 (spam), the system directly asks DeepSeek to generate a spam warning.
- 3) *DeepSeek generates feedback and returns it to the user.*
 - For ham: produce a concise summary to help the user understand the email quickly.
 - For spam: provide a clear warning and remind the user of potential risks.
- 4) *Email detection and feedback are completed.*

Through the Dify platform, the agent can automatically adapt its behavior according to the classification result: it not only completes spam

detection but also delivers natural-language feedback via DeepSeek, enabling interactive user experience.

IV. EXPERIMENT AND ANALYSIS

A. Experimental Setup

All model training and experiments in this study were conducted locally, accelerated by an NVIDIA RTX 4060 GPU. The integrated development environment is PyCharm 2024.2.4. The Python interpreter is 3.10.8, the deep learning framework is PyTorch 2.5.1, and the CUDA build version is cu12.1.

During the experiments, model training mainly relied on the Transformers 4.50.3 library from Hugging Face to load and fine-tune the BERT model. pandas 2.2.3 and numpy 2.2.4 were used for data processing, while scikit-learn 1.6.1 and matplotlib 3.10.1 were used for model evaluation and visualization. In addition, model inference and agent integration were completed on the Dify platform, enabling simple and efficient system deployment[8].

B. Experimental Procedure

This study uses the email dataset provided by Kaggle, containing a total of 33,715 emails, including 17,170 spam and 16,545 ham. The dataset was split into training, validation, and test sets in a 70%: 15%: 15% ratio. The distribution is shown in Table 1.

TABLE I. DATASET DISTRIBUTION

	Training Set	Validation Set	Test Set
Spam	12,059	2,540	2,571
Ham	11,541	2,517	2,487

Fine-tuning adopts the bert-base-uncased pretrained model for the spam-classification task. BertTokenizer is used to encode email text, with batch size 16, learning rate 1e-5, maximum input length 512, 3 training epochs, the AdamW optimizer, and cross-entropy loss. After each epoch, the model is evaluated on the validation set, and accuracy is recorded to monitor performance.

C. Results and Analysis

1) Training results analysis.

During training, BERT showed rapid convergence. By the end of the third training epoch, validation accuracy reached 99.49%, and the loss converged to 0.0045, indicating that the model successfully learned discriminative features for email classification without overfitting. The steady improvement of validation accuracy suggests strong generalization on unseen data. Given BERT's strong feature-extraction capability, we set 3 training epochs to avoid over-training and the risk of overfitting[6].

2) Test results analysis

We compare the BERT model with traditional machine-learning models (Naive Bayes, Decision Tree, and LinearSVC) on the spam-classification task. Table 2 presents the performance comparison across different models.

TABLE II. MODEL EVALUATION METRICS

model	Acc(%)	Average Precision(%)	Average Recall(%)	Average F1-Score(%)
MultinomialNB	98.08	97.61	98.64	98.12
Decision Tree	95.89	96.03	95.88	95.95
LinearSVC	98.99	98.69	99.34	99.01
BERT	99.49	99.42	99.57	99.49

BERT performs excellently across accuracy, precision, recall, and F1, and it substantially surpasses traditional machine-learning models in capturing deep semantics and contextual information. This indicates that BERT has significant advantages for spam classification, effectively reducing both missed detections and false alarms. We also plot the confusion matrix of the BERT model on the test set (see Figure 2) to intuitively demonstrate per-class performance[7].

From the confusion matrix, the BERT model correctly classifies most spam (2,560) and ham (2,472) emails. Although a small number of misclassifications exist (15 ham misclassified as spam and 11 spam misclassified as ham), the overall classification performance remains excellent, showing high accuracy and reliability.

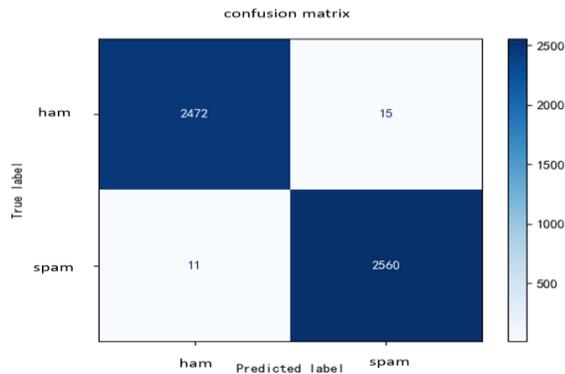


Figure 2. Confusion matrix of BERT on the test set

V. CONSTRUCTION AND IMPLEMENTATION OF THE AGENT

A. Environment Setup for the Dify Platform

To build a local agent platform with tool-calling capability, we deploy the open-source Dify platform. Dify supports multi-model, multi-function integration and offers a user-friendly

interface with comprehensive features. Because the platform depends on multiple software components, we use Docker for containerized deployment to enable one-click configuration and rapid startup.

