Are Foundation Models Useful for Bankruptcy Prediction?

Marcin Kostrzewa[1], **Oleksii Furman**[1], **Roman Furman**[3],
Sebastian Tomczak[1], **Maciej Zięba**[1,2]
[1]Wrocław University of Science and Technology, [2]Tooploox, [3]Opera
E-mail: marcin.kostrzewa@pwr.edu.pl

Abstract

Foundation models have shown promise across various financial applications, yet their effectiveness for corporate bankruptcy prediction remains systematically unevaluated against established methods. We study bankruptcy forecasting using Llama-3.3-70B-Instruct and TabPFN, evaluated on large, highly imbalanced datasets of over one million company records from the Visegrád Group. We provide the first systematic comparison of foundation models against classical machine learning baselines for this task. Our results show that models such as XGBoost and CatBoost consistently outperform foundation models across all prediction horizons. LLM-based approaches suffer from unreliable probability estimates, undermining their use in risk-sensitive financial settings. TabPFN shows mixed results — underperforming on ROC-AUC but competitive on F_1-score — while requiring substantial computational resources that cannot be justified by its performance. These findings suggest that, despite their generality, current foundation models remain less effective than specialized methods for bankruptcy forecasting.

1 Introduction

Foundation models such as large language models (LLMs) are increasingly applied across various financial domains, as documented in recent survey papers [12, 17, 25]. Applications span sentiment analysis, algorithmic trading, fraud detection, credit rating, and other financial tasks. Their appeal lies mainly in the ability to provide solutions without extensive task-specific engineering.

One of the most extensively studied financial prediction problems is corporate bankruptcy[1] forecasting [7, 9, 11, 21, 27]. Accurate default prediction is of practical relevance to investors, regulators, and policymakers, while the task's structured nature and requirement for reliable probability estimates present unique challenges for general-purpose models. While foundation models have been applied to various financial risk tasks, systematic evaluation against classical methods for bankruptcy prediction, particularly on large-scale structured datasets, remains unexplored.

In this work, we address this gap by applying two foundation models — Llama-3.3-70B-Instruct [1] for textual inputs and TabPFN [8] for tabular data — to corporate default forecasting. We evaluate them on highly imbalanced datasets of financial records from the Visegrád Group, containing hundreds of thousands of samples. While recent work has addressed TabPFN's scalability limitations [14, 23], these solutions remain untested for financial prediction tasks against domain-optimized classical methods. We extensively compare these approaches against classical machine learning baselines and analyze the reliability of foundation model outputs for structured financial prediction.

Our contributions can be summarized as follows:

[1]Throughout the paper we will use terms *default*, *financial distress*, and *bankruptcy* interchangeably.

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: Generative AI in Finance.

- We provide the first systematic comparison of foundation models (LLMs and TabPFN) against classical ML baselines for bankruptcy prediction on large-scale structured financial data.
- We show that LLM self-reported probability estimates are poorly calibrated and discretized, undermining their reliability for risk assessment, and that TabPFN incurs substantial computational overhead without commensurate performance gains.

2 Related Work

Foundation models in finance. Recent work has demonstrated foundation model capabilities across financial domains, from sentiment analysis and time series forecasting [5, 13, 17] to risk prediction. FinPT applies LLMs to personal financial risk prediction [24], while time series foundation models like FinCast and Kronos show strong performance for market forecasting [19, 26]. For bankruptcy prediction specifically, multimodal approaches have been explored using textual and numerical data from US markets [3, 16], but systematic evaluation against classical methods on structured data remains limited.

Bankruptcy prediction methods. Traditional machine learning approaches dominate bankruptcy prediction, with gradient boosting methods (XGBoost, CatBoost) and neural networks consistently achieving strong performance on financial datasets [4, 10, 27]. Recent studies have incorporated more advanced deep learning techniques [9, 22] and ensemble methods [15]. Despite foundation model advances in other financial areas, no systematic comparison exists evaluating their effectiveness against these established approaches for bankruptcy prediction on large-scale datasets.

3 Data and methodology

3.1 Data

We construct five datasets based on over one million financial statement records from 2006-2021 of companies from the V4 group (Czech Republic, Hungary, Poland, Slovakia). The data is a part of our ongoing research and will be described in detail in a separate publication.

Each dataset corresponds to a different prediction horizon: immediate bankruptcy risk (0 years ahead) up to four years ahead. The target variable is binary, indicating whether a company will experience financial distress as defined by a set of financial ratio thresholds. This multi-horizon approach enables systematic evaluation of model performance as prediction difficulty increases with longer time periods. Table 1 summarizes the five datasets, showing the expected decline in both sample size and bankruptcy cases as prediction horizons extend. The severe class imbalance (bankruptcy rates below 1%) reflects real-world conditions and necessitates careful evaluation using metrics appropriate for imbalanced classification.

Detailed feature descriptions and target variable construction are provided in Appendix B.

Table 1: Overview of datasets and prediction tasks.

Prediction Horizon	Total Instances	Bankruptcy	Non-bankruptcy
0 years (current)	1,000,087	3,587	996,500
1 year ahead	996,500	3,054	993,446
2 years ahead	898,692	2,374	896,318
3 years ahead	793,234	1,896	791,338
4 years ahead	700,041	1,485	698,556

Evaluation subsets. For each prediction horizon, all models are evaluated on the same stratified test subset of 20,000 samples to ensure computational feasibility and fair comparison across methods. The remaining data is split into training and validation sets, with validation data used for hyperparameter optimization and probability threshold calibration.

3.2 Methods

To evaluate foundation models against classical approaches for bankruptcy prediction, we employ both a LLM and a tabular foundation model alongside established baselines. For each dataset, we report ROC-AUC and F_1-score, with the latter being particularly important for highly imbalanced data. The decision threshold is calibrated on the validation dataset by choosing the value maximizing F_1-score. Detailed results for additional metrics are provided in Appendix D.

Foundation models We evaluate two representative foundation models, each addressing different data modalities. For textual input processing, we use Llama-3.3-70B-Instruct [1], serializing company features into natural-language prompts that request binary labels and self-reported probabilities. We test both zero-shot prediction and in-context learning (ICL) with 20 examples (10 positive, 10 negative). We access Llama-3.3 via API, effectively treating it as a closed model since we cannot examine weights or internal representations.

For direct tabular processing, we utilize TabPFN [8], a transformer-based model designed for small tabular datasets. Since TabPFN is optimized for datasets under 10,000 samples while our datasets contain hundreds of thousands of records, we implement a *partition-then-predict* approach where a decision tree partitions the feature space and TabPFN processes smaller leaf subsets. Detailed prompt design and scaling approaches are described in Appendices C.1 and C.2.

Classical baselines. We compare foundation models against five established methods: logistic regression, multi-layer perceptron, XGBoost, LightGBM, and CatBoost. We follow a standard training protocol with hyperparameter optimization via grid search on validation data. Hyperparameters are selected to maximize F_1-score, prioritizing performance on the minority class. Complete hyperparameter grids and training specifications are provided in Appendix C.3.

4 Experiments and results

The results indicate that classical methods consistently outperform foundation models across all prediction horizons. Table 2 shows that traditional ML methods maintain stable performance across all time horizons, with ROC-AUC scores declining gradually from $0.99+$ at $h = 0$ to 0.85–0.89 at $h = 4$.

Foundation models show weaker performance. TabPFN achieves 0.987 ROC-AUC at $h = 0$, declining to 0.771 at $h = 4$, underperforming all standard ML methods across most horizons. While TabPFN outperforms MLP and LR on F_1-scores at most horizons, it still trails behind gradient boosting. LLM-based approaches exhibit the weakest performance in terms of ROC-AUC. The F_1-scores reveal even larger gaps on this imbalanced dataset, with strong classical methods achieving 0.024–0.069 at $h = 4$ compared to foundation models' 0.012–0.024.

Beyond overall performance gaps, LLMs exhibit specific reliability issues that undermine their practical utility. The self-reported probability estimates are poorly calibrated and discretized, as shown in Figure 1. The model outputs cluster around fixed values (0.1, 0.2, 0.7, 0.9) rather than providing smooth probability distributions, making them unsuitable for risk assessment applications.

Table 2: Performance across prediction horizons (ROC-AUC / F_1-score).

Model	$h = 0$	$h = 1$	$h = 2$	$h = 3$	$h = 4$
XGBoost	$\mathbf{0.996}/\mathbf{0.465}$	$\mathbf{0.968}/0.200$	$\mathbf{0.894}/0.198$	$0.896/0.055$	$\mathbf{0.891}/0.024$
CatBoost	$\mathbf{0.996}/0.431$	$0.965/\mathbf{0.238}$	$0.886/0.161$	$\mathbf{0.901}/0.067$	$0.883/0.062$
LightGBM	$\mathbf{0.996}/0.459$	$0.965/0.229$	$0.878/\mathbf{0.180}$	$0.888/0.060$	$0.877/\mathbf{0.069}$
MLP	$0.994/0.404$	$0.959/0.157$	$0.848/0.059$	$0.895/\mathbf{0.071}$	$0.877/0.021$
LR	$0.983/0.167$	$0.952/0.148$	$0.853/0.064$	$0.858/0.045$	$0.850/0.012$
TabPFN	$0.987/0.400$	$0.951/0.196$	$0.800/0.131$	$0.823/0.063$	$0.771/0.024$
Llama-3.3	$0.945/0.141$	$0.914/0.091$	$0.796/0.020$	$0.823/0.010$	$0.782/0.012$
Llama-3.3 (ICL)	$0.966/0.114$	$0.932/0.064$	$0.817/0.022$	$0.807/0.019$	$0.780/0.015$

Foundation models require substantial computational resources without commensurate performance gains. Our timing analysis (Table 3) demonstrates significant computational overhead for foundation models, which require specialized hardware and substantially longer processing times compared to classical methods.

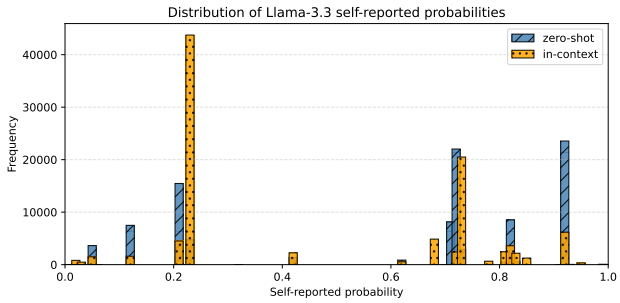

Figure 1: Distribution of self-reported classification output probabilities for Llama-3.3 in both zero-shot and in-context learning variants across all datasets.

Table 3: Inference timing. Each entry is the mean±std over 5 runs; we also report throughput (samples/s).

Model	**Hardware**	**Batch**	**Time (s)** $\mu \pm \sigma$	**Throughput** (samples/s)
XGBoost	CPU (M4 Pro 4 threads)	20,000	0.007 ± 0.001	$2,833,570.17$
TabPFN	GPU (NVIDIA A100-40GB)	20,000	23.782 ± 1.668	844.77
Llama	API (8 concurrent requests)	20,000	5357.400 ± 205.563	3.739

5 Limitations

Our study's limitations primarily concern the LLM approach. We use Llama-3.3-70B-Instruct, which is not among currently available top models. Advanced reasoning models (GPT-5-thinking [18], Claude Opus 4.1 [2], DeepSeek-V3.1 [6], Qwen3 [20]) may deliver stronger performance. Additionally, API-level access restricts us to returned probability estimates; direct access to weights and logits could yield more reliable probabilities.

6 Conclusions

In this study, we conducted the first evaluation of foundation models for corporate bankruptcy prediction. Using a large-scale dataset of Central European companies, we compared Llama-3.3 and TabPFN with established machine learning methods.

Our findings indicate that classical machine learning approaches maintain advantages over general-purpose foundation models in structured financial prediction. The performance gaps we observed, spanning accuracy, computational efficiency, and probability calibration, were not marginal but substantial, suggesting that foundation models are not an adequate choice for such task.

Most critically, we identified that LLM probability estimates exhibit a degenerate distribution, clustering around discrete values rather than providing the reliable confidence measures essential for financial decision-making. The computational overhead of foundation models without corresponding performance benefits further undermines their business case. While TabPFN demonstrates competitive F_1-scores on this imbalanced dataset, outperforming simpler baselines at most horizons, it underperforms on ROC-AUC and requires specialized hardware that cannot be justified when gradient boosting methods consistently deliver superior results on standard hardware.

These findings suggest that current foundation models are not yet a viable replacement for specialized machine learning approaches in bankruptcy forecasting. Nonetheless, our work highlights important challenges and points toward directions for improvement. Future research should explore LLMs with reasoning capabilities, models with accessible weights and logits, and hybrid multimodal approaches that combine textual and numerical financial data.

References

[1] Meta AI. Llama 3.3 model card. https://github.com/meta-llama/llama3/blob/main/MODEL_CARD.md, 2024.

[2] Anthropic. Claude 4.1 system card. https://assets.anthropic.com/m/4c024b86c698d3d4/original/Claude-4-1-System-Card.pdf, 2025.

[3] Henri Arno, Klaas Mulier, Joke Baeck, and Thomas Demeester. From numbers to words: Multi-modal bankruptcy prediction using the ecl dataset. *arXiv preprint arXiv:2401.12652*, 2024.

[4] Sami Ben Jabeur, Nicolae Stef, and Pedro Carmona. Bankruptcy prediction using the xgboost algorithm and variable importance feature engineering. *Computational Economics*, 61(2): 715–741, 2023.

[5] Liyuan Chen, Shuoling Liu, Jiangpeng Yan, Xiaoyu Wang, Henglin Liu, Chuang Li, Kecheng Jiao, Jixuan Ying, Yang Veronica Liu, Qiang Yang, and Xiu Li. Advancing financial engineering with foundation models: Progress, applications, and challenges, 2025. URL https://arxiv.org/abs/2507.18577.

[6] DeepSeek-AI. Deepseek-v3 technical report, 2024. URL https://arxiv.org/abs/2412.19437.

[7] Jairaj Gupta, Andros Gregoriou, and Tahera Ebrahimi. Empirical comparison of hazard models in predicting smes failure. *Quantitative Finance*, 18(3):437–466, 2018.

[8] Noah Hollmann, Samuel Müller, Lennart Purucker, Arjun Krishnakumar, Max Körfer, Shi Bin Hoo, Robin Tibor Schirrmeister, and Frank Hutter. Accurate predictions on small data with a tabular foundation model. *Nature*, 637(8045):319–326, January 2025. ISSN 1476-4687. doi: 10.1038/s41586-024-08328-6. URL http://dx.doi.org/10.1038/s41586-024-08328-6.

[9] Tadaaki Hosaka. Bankruptcy prediction using imaged financial ratios and convolutional neural networks. *Expert systems with applications*, 117:287–299, 2019.

[10] Sami Ben Jabeur, Cheima Gharib, Salma Mefteh-Wali, and Wissal Ben Arfi. Catboost model and artificial intelligence techniques for corporate failure prediction. *Technological Forecasting and Social Change*, 166:120658, 2021.

[11] Yi Jiang and Stewart Jones. Corporate distress prediction in china: A machine learning approach. *Accounting & Finance*, 58(4):1063–1109, 2018.

[12] David Kuo Chuen Lee, Chong Guan, Yinghui Yu, and Qinxu Ding. A comprehensive review of generative ai in finance. *FinTech*, 2024. URL https://api.semanticscholar.org/CorpusID:272766486.

[13] Jean Lee, Nicholas Stevens, Soyeon Caren Han, and Minseok Song. A survey of large language models in finance (finllms). *ArXiv*, abs/2402.02315, 2024. URL https://api.semanticscholar.org/CorpusID:267412025.

[14] Si-Yang Liu and Han-Jia Ye. Tabpfn unleashed: A scalable and effective solution to tabular classification problems. *ArXiv*, abs/2502.02527, 2025. URL https://api.semanticscholar.org/CorpusID:276107889.

[15] Wei Liu, Yoshihisa Suzuki, and Shuyi Du. Ensemble learning algorithms based on easyensemble sampling for financial distress prediction. *Annals of Operations Research*, pages 1–32, 2025.

[16] Rogelio A Mancisidor and Kjersti Aas. Multimodal generative models for bankruptcy prediction using textual data. *arXiv preprint arXiv:2211.08405*, 2022.

[17] Yuqi Nie, Yaxuan Kong, Xiaowen Dong, John M. Mulvey, H. Vincent Poor, Qingsong Wen, and Stefan Zohren. A survey of large language models for financial applications: Progress, prospects and challenges. *ArXiv*, abs/2406.11903, 2024. URL https://api.semanticscholar.org/CorpusID:270562262.

[18] OpenAI. Gpt-5 system card. https://cdn.openai.com/gpt-5-system-card.pdf, 2025.

[19] Yu Shi, Zongliang Fu, Shuo Chen, Bohan Zhao, Wei Xu, Changshui Zhang, and Jian Li. Kronos: A foundation model for the language of financial markets, 2025. URL https://arxiv.org/abs/2508.02739.

[20] Qwen Team. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

[21] Shanshan Wang and Guotai Chi. Cost-sensitive stacking ensemble learning for company financial distress prediction. *Expert Systems with Applications*, 255:124525, 2024.

[22] Chenyang Wu, Cuiqing Jiang, Zhao Wang, and Yong Ding. Predicting financial distress using current reports: A novel deep learning method based on user-response-guided attention. *Decision Support Systems*, 179:114176, 2024.

[23] Han-Jia Ye, Si-Yang Liu, and Wei-Lun Chao. A closer look at tabpfn v2: Understanding its strengths and extending its capabilities. 2025. URL https://api.semanticscholar.org/CorpusID:279306362.

[24] Yuwei Yin, Yazheng Yang, Jian Yang, and Qi Liu. Finpt: Financial risk prediction with profile tuning on pretrained foundation models. Papers 2308.00065, arXiv.org, Jul 2023. URL https://ideas.repec.org/p/arx/papers/2308.00065.html.

[25] Huaqin Zhao, Zhengliang Liu, Zihao Wu, Yiwei Li, Tianze Yang, Peng Shu, Shaochen Xu, Haixing Dai, Lin Zhao, Hanqi Jiang, Yi Pan, Junhao Chen, Yifan Zhou, Gengchen Mai, Ninghao Liu, and Tianming Liu. Revolutionizing finance with llms: An overview of applications and insights, 2024. URL https://arxiv.org/abs/2401.11641.

[26] Zhuohang Zhu, Haodong Chen, Qiang Qu, and Vera Chung. Fincast: A foundation model for financial time-series forecasting, 2025. URL https://arxiv.org/abs/2508.19609.

[27] Yao Zou, Changchun Gao, and Han Gao. Business failure prediction based on a cost-sensitive extreme gradient boosting machine. *IEEE Access*, 10:42623–42639, 2022.

A Acknowledgements

The research in this paper has been partially supported by the National Science Centre (NCN, Poland), under Grant no. 2020/39/D/HS4/02384 and under Grant no. 2024/55/B/ST6/02100.

We would like to also gratefully acknowledge the Polish high-performance computing infrastructure PLGrid (HPC Center: ACK Cyfronet AGH) for providing computer facilities and support within computational grant no. PLG/2025/018494.

We would also like to thank CLARIN-PL for granting access to their infrastructure and API services.

B Dataset details

B.1 Data Source and Coverage

Our evaluation utilizes a comprehensive financial dataset sourced from the Emerging Markets Information Service (EMIS) database, covering companies across four Central European countries: Poland, Hungary, Slovakia, and the Czech Republic. The dataset spans the period from 2006 to 2021, providing a robust foundation for financial distress prediction analysis.

The dataset comprises 203,900 unique companies with a total of 1,106,879 company-year observations distributed as follows: Poland (628,499 entries), Hungary (358,486 entries), Slovakia (62,141 entries), and Czech Republic (57,753 entries). Each record represents a company's financial data for a specific year.

B.2 Financial Distress Definition and Classification Task

To construct the target variable for our predictive models, we defined financial default based on a set of financial indicators from a company's last available annual report. A company was labeled as defaulted if it simultaneously met three criteria in its final reporting year: a negative equity to total assets (*equity/total_assets* < 0), negative EBITDA relative to total assets (*EBITDA/total_assets* < 0), and a current ratio below 0.6 (*current_assets/current_liabilities* < 0.6). Companies whose last available report was for the year 2021 were excluded from being labeled as distressed as we collected the data up to 2021 and we lacked the required information. For a task of predicting default 4 years in advance, we took companies that met the default criteria, removed their data for the final 4 years, and then assigned a positive label to the new final year of data for each of these firms. All other company-year observations in the datasets were assigned a negative label. This process resulted in a dataset tailored for predicting financial default at a horizon of 4 years before the observed default. In a similar manner, we construct datasets with prediction horizons $h = 1, 2, 3$.

B.3 Feature Set and Financial Indicators

The dataset contains 131 features encompassing various categories of financial indicators. A detailed features list with descriptions is presented in Table 4.

Table 4: The complete set of features considered in the classification process.

Feature ID	Description
Company Identifiers & Metadata	
X1	Country
X2	Has multiple industries flag (binary)
X3	Incorporation date 1
X4	Incorporation date 2
X5	Legal form (LLC, Corp, etc.)
X6	NAICS 2-digit classification
X7	NAICS 3-digit classification
X8	Number of employees
X9	Operational status (Active, Inactive, etc.)
X10	Primary NAICS (encoded)
X11	Secondary NAICS (encoded)
X12	Sector 1
X13	State/region
X14	Report year
Liquidity and Profitability Ratios	
X15	Cash / sales
X16	Cash / total assets
X17	Cash / total operating revenue
X18	(Current assets - inventories - receivables) / short term liabilities
X19	(Current assets - inventories) / short term liabilities
X20	Current assets / sales
X21	Current assets / short term liabilities
X22	EBIT / equity
X23	EBIT / financial costs
X24	EBIT / total assets
X25	EBIT / total costs
X26	EBIT / total liabilities
X27	EBIT / total operating revenue
X28	EBITDA / fixed assets
X29	EBITDA / total assets
X30	EBITDA / total operating revenue
X31	(Gross profit + depreciation) / total liabilities
X32	Gross profit / short term liabilities

Feature ID	Description
X33	Gross profit / total assets
X34	Gross profit / total operating revenue
X35	Interest expense / revenue
X36	Inventories / working capital
X37	(Net profit + depreciation) / current liabilities
X38	(Net profit + depreciation) / total liabilities
X39	Net profit / equity
X40	Net profit / fixed assets
X41	Net profit / inventories
X42	Net profit / total assets
X43	Net profit / total operating revenue
X44	Net profit / current assets
X45	Operational expenses / short term liabilities
X46	Operational expenses / total liabilities
X47	Quick assets / sales
X48	Retained profit / short term liabilities
X49	Retained profit / total assets
X50	Working capital (absolute value)
X51	Working capital / equity
X52	Working capital / fixed assets
X53	Working capital / sales
X54	Working capital / total assets
X55	Working capital / total liabilities
X56	Working capital / total operating revenue
X57	Cash flow / sales
X58	Cash flow / total debt
X59	Loss flag (Net Profit (Loss) for the Period < 0)
Turnover and Cycle Ratios	
X60	Cash conversion cycle (days)
X61	Inventories / total operating revenue
X62	Operating cycle (days)
X63	Operating expenses / sales
X64	Receivables turnover days
X65	Revenue / current assets
X66	Revenue / long term liabilities
X67	Revenue / total liabilities
X68	Short term liabilities turnover days
X69	Total operating revenue / fixed assets
X70	Total operating revenue / inventories
X71	Total operating revenue / receivables
X72	Total operating revenue / short term liabilities
X73	Total operating revenue / total assets
Solvency and Capital Structure Ratios	
X74	Constant capital / fixed assets
X75	Constant capital / total assets
X76	Current assets / total liabilities
X77	Current assets / total operating revenue
X78	Current liabilities / total liabilities
X79	Current liabilities / current assets
X80	Current liabilities / equity
X81	Short term liabilities / total assets
X82	(Equity - share capital) / fixed assets
X83	Equity / fixed assets
X84	Equity / long term liabilities
X85	Equity / sales
X86	Equity / total assets

Feature ID	Description
X87	Equity / total liabilities
X88	Equity ratio classification
X89	Fixed assets / long term liabilities
X90	Fixed assets / total assets
X91	(Inventories + receivables) / equity
X92	Inventory / current liabilities
X93	Long term liabilities / current assets
X94	Long term liabilities / equity
X95	(Total liabilities - cash) / EBITDA
X96	(Total liabilities - cash) / total operating revenue
X97	Total liabilities / total assets
X98	Insolvency flag (Total liabilities $>$ Total assets)
Growth Ratios (Year-over-Year)	
X99	Current assets growth (YoY)
X100	Inventories growth (YoY)
X101	Net profit growth (YoY)
X102	Operating profit growth (YoY)
X103	Operating revenue growth (YoY)
X104	Receivables growth (YoY)
X105	Short term liabilities growth (YoY)
X106	Total assets growth (YoY)
Additional and Derived Indicators (Size and Macro)	
X107	Logarithm of current assets
X108	Logarithm of (net profit / GDP)
X109	Logarithm of (operating profit / GDP)
X110	Logarithm of (revenue / GDP)
X111	Logarