Risk-MCTS: Table-Reward Enhanced LLM with Monte Carlo Tree Search for Interpretable Financial Risk Detection

Anonymous ACL submission

Abstract

Financial risk detection is an important yet challenging task. Existing machine learning or deep learning-based approaches have primarily treated it as a binary classification task. Although these approaches already achieved good model performance, they still fail to capture complex risk patterns as well as to provide interpretable steps for financial risk detection. To address aforementioned research limitations, we propose this Risk-MCTS, a novel framework integrating large language model with monte-carlo tree search method, which leverages both cell data and headers in financial tables for step-by-step risk inference. To better understanding financial tabular data, we carefully design a table reward model which quantitatively evaluates table content during the analytical process, thereby enhancing the detection of salient financial content. Extensive experiments demonstrate that the proposed Risk-MCTS achieves the SOTA model performance on real world datasets with respect to a number of evaluation criteria.

1 Introduction

Financial risk detection (Dyck et al., 2023) has long been investigated in the domain of artificial intelligence, which is an important yet challenging task (Wang et al., 2024a). Existing approaches could be roughly classified into two categories: tabular data analysis-based approaches and textual data analysis-based approaches.

For tabular data analysis approaches, most of the approaches are machine learning or deep learning-based approaches (Bao et al., 2019; Dechow et al., 2010; Cecchini et al., 2010). They often treat financial risk detection as a binary classification task using numerical features extracted from financial statements. For textual data analysis approaches, a few natural language processing approaches have been proposed for analyzing such as financial reports and financial news. For instance, (Xiuguo and Shengyong, 2022) analyzes the Management Discussion and Analysis (MD&A) sections and (Craja et al., 2020) combining quantitative data with textual features.

Obviously, aforementioned approaches have three fundamental limitations: **limited semantic understanding ability** for tabular data, **uninterpretable black-box model design** and **high annotation cost** needed for model training. First for **limited semantic understanding ability** for tabular data, existing approaches treat financial statements as pure numerical data, failing to detect the complex risky patterns hidden across multiple table columns, rows, headers or even across multiple tables.(Woźnica et al., 2023; Cheng et al., 2024) Second for **uninterpretable black-box model design**, most existing approaches typically design a non-linear high-dimensional kernel mapping functions or a non-linear neural activation layer, which fails to explain how the detection results are generated or lacks an interpretable reasoning process with detailed steps.(Samek and Müller, 2019; Zhou et al., 2015) Third for **high annotation cost** challenge, it is well known that most conventional approaches need supervised training data as more as possible to achieve the SOTA performance. However, real-world financial risk cases are inherently limited and seriously imbalanced which limits existing model performance.

To address these challenges, we propose this **Risk-MCTS** approach, which adapts Monte Carlo Tree Search for financial risk detection by designing domain-specific components enabling the employed LLMs to analyze financial statements (Figure 1). Unlike previous LLM-MCTS approaches which focuses on general reasoning tasks, our framework is specifically designed to cope with the complex financial tabular data through following components. **A policy model** reformulates risk detection as a structural reasoning process. This model decomposes complex financial evaluation

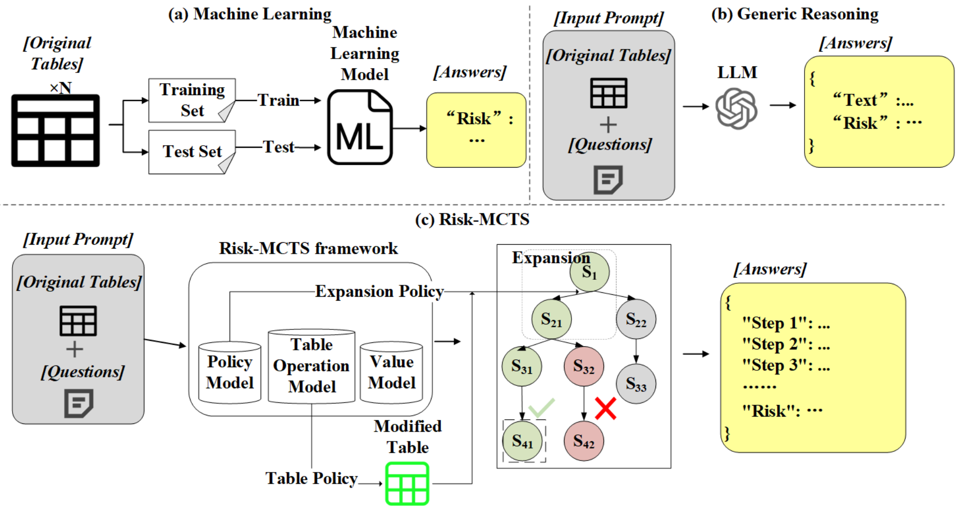

Figure 1: Comparison between 3 financial risk detection methods. (a) Machine Learning method requires training and output answers directly without intermediate steps. (b) Generic Reasoning with LLM can perform zero-shot inference and give simple intermediate steps. (c) Risk-MCTS construct a Monte-Carlo search tree iteratively, it can perform zero-shot inference and give intermediate steps in detail.

into interpretable sub-questions, enabling LLMs to analyze both quantitative patterns and semantic relationships hidden among the input financial statements. **A table operation model** enhances data analysis capabilities through a structured operation pool. Inspired by (Wang et al., 2024b) and (Ji et al., 2024), our model combines basic table operations with financial operations, providing LLMs with the necessary tools to process structured data systematically. **A value model** guides the tree search process by evaluating the quality of intermediate reasoning steps. This enables the framework to construct verifiable reasoning chains for risk assessment. Through the Monte Carlo Tree Search process, inspired by (Qin et al., 2024) and (Huang et al., 2024), our framework iteratively constructs a solution space which balances exploration of different analytical paths with exploitation of promising reasoning directions. This approach enables effective risk detection even with limited training data, as it leverages the pre-trained knowledge in LLMs while maintaining interpretable decision processes.

Our contributions are summarized as below:

- We propose Risk-MCTS, a framework that integrates LLMs with Monte Carlo Tree Search for financial risk detection, providing step-by-step reasoning using both numerical and semantic information.

- We develop a table operation model that processes financial data through a combined pool of tabular and financial operations. We design a value-guided search mechanism that evaluates intermediate reasoning steps by combining probability analysis with domain knowledge.

- We evaluate Risk-MCTS on AAER and CSMAR datasets, showing improved performance over both machine learning-based approaches and the SOTA LLMs while maintaining interpretable reasoning.

2 Related Work

Financial Risk Detection Prior works on financial risk detection (Dechow et al., 2010; Cecchini et al., 2010; Bao et al., 2019) primarily focused on machine learning approaches for analyzing tabular financial data. Most recently, methods leveraging large language models have shown promise in this domain. (Yang et al., 2023) demonstrated improved risk detection by optimizing model performance on key financial terminology. However, to the best of our knowledge, no prior work has proposed a comprehensive LLM-based framework specifically designed for financial risk detection that maintains both accuracy and interpretability.

MCTS-based Long Reasoning LLM The challenge of enabling LLMs to perform complex, multi-step reasoning has gained significant attention since the release of OpenAI-o1 and Deepseek-R1, with numerous works emerging to enhance LLMs' long-form reasoning capabilities (Qin et al., 2024; Huang et al., 2024; DeepSeek-AI et al., 2025). Among these approaches, Monte Carlo Tree Search

(MCTS) has emerged as a particularly promising framework for improving LLM reasoning. (Zhang et al., 2024a) pioneered this direction by proposing a self-training framework that leverages MCTS to generate trajectories from previous iterations, using these to train the LLM and achieve enhanced performance. Building on this work, (Zhao et al., 2024) demonstrated how integrating LLMs with MCTS could substantially improve reasoning capabilities across various tasks. In the specific context of tabular data, (Ji et al., 2024; Deng et al., 2023) have made some efforts but still lack domain-specific capabilities needed for financial risk detection.

3 The Proposed Approach

3.1 Overview

We propose Risk-MCTS, a framework that combines Monte Carlo Tree Search with large language models to address key challenges in financial risk detection. Unlike traditional methods that treat risk detection as binary classification, our approach decomposes complex financial analysis into interpretable reasoning steps. Following (Lan et al., 2019; Luo et al., 2024; Zhang et al., 2024a), we adapt the Monte Carlo Tree Search methodology to financial risk detection through three key innovations: (1) **Domain-Specific Tree Construction**: While existing MCTS-LLM approaches (Zhang et al., 2024c,b) focus on general reasoning, Risk-MCTS constructs a specialized reasoning tree for financial analysis, implementing the four MCTS processes (Chaslot et al., 2008) with financial domain-specific components. (2) **Table-Reward Enhanced Evaluation**: Inspired by (Wang et al., 2024b) and (Ji et al., 2024), we develop a specialized table reward mechanism that combines probability analysis of LLM outputs with financial domain knowledge for more reliable evaluation of intermediate reasoning steps. (3) **Structured Operation Space**: Risk-MCTS constrains exploration through a carefully designed operation pool that combines basic table operations with financial computations, ensuring all reasoning steps are grounded in valid financial analysis.

3.2 Problem Formulation

Given a company's financial statements, we formalize the risk detection task as a structured reasoning problem over tabular financial data. Let $\mathcal{F} = \{T_1, ..., T_M\}$ be the set of financial tables, where each table T_i consists of paired header and data elements:

$$T_i = \{(h_j, t_j)\}_{j=1}^{N_i} \quad (1)$$

where h_j represents the header of the j-th column, t_j represents the corresponding numerical data, and N_i is the number of columns in table T_i.

The objective is to construct a reasoning chain $C = \{(Q_i, T_i)\}_{i=1}^{N}$ through Monte Carlo Tree Search, where each step s_i consists of: a question-answer pair Q_i analyzing specific financial aspects, a table operation sequence $\mathcal{O}_i$ processing relevant financial data an intermediate conclusion supporting the final risk assessment. At each step i:

$$Q_{i+1} = \pi(Q_i, T_i)$$
$$\mathcal{O}_{i+1} = \tau(Q_i, T_i)$$
$$T_{i+1} = \text{Execute}(\mathcal{O}_{i+1}, T_i)$$

where π is the policy model generating the next analytical question, τ is the table operation model selecting appropriate financial operations, Execute applies the selected operations to produce updated table views.

The final output is a binary risk assessment $y \in \{0, 1\}$ along with the complete reasoning chain C documenting the analytical process.

3.3 Framework Components

Policy Model guides the systematic exploration of financial risk indicators through structured reasoning. Formally, given the current state (Q_i, T_i), the policy model π generates the next partial solutions step:

$$Q_{i+1} = \pi(Q_i, T_i) \quad (2)$$

The process of building a Monte Carlo search tree requires the Policy Model to generate several different answers based on the same input prompt. As a consequence, we set temperature of Policy Model to be greater than 0 to encourage diversity in the generated reasoning paths.

Table Operation Model aims to enhance the understanding of tabular data through a structured operation space $\mathcal{O}$ designed specifically for financial statement analysis. The operation space consists of two hierarchical levels:

$$\mathcal{O} = \mathcal{O}_{\text{basic}} \cup \mathcal{O}_{\text{financial}} \quad (3)$$

where basic operations handle table manipulation:

$$\mathcal{O}_{\text{basic}} = \{\text{SELECT}, \text{FILTER}, \text{JOIN}...\} \quad (4)$$

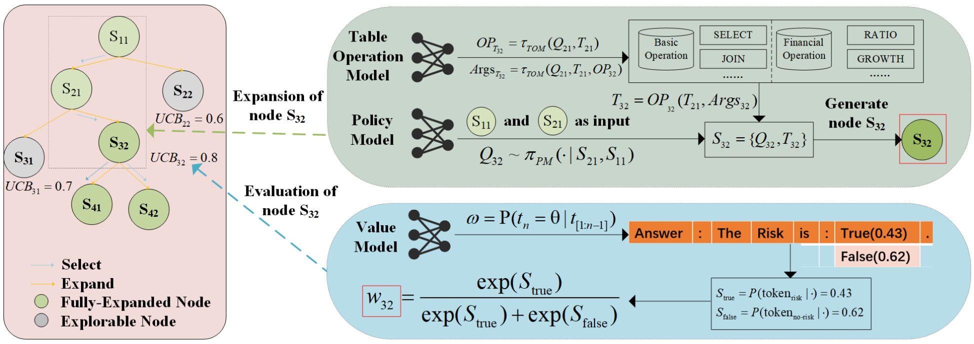

Figure 2: Construction of the Monte-Carlo search tree in Risk-MCTS. The framework consists of four phases: Selection using UCB scores (Equation 16, Expansion where Policy Model generates reasoning steps (Equation 2), Evaluation through the probability-based scoring, and Backpropagation. Right upper box: expansion phase demonstrates how Policy Model and Table Operation Model interact via operations. Right lower box: evaluation phase illustrates the value estimation process. Solid arrows indicate operations; dashed arrows show information flow.

and financial operations compute domain-specific metrics:

$$\mathcal{O}_{\text{financial}} = \{\text{RATIO}, \text{YOY}, \text{GROWTH}...\} \quad (5)$$

The hierarchical design of the operation space reflects two key considerations: (1) Basic operations ensure fundamental table manipulation capabilities that are essential for any financial analysis, while (2) financial operations encode domain-specific computations commonly used in risk assessment. This separation allows the model to combine generic table processing with specialized financial analysis in a structured manner.

For each analytical step, the table operation model τ performs two sequential decisions, the first is Operation Selection:

$$OP_{T_{i+1}} = \tau(Q_i, T_i) \in \mathcal{O} \quad (6)$$

The second is Argument Generation:

$$\text{Args}_{T_{i+1}} = \tau(Q_i, T_i, OP_{T_{i+1}}) \in \mathcal{A}(OP_{T_{i+1}}) \quad (7)$$

where $\mathcal{A}(OP_{T_{i+1}})$ defines the valid argument space for the selected operation. For example:

$$\mathcal{A}(\text{RATIO}) = \{(x, y)|x, y \in T_i, y \neq 0\} \quad (8)$$

$$\mathcal{A}(\text{YOY}) = \{x|x \in T_i, \exists x_{t-1}\} \quad (9)$$

The selected operation and arguments are then executed through a parser to obtain a new sub-table:

$$T_{i+1} = \text{Parser}(OP_{T_{i+1}}(T_i, \text{Args}_{T_{i+1}})) \quad (10)$$

This structured approach ensures that all operations are financially meaningful, arguments satisfy operational constraints and the results maintain data consistency.

Value Model evaluates the quality of intermediate generated steps through a probability-based approach that combines LLM confidence with financial domain knowledge. Rather than directly generating numerical scores, our approach leverages the LLM's next-token detection distribution for more reliable evaluation. This design is motivated by two factors: (1) LLMs typically show better calibration in their token probabilities compared to direct numerical outputs, and (2) the distribution over possible outcomes provides richer information for guiding the search process.

Formally, we define a probability distribution vector ω over the model's vocabulary, computed from the LLM's token logits:

$$\omega = \mathcal{P}(t_n = \theta | t_{[1:n-1]}), \forall \theta \in vocab \quad (11)$$

where $t_{[1:n-1]}$ represents the context tokens and $vocab$ represents vocabulary of the Policy Model. This distribution is normalized through a softmax function:

$$\sigma(\mathbf{z}_j) = \frac{exp(z_j)}{\sum_{k=1}^{K} exp(z_k)}, j = 1, ..., K \quad (12)$$

For financial risk assessment, we specifically focus on the probabilities of risk-indicating tokens:

$$S_{\text{true}} = P(\text{token}_{\text{risk}}|\cdot) \quad (13)$$

$$S_{\text{false}} = P(\text{token}_{\text{no-risk}}|\cdot) \quad (14)$$

These probabilities are combined into a quality score through a modified softmax function:

$$w = \frac{exp(S_{true})}{exp(S_{true}) + exp(S_{false})} \quad (15)$$

where w is quality value of current step.

	AUC	Recall	F1-score	balanced accuracy	Intermediate
LLM					
Llama3.3-70b	0.551	-	0.185	0.551	-
Qwen2-72b	0.610	-	0.701	0.554	-
DeepSeek-R1	<u>0.736</u>	-	0.723	0.612	-
Grok-3	0.630	-	0.729	0.364	-
TableGpt2-7b	0.266	-	0.372	0.436	-
COT					
Llama3-70b-cot	0.662	-	<u>0.853</u>	0.677	2.48
Qwen2-72b-cot	0.703	-	0.736	<u>0.797</u>	<u>2.73</u>
DeepSeek-R1-cot	0.735	-	0.785	0.642	<u>2.79</u>
Grok-3-cot	0.649	-	0.773	0.637	2.41
TableGPT2-7b-cot	0.544	-	0.569	0.627	2.30
Machine Learning					
NeuralNet	0.718	-	0.882	0.505	-
RandomForest	0.731	-	0.857	0.502	-
XGBoost	0.667	-	0.849	0.499	-
Ours					
Risk-MCTS	**0.814**	-	**0.893**	**0.870**	**3.31**
↑	10.6%	-	3.7%	9.2%	18.6%

Table 1: AAER evaluation results. **Intermediate** refers to the human evaluation score of intermediate steps. Because there are only 0.66% positive samples in AAER dataset, **Recall** may not reflect the full ability of models in this dataset. The ↑ refers to the improvement compared to the LLM-based approach.

3.4 Monte Carlo Tree Search

Our framework adapts the classical MCTS algorithm for financial risk detection through domain-specific selection, expansion, evaluation, and back-propagation strategies. Each node in the search tree represents a state (Q_i, T_i).

Selection In the selection stage, we select the node that is most worth expanding as root node to expand. We use the Upper Confidence Bound (UCB) as an indicator to determine whether a node is worth expanding.

UCB are designed to balance between the exploitation of high value steps and the exploration of less explored steps.

$$UCB = w_{exploration} + c * V_{exploitation} \tag{16}$$

where c is is the exploration hyper-parameter.

In the traditional MCTS process, $V_{exploitation}$ is often calculated using simulations, which performs poorly when the problem is too complex. In our approach, we improve the UCB formula to better adapt the financial tabular data by changing $V_{exploration}$ and $V_{exploitation}$.

$$V_{exploitation} = \mu(T_{node}, D_{node}) \tag{17}$$

$$V_{exploration} = \sqrt{\frac{\ln \text{visit\_num}_{node}}{\sum_{child} \text{visit\_num}_{child}}} \tag{18}$$

where T_{node} and D_{node} denotes the table and question of the node respectively, visit_num denotes the total number of visits to the node. Traditional MCTS implementations often rely on random rollouts for evaluation, which becomes inefficient for complex financial analysis. Our adaptation replaces rollouts with the Value Model's probability-based scoring, making the search process more focused and computationally efficient while maintaining the exploration-exploitation balance through the UCB mechanism.

Expansion After selecting the best node R, Unless R reaching the end stage, we create child nodes of R as new leaf nodes.

In Risk-MCTS, the policy model and table operation model is responsible for node expansion. We prompt policy model and table operation model to generate the next sub-solution D_{i+1} and sub-table T_{i+1} based on D_i and T_i using Equation 2 and Equation 10.

Evaluation Traditional MCTS uses roll-out policy to evaluate the quality of current step, which is ineffective in a LLM-based MCTS framework. In our approach, we use value model to judge the quality of one step. we prompt value model to generate quality of current step w_{i+1}. W_{i+1} will be used in Selection phase.

Back-propagation The final phase updates node values from the evaluated leaf node to the root. Using the value w_{i+1} from the Evaluation phase, we recursively update the quality score w for each node along the path:

$$w_{\text{new}}(Q_i, T_i) = \frac{N(Q_i, T_i) \cdot w(Q_i, T_i) + w_{i+1}}{N(Q_i, T_i) + 1} \tag{19}$$

where $N(Q_i, T_i)$ denotes the visit count of state (Q_i, T_i). This update rule ensures that frequently visited promising paths maintain higher scores while allowing for exploration of alternative reasoning chains.

4 Experiments

4.1 Datasets

We evaluate our approach on two complementary financial risk datasets: the SEC's AAER dataset from the U.S. market and the CSMAR dataset from the Chinese market. Both contain standard financial statements (balance sheet, income statement, and cash flow statement) that are publicly available.

Following prior work (Bao et al., 2019), we use the material accounting misstatements from SEC's AAERs, containing 146,044 samples with 42 financial attributes each. The CSMAR dataset provides Chinese financial data and violation information from CSRC's Enforcement Actions. After filtering columns with >20% missing values, this dataset contains 225,000 samples with 248 financial attributes per sample.

4.2 Model Architecture and Baselines

Our Risk-MCTS framework uses **llama-3-7b-instruct** (lla, 2024) and **glm-4-8b-chat** (GLM et al., 2024) as base models for the value model, policy model, and table operation model, implementing zero-shot learning (Kojima et al., 2023) for task-specific adaptation. For comparison, we evaluate against traditional machine learning approaches including **NeuralNet**, **RandomForest**, and **XGBoost** models trained using **Autogluon**, maintaining a 24-month gap between training and test periods following (Dyck et al., 2007). We also compare against larger language models including **llama-3.3-70b-instruct** (lla, 2024), **qwen2-72b** (Yang et al., 2024), DeepSeek-R1 (DeepSeek-AI et al., 2025) and Grok-3 which have significantly larger parameter counts (>70B) than our base models. These are evaluated both with and without Chain of Thought (CoT) prompting (Wang et al., 2024b). Additionally, we include **tablegpt2-7b** (Su et al., 2024), a model specifically fine-tuned for tabular data processing, to evaluate our framework against domain-specialized approaches.

4.3 Evaluation Method

We evaluate model performance through both automatic metrics and human evaluation. For automatic evaluation, we focus on metrics suitable for imbalanced datasets: AUC, F1-score, and balanced accuracy. Due to the extreme class imbalance in AAER (0.66% positive samples) and CSMAR (2.11% positive samples), we note that Recall may not be fully representative, particularly for AAER. For human evaluation, we randomly sample 100 predictions from each model and have 3 financial experts rate the quality of their intermediate reasoning steps on a 0-5 scale, focusing on logical coherence and financial soundness.

4.4 Main Results

4.4.1 Automatic Evaluation

Our experimental results (Table 2 and 3) demonstrate how Risk-MCTS addresses the key challenges in financial risk detection:

First, Risk-MCTS significantly outperforms larger language models, achieving a 31.9% improvement in AUC and 9.0% improvement in F1-score on CSMAR compared to qwen2-72b-cot. This improvement, despite using fewer parameters (7B vs 70B+).

Second, traditional machine learning methods show strong performance on numerical features but lack interpretability. While Random Forest achieves competitive AUC (0.731 on both datasets), Risk-MCTS provides comparable or better performance while generating explanatory reasoning chains.

Third, Chain of Thought (CoT) prompting improves baseline LLM performance, but Risk-MCTS shows superior gains (15.8% in AUC and 21.3% in F1-score on AAER). The value model's probability-based scoring mechanism proves particularly effective in guiding the search process, as shown by the consistent performance across both datasets despite their different distributions.

4.4.2 Human Evaluation

The human evaluation results are shown in the **Intermediate** column in Table 2 and Table 3. Risk-MCTS achieved significantly higher expert ratings (3.48 on CSMAR, 3.31 on AAER) compared to the best baseline models (2.73 and 2.63 respectively), representing improvements of 32.3% and 44.4%. This substantial gap in interpretability scores demonstrates that our framework's structured reasoning approach produces more coherent and financially sound analysis chains.

4.5 Ablation study

We evaluate the table operation model by comparing three variants: (1) the full model, (2) basic operations only (removing financial operations), and

	AUC	Recall	F1-score	balanced accuracy	Intermediate
LLM					
Llama3.3-70b	0.526	0.714	0.325	0.526	-
Qwen2-72b	0.543	0.643	0.496	0.554	-
DeepSeek-R1	0.619	0.357	0.680	0.526	-
Grok-3	0.527	0.381	0.703	0.549	-
TableGPT2-7b	0.367	0.524	0.312	0.413	-
COT					
Llama3-70b-cot	0.541	0.262	0.785	0.535	2.53
Qwen2-72b-cot	0.565	0.714	0.810	0.557	2.54
DeepSeek-R1-cot	0.593	0.286	0.786	0.546	2.63
Grok-3-cot	0.552	0.119	0.841	0.496	2.41
TableGPT2-7b-cot	0.469	0.238	0.728	0.493	2.19
Machine Learning					
NeuralNet	0.659	0.725	0.856	0.504	-
RandomForest	0.731	0.730	0.863	0.502	-
XGBoost	0.736	0.718	0.738	0.499	-
Ours					
Risk-MCTS	**0.745**	**0.857**	**0.883**	**0.666**	**3.48**
$\uparrow$	20.4%	20.0%	9.0%	19.6%	32.3%

Table 2: CSMAR evaluation results. **Intermediate** refers to the human evaluation score of intermediate steps. The $\uparrow$ refers to the improvement compared to the LLM-based approach.

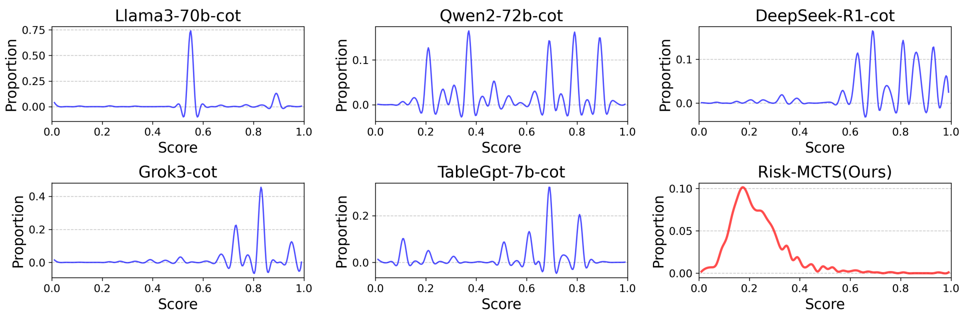

Figure 3: Score distribution of different models

Datasets	Total	Positive	%	Attributes
AAER	146,044	964	0.66	42
CSMAR	225,000	4659	2.11	248

Table 3: Information of two datasets,% represents the proportion of positive examples in the dataset

	AUC	F1-Score
Ours	0.814	0.893
Ours w/o financial OPs	0.758	0.499
Ours w/o full OPs	0.715	0.327

Table 4: Results of ablation study

(3) no structured operations. As shown in Table 4, removing financial operations leads to a significant drop in F1-score (0.893 $\rightarrow$ 0.499), while removing all structured operations further degrades performance (F1-score: 0.327). This degradation pattern demonstrates that both general table operations and domain-specific financial operations are crucial for effective risk detection.

4.6 Score calculation formulas of value model

We evaluate the value model by prompting the value model to directly output the score instead of using the next token logits as score. Additionally, we modify the quality value calculation formula (Equation 15) to:

$$w = \frac{S_{true}}{S_{true} + S_{false}} \tag{20}$$

Results in Table 5 show that the direct score method performs significantly worse than the next token logits method (AUC: 0.642 vs 0.814), validating our probability-based scoring mechanism. The modified score calculation formula has minor impact on performance (AUC: 0.810), suggesting the robustness of our token probability approach.

4.7 Sensitivity Analysis

In order to verify the effectiveness of the MCTS framework, we evaluate the performance of the model under branch 1, 2 and 4, then compare their performance. Results in Figure 5 show that building a tree instead of a chain significantly improves performance (AUC: 0.546 vs 0.814). However,

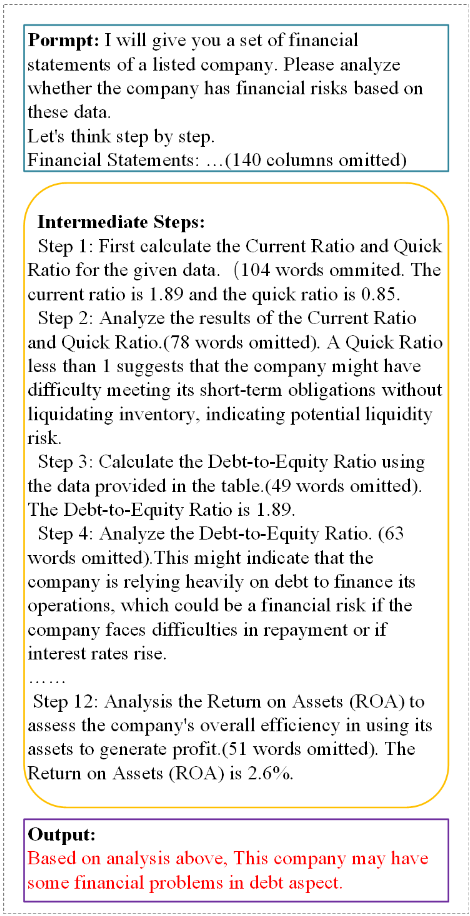

Figure 4: an example of Risk-MCTS expands the solution space for correct answers.

increasing the number of branches does not significantly improve performance, and the complexity of the method increases significantly.

4.8 Score distribution of Value Models

In Figure 3 we compare the distribution of the final risk score evaluated by the value model in Risk-MCTS with the final risk score distribution of all COT methods. We observe that scores of COT methods are very unevenly distributed, reaching a very high proportion in a certain score range (Llama3-70b-cot>60%, Grok3-cot>40%, TableGPT2-7B-cot>30%), which indicates that these three models are likely to perform poorly in risk prediction tasks. However, the score distribution of the Risk-MCTS method is more even and has a higher proportion of low scores, which matches the imbalanced distribution of the dataset and explains why the Risk-MCTS method performs well in risk prediction tasks.

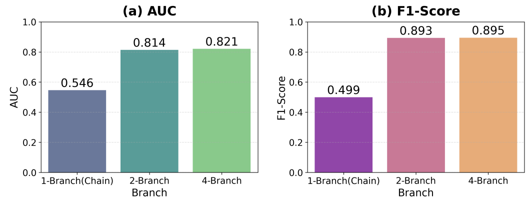

Figure 5: AUC and F1-Score in different branch settings

	AUC	F1-Score
Original score	0.814	0.893
Direct score	0.642	0.261
modified formula	0.810	0.876

Table 5: Performance of Risk-MCTS under different score calculation formulas

4.9 Case Study

To demonstrate the interpretability of Risk-MCTS, we present a detailed example of reasoning trajectory in Figure 4. The framework systematically analyzes financial risk through multiple steps:

First, Risk-MCTS calculates key financial ratios from the raw financial statements, focusing on indicators that could signal potential risks. Then, it systematically analyzes each ratio, providing detailed reasoning for why certain patterns might indicate financial irregularities. Finally, it synthesizes these individual analyses into a comprehensive risk assessment, supported by the evidence gathered through each step.

This example illustrates how Risk-MCTS provides transparent and verifiable decision-making, contrasting with the black-box nature of traditional approaches.

5 Conclusions

In this work, we propose a novel framework called Risk-MCTS to solve the financial risk detection problem based on financial statements. This framework is composed of LLM-based policy model, value model and table operation model. In the iterative process of building a Monte Carlo search tree, the policy model provides expansion policy, the value model evaluates every node, and table operation model is designed to modify tables in every node and enhance the understanding of tabular data. In our experiments, we compare Risk-MCTS with base LLMs, LLMs with COT, LLM specialized for tabular data and machine learning methods. Experiments show that we achieve SOTA in both accuracy and explainability.

Limitations

Our research is a significant step forward in applying large language models to diverse downstream tasks. However, there are still some challenges. The main constraint is that our framework requires LLM to generate text content iteratively, which costs a large number of tokens and time, and We have not yet found a more efficient way to build a MCTS search tree. In addition, since our framework requires LLM to generate multiple different answers for the same input prompt, we need to set a higher temperature for LLM, which will lead to unstable generated results. Further research may need to focus on these limitations and seek for a better solution.

References

2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Yang Bao, Bin Ke, Bin Li, Yingri Yu, and Jie Zhang. 2019. Detecting accounting fraud in publicly traded u.s. firms using a machine learning approach. *Journal of Accounting Research*, 58.

Mark Cecchini, Haldun Aytug, Gary J. Koehler, and Praveen Pathak. 2010. Detecting management fraud in public companies. *Manag. Sci.*, 56:1146–1160.

Guillaume Chaslot, Mark Winands, H. Herik, Jos Uiterwijk, and Bruno Bouzy. 2008. Progressive strategies for monte-carlo tree search. *New Mathematics and Natural Computation*, 04:343–357.

Yi Cheng, Renjun Hu, Haochao Ying, Xing Shi, Jian Wu, and Wei Lin. 2024. Arithmetic feature interaction is necessary for deep tabular learning. *Preprint*, arXiv:2402.02334.

Patricia Craja, Alisa Kim, and Stefan Lessmann. 2020. Deep learning for detecting financial statement fraud. *Decision Support Systems*, 139:113421.

Patricia Dechow, Weili Ge, Chad Larson, and Richard Sloan. 2010. Predicting material accounting misstatements. *Contemporary Accounting Research*, 28.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, and Xiaokang Zhang. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

Yang Deng, Wenqiang Lei, Wenxuan Zhang, Wai Lam, and Tat-Seng Chua. 2023. Pacific: Towards proactive conversational question answering over tabular and textual data in finance. *Preprint*, arXiv:2210.08817.

Alexander Dyck, Adair Morse, and Luigi Zingales. 2007. Who blows the whistle on corporate fraud? *The Journal of Finance*, 65.

Alexander Dyck, Adair Morse, and Luigi Zingales. 2023. How pervasive is corporate fraud? *Review of Accounting Studies*, 29.

Team GLM, :, Aohan Zeng, Bin Xu, Bowen Wang, Chenhui Zhang, Da Yin, Dan Zhang, Diego Rojas, Guanyu Feng, and Hanlin Zhao. 2024. Chatglm: A family of large language models from glm-130b to glm-4 all tools. *Preprint*, arXiv:2406.12793.

Zhen Huang, Haoyang Zou, Xuefeng Li, Yixiu Liu, Yuxiang Zheng, Ethan Chern, Shijie Xia, Yiwei Qin, Weizhe Yuan, and Pengfei Liu. 2024. O1 Replication Journey – Part 2: Surpassing O1-preview through Simple Distillation, Big Progress or Bitter Lesson? *arXiv preprint*.

Deyi Ji, Lanyun Zhu, Siqi Gao, Peng Xu, Hongtao Lu, Jieping Ye, and Feng Zhao. 2024. Tree-of-Table: Unleashing the Power of LLMs for Enhanced Large-Scale Table Understanding. *arXiv preprint*.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2023. Large language models are zero-shot reasoners. *Preprint*, arXiv:2205.11916.

Li-Cheng Lan, Wei Li, Ting-Han Wei, and I-Chen Wu. 2019. Multiple policy value monte carlo tree search. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 4704–4710. International Joint Conferences on Artificial Intelligence Organization.

Qian Luo, Tien Ping Tan, Daochen Zha, and Tianqiao Zhang. 2024. Enhanced doudizhu card game strategy using oracle guiding and adaptive deep monte carlo method. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI-24*.

Yiwei Qin, Xuefeng Li, Haoyang Zou, Yixiu Liu, Shijie Xia, Zhen Huang, Yixin Ye, Weizhe Yuan, Hector Liu, Yuanzhi Li, and Pengfei Liu. 2024. O1 Replication Journey: A Strategic Progress Report – Part 1. *arXiv preprint*.

Wojciech Samek and Klaus-Robert Müller. 2019. *Towards Explainable Artificial Intelligence*, page 5–22. Springer International Publishing.

Aofeng Su, Aowen Wang, Chao Ye, Chen Zhou, Ga Zhang, Gang Chen, Guangcheng Zhu, Haobo Wang, Haokai Xu, Hao Chen, and Haoze Li. 2024. Tablegpt2: A large multimodal model with tabular data integration. *Preprint*, arXiv:2411.02059.

Liyang Wang, Yu Cheng, Ao Xiang, Jingyu Zhang, and Haowei Yang. 2024a. Application of natural language processing in financial risk detection. *Preprint*, arXiv:2406.09765.

Zilong Wang, Hao Zhang, Chun-Liang Li, Julian Martin Eisenschlos, Vincent Perot, Zifeng Wang, Lesly Miculicich, Yasuhisa Fujii, Jingbo Shang, Chen-Yu Lee, and Tomas Pfister. 2024b. Chain-of-Table: Evolving Tables in the Reasoning Chain for Table Understanding. *arXiv preprint*.

Katarzyna Woźnica, Piotr Wilczyński, and Przemysław Biecek. 2023. Sefnet: Bridging tabular datasets with semantic feature nets. *Preprint*, arXiv:2306.11636.

Wu Xiuguo and Du Shengyong. 2022. An Analysis on Financial Statement Fraud Detection for Chinese Listed Companies Using Deep Learning. *IEEE Access*, 10:22516–22532.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, and Jian Yang. 2024. Qwen2 technical report. *Preprint*, arXiv:2407.10671.

Xinze Yang, Chunkai Zhang, Yizhi Sun, Kairui Pang, Luru Jing, Shiyun Wa, and Chunli Lv. 2023. Finchain-bert: A high-accuracy automatic fraud detection model based on nlp methods for financial scenarios. *Information*, 14:499.

Dan Zhang, Sining Zhoubian, Ziniu Hu, Yisong Yue, Yuxiao Dong, and Jie Tang. 2024a. ReST-MCTS*: LLM Self-Training via Process Reward Guided Tree Search. *arXiv preprint*.

Di Zhang, Xiaoshui Huang, Dongzhan Zhou, Yuqiang Li, and Wanli Ouyang. 2024b. Accessing gpt-4 level mathematical olympiad solutions via monte carlo tree self-refine with llama-3 8b. *Preprint*, arXiv:2406.07394.

Di Zhang, Jianbo Wu, Jingdi Lei, Tong Che, Jiatong Li, Tong Xie, Xiaoshui Huang, Shufei Zhang, Marco Pavone, Yuqiang Li, Wanli Ouyang, and Dongzhan Zhou. 2024c. Llama-berry: Pairwise optimization for o1-like olympiad-level mathematical reasoning. *Preprint*, arXiv:2410.02884.

Yu Zhao, Huifeng Yin, Bo Zeng, Hao Wang, Tianqi Shi, Chenyang Lyu, Longyue Wang, Weihua Luo, and Kaifu Zhang. 2024. Marco-o1: Towards open reasoning models for open-ended solutions. *Preprint*, arXiv:2411.14405.

Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. 2015. Object detectors emerge in deep scene cnns. *Preprint*, arXiv:1412.6856.