First, download the latest Dify source code package from the official GitHub repository and extract it to a local path. Navigate to the `docker/` folder in the project directory, rename the provided `.env.example` file to `.env`, and adjust environment variables (e.g., port settings) as needed. Then, in the directory where `.env` resides, open a terminal and run `docker compose up -d` to start the platform.

Docker greatly simplifies deployment: compared with manual installation and configuration, it provides isolated, modular environments that avoid version conflicts and improve portability and maintainability. All required images are pulled automatically on first run and corresponding local containers are started (see Figure 3).

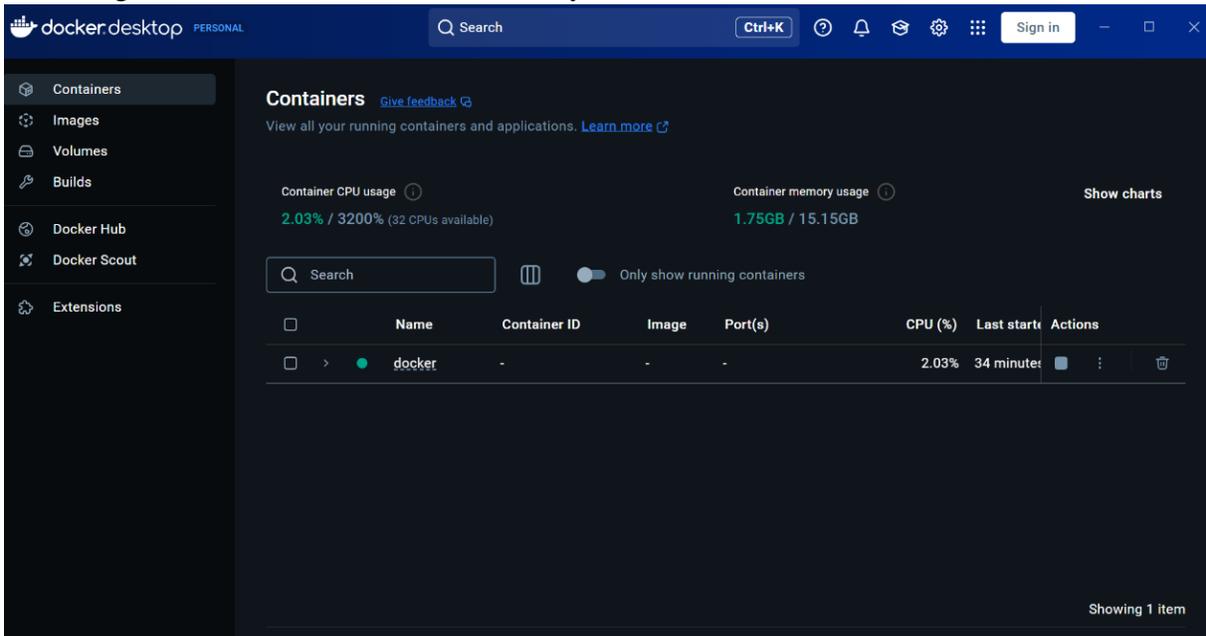


Figure 3. Docker containers running locally

By following the above steps, we successfully deploy and run Dify locally, laying the foundation for integrating large language models and building task-executing agents. The platform can be accessed by visiting `localhost/apps` in a browser.

B. DeepSeek-R1 API Integration

After deploying Dify and verifying that it runs correctly, we integrate the DeepSeek-R1 model via its hosted, full-capability API to endow the

agent with stronger language understanding and tool-use capabilities. DeepSeek-R1, provided by Hangzhou DeepSeek AI, supports function calling, multi-turn dialogue, and structured outputs with robust tool-calling and contextual reasoning. The model is offered as an API service; developers obtain an API key to invoke it. Dify loads the DeepSeek model through the API as shown in Figure 4.

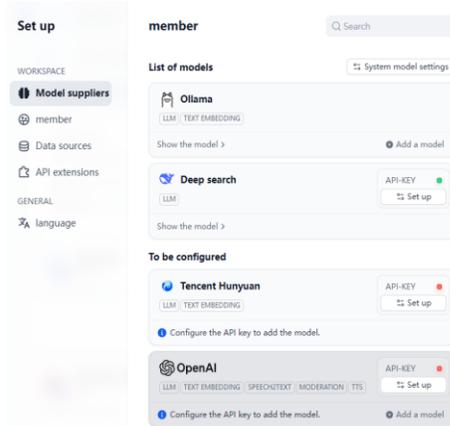


Figure 4. Loading the DeepSeek model via API in Dify

C. Workflow Construction

After deploying the Dify platform and integrating the DeepSeek-R1 API, we designed an end-to-end agent workflow following a “discriminate–reason–feedback” paradigm to achieve automatic spam identification with explainable responses. The pipeline comprises four categories of nodes: (1) input and preprocessing, (2) classification and decision, (3) semantic generation and safety prompts, and (4) result aggregation and output. At its core, the locally hosted BERT classification service and the DeepSeek-R1 language model are decoupled and interconnected via Dify’s flow-based orchestration, which ensures stability and controllability in the classification stage while enhancing user-readable explanations and summaries, thereby forming a closed loop from content ingestion to decision and feedback (see Figure 5).

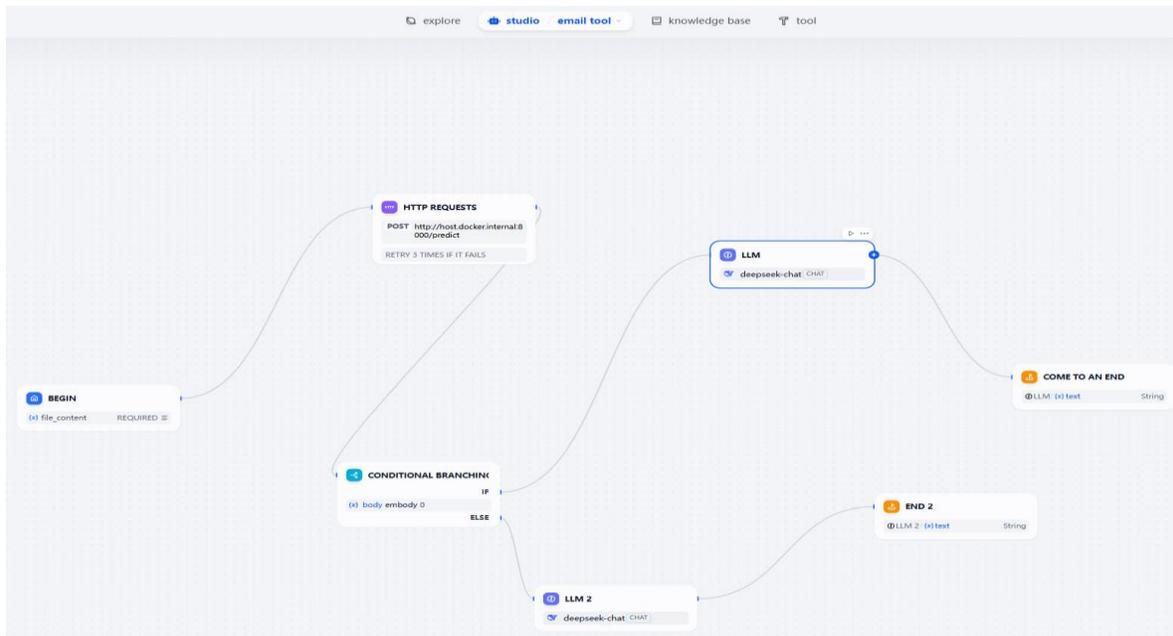


Figure 5. Workflow for the email detection tool

The workflow starts at the Start node, where the user inputs an email text as the pipeline’s input. The text is then passed to an HTTP Request node,

which submits the raw content via POST to the locally deployed BERT classification API (Content-Type: text/plain). The model performs

binary classification, returning 0 for ham and 1 for spam. Testing the API with a spam email yields a return value of 1 (see Figure 6).

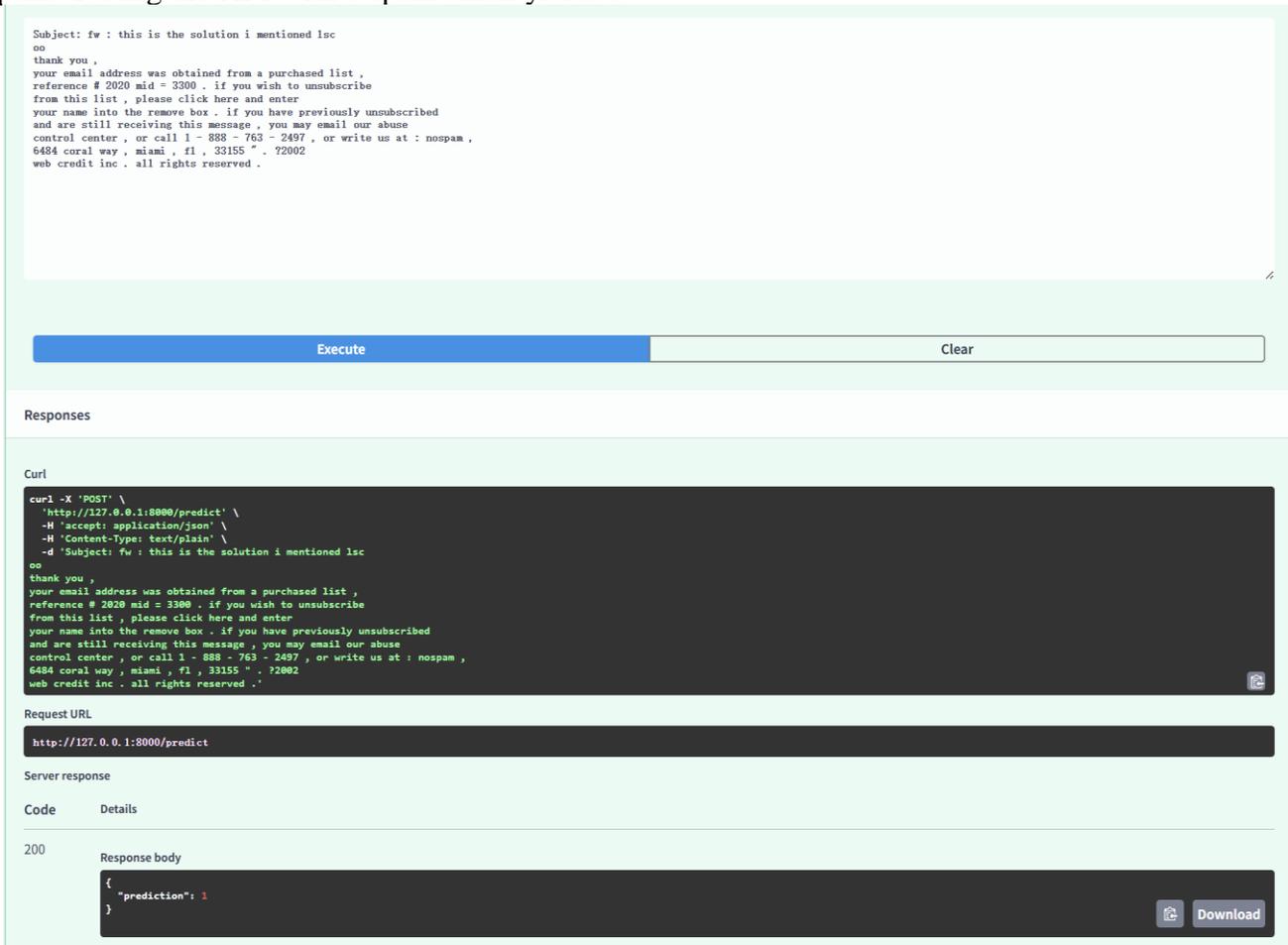


Figure 6. BERT API endpoint test

Next, the workflow enters a Conditional (If/Else) node. Based on the HTTP response body, if the result contains 0, the system routes to the ham branch; if it contains 1, it routes to the spam branch.

- In the ham branch, the workflow calls a DeepSeek-R1 node with English prompts that first inform the user “this is a normal email,” then generate a concise summary so the user can quickly grasp the intent.
- In the spam branch, the DeepSeek-R1 node explicitly states “this is a spam email,” provides a brief explanation, and issues safety warnings to enhance user awareness.

Both branches output their responses via the respective model nodes and connect to the

workflow’s end nodes, marking the completion of the process. This workflow not only distinguishes spam from ham accurately, but also uses the LLM to generate natural-language feedback, yielding a more interactive and user-friendly presentation.

Overall, the workflow achieves effective cooperation between a local traditional model and a cloud LLM. Combined with Dify’s graphical orchestration, it makes complex agent construction more modular and visual, providing a solid framework for future extensions such as multimodal inputs and plugin/tool integrations.

D. Agent Testing and Applications

1) Agent Deployment

After completing the construction of the spam-

detection flow, we created an agent instance named “Email Detection Agent” on the Dify platform to enable interactive email security detection. This agent integrates the previously built “email tool” workflow, allowing users to interact with the model in natural language and trigger the full email analysis process. To improve the model’s understanding and task triggering accuracy, we set the following system prompts: 1) Think and respond in English. 2) Use a tool to perform email verification.

These prompts help the large language model (DeepSeek-R1) clearly understand its role—to respond in English and to actively invoke the bound tool to analyze and judge email content.

To further optimize user experience, the agent is configured with the following greeting to guide user operations: Hello, I’m your intelligent email

detection assistant. I can automatically analyze and identify spam emails in your inbox, helping you save time and avoid risks. Whether it’s promotional ads, phishing emails, or suspicious links, I will promptly screen and label them to ensure your email security and communication efficiency. Please tell me which emails you’d like to check, or send me the email content — I’ll start analyzing right away.

With this opening message, the agent presents a professional and trustworthy persona and clarifies how users should proceed. Overall, the deployment process is simple and efficient; combined with Dify’s graphical configuration interface, it enables rapid model integration, tool binding, and instruction-template setup, laying a solid foundation for subsequent interactive testing. The created agent is shown in Figure 7.

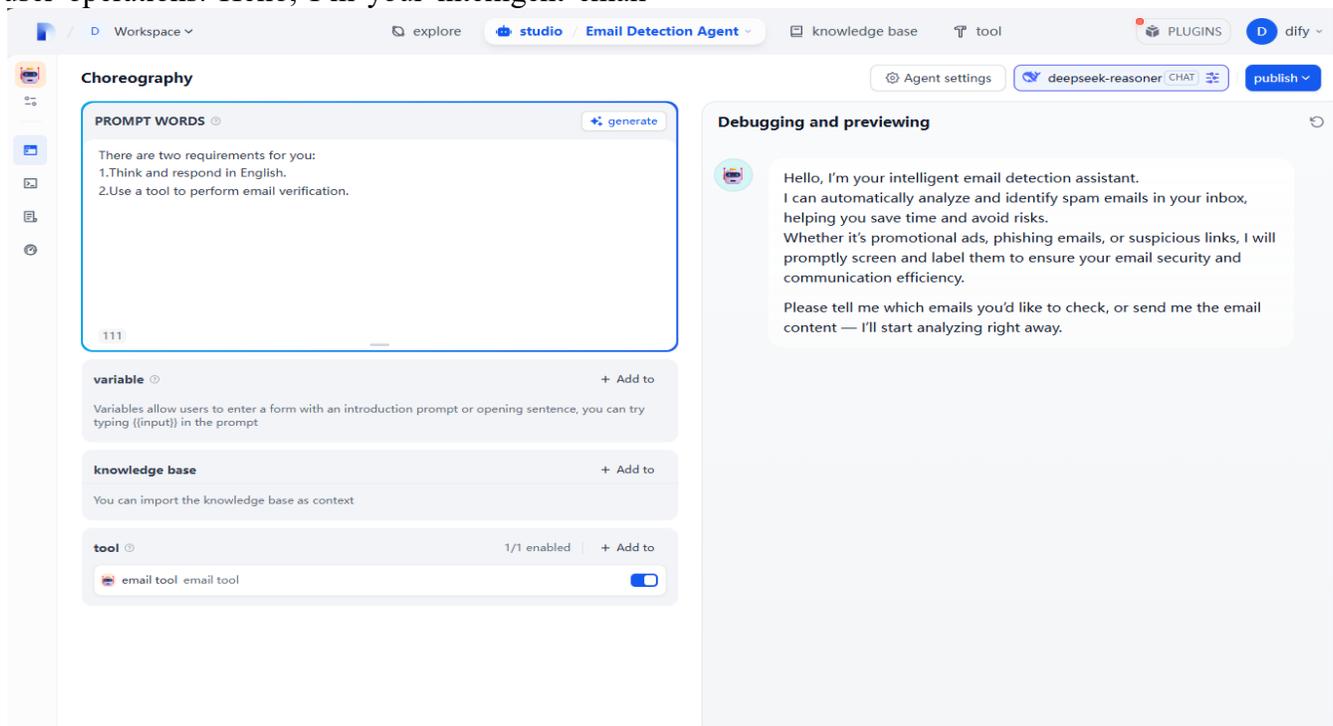


Figure 7. Agent interface

2) Ham Email Detection

To evaluate the agent’s ability to recognize non-spam (ham) emails, we use a notice about personnel changes in the GSS (Global Strategic Sourcing) organization as input. The email introduces three new members—Tracy Ramsey, Cheryl Slone, and Amanda Becher—and briefly

outlines their backgrounds and responsibilities. Using the Email Detection Agent configured on the Dify platform, the user submits the raw email text to the agent interface; the system then automatically invokes the email tool according to preset prompts to process the message. The agent’s reasoning trace is shown in Figure 8.

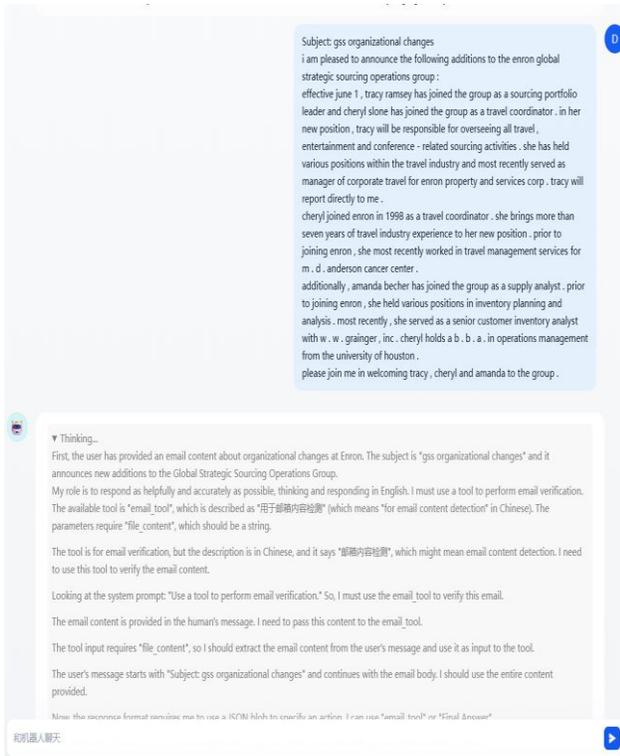


Figure 8. Reasoning view for normal-email detection

The analysis is led by the DeepSeek-R1 node, which forwards the content via a bound HTTP endpoint to the locally deployed BERT classifier. Based on the email content, BERT returns 0 (non-spam). The workflow then routes to DeepSeek-R1’s “ham” branch to generate a structured response. This node is guided by English task prompts and is required to produce a concise, English summary and feedback according to the classification result. The tool-calling trace is shown in Figure 9.

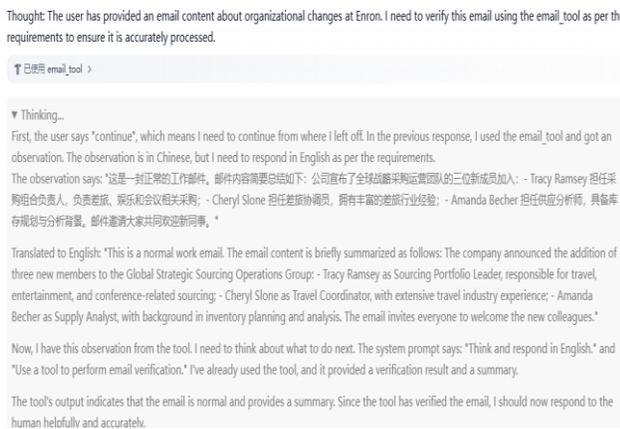


Figure 9. Tool-calling trace for normal-email detection

In this run, the system confirms that the tool call succeeded and BERT returned the label 0 (normal). The agent then performs semantic analysis and consistency checks, confirming that the email concerns organizational restructuring with newly added roles and appointments. The final reply provides a concise English summary of key points, and no formatting or security risks are detected. The final result is shown in Figure 10.

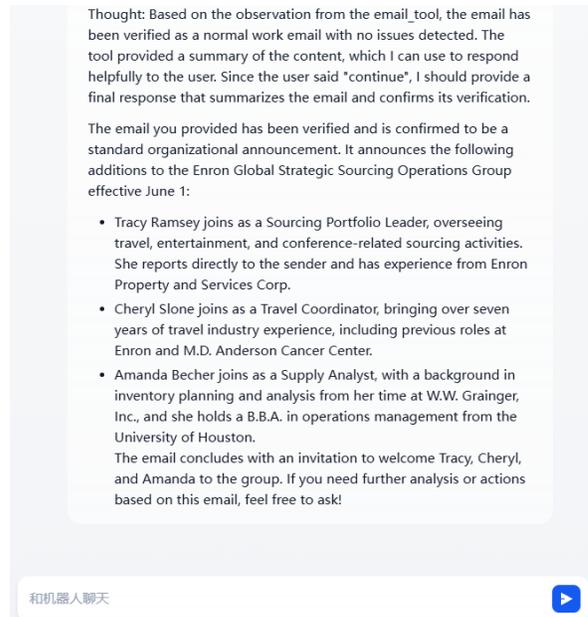


Figure 10. Result of normal-email detection

In summary, during this test the Email Detection Agent correctly identified the message as normal (non-spam) and produced a clear English summary, demonstrating strong comprehension and text-analysis capabilities that support practical use.

3) Spam Email Detection

To evaluate the agent’s ability to identify spam, we use a typical promotional email as input. The subject is “Too many credit card bills! – this fixes that!” The body contains numerous credit-card debt-relief claims—such as “reduce monthly payment up to 60%,” “no credit check,” “no collateral,” and “handled by specialists”—which are characteristic of deceptive advertising. In addition, the unsubscribe link is a raw IP address (<http://65.162.84.5/...>) rather than a legitimate domain, posing a higher security risk. The agent’s reasoning view is shown in Figure 11.

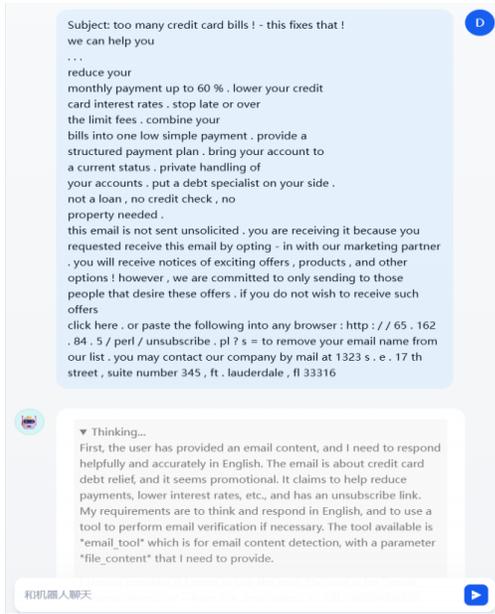


Figure 11. Reasoning view for spam-email detection

Using the Email Detection Agent on the Dify platform, the user submits the original email text to the system. Guided by its task prompts, the DeepSeek-R1 node automatically invokes the email tool and forwards the content to the locally trained BERT classifier. The model returns “1”, indicating spam. The workflow then routes to the spam branch, where DeepSeek-R1 generates the final response. The tool-calling trace is shown in Figure 12.

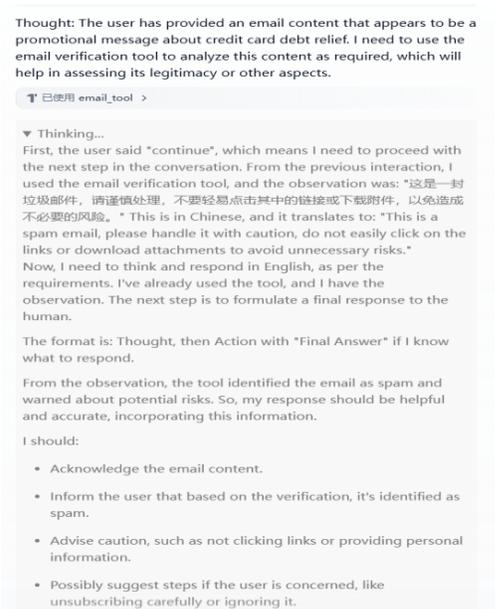


Figure 12. Tool-calling trace for spam-email detection

The results show that the system flags the message as spam and issues general safety advice: do not click any links, do not download attachments or provide personal information; if you did not actively subscribe, it is advisable to ignore or delete the email; for debt-relief consultation, contact accredited institutions or certified financial advisors first. The final output is shown in Figure 13.



Figure 13. Result of spam-email detection

Overall, the Email Detection Agent effectively identifies this type of high-risk promotional email and produces clear English safety guidance, demonstrating good practicality and scenario adaptability.

VI. CONCLUSION AND FUTURE WORK

This paper presents an email detection agent built through the collaboration of BERT, DeepSeek-R1, and Dify. Experiments show that the agent can semantically understand and classify email content, dynamically invoke appropriate tools based on the classification result, and provide explanatory feedback consistent with human reasoning, demonstrating strong practicality and intelligence.

Despite these initial results, several limitations remain. First, overall accuracy depends on the training quality of the local BERT model and may be affected by dataset distribution bias. Second, the current system does not yet handle non-text content such as attachments or images; future extensions should broaden the detection scope to improve coverage.

Future work will focus on: (1) introducing multimodal models to jointly recognize and assess multiple sources of information in emails (e.g., images and attachments); (2) exploring agent self-adaptation and online learning to improve robustness against emerging spam patterns; and (3) integrating real-time threat intelligence to build an LLM-driven email security system that deeply integrates with enterprise mail services, achieving an intelligent, explainable, and deployable solution.

REFERENCES

- [1] Li, Aihua; Wei, Ying; Liu, Jun. Practice and Reflections on the Local Deployment of the DeepSeek Large Model at the Changping District Archives. *Beijing Archives*, 2025(04): 61–64.
- [2] Zhang, Wenjie. Prompt Governance: A Human–AI Collaborative Model for Content Generation in Domestic LLMs such as DeepSeek. *Journal of Soochow University (Philosophy & Social Science Edition)*, pp. 1–12 [2025-04-28].
- [3] Cheng, Jianning. Applications of AI Large Models in the Field of Tendering and Procurement: Insights from DeepSeek. *China Tendering*, 2025(04): 20–25.
- [4] Liu, Cheng; Mo, Shengye. Clarifying Ideological-Security Boundaries, Risk Monitoring, and Regulatory Logic in Generative AI Applications—Reflections Based on DeepSeek, Manus, ChatGPT, and Sora. *Journal of Chongqing University (Social Science Edition)*, pp. 1–14 [2025-04-28].
- [5] Peng, Chaoyun; Yao, Miaobo; Zeng, Dongchen. A Hybrid Recommendation System for Person–Job Matching Based on the BERT Model. *Information Recording Materials*, 2025, 26(04): 236–238.
- [6] Shi, Zhixiong; Duan, Gaijia; Ma, Longxuan; et al. Optimizing AI Question Answering via LLM-Based Named Entity Recognition. *Posts & Telecommunications Design Technology*, 2025(03): 80–84.
- [7] Gongque, Duojie; Suonang, Cairang; Caizang, Tai. Tibetan Sentence Sentiment Classification Using Dictionary-Augmented BERT-BiGRU. *Computer Engineering and Design*, 2025, 46(03): 918–926.
- [8] Jin, Shucheng; Wang, Jiamei. Sentiment Analysis of Customer Reviews Based on a BERT–MABL Model. *Journal of Yunnan Minzu University (Natural Science Edition)*, pp. 1–11 [2025-04-28].
- [9] Gao, Yubo; He, Ping; Zhou, Zheng; et al. Design and Implementation of an Agent-Based Q&A System for G-Code Generation. *Mini-Micro Systems*, pp. 1–11 [2025-04-28].
- [10] Huang, Shuangbin; Wang, Meijia; Gao, Liuyang. Question Classification for Cattle-Disease Queries Based on BERT DPCNN. *Intelligent Computer and Applications*, 2025, 15(03): 140–144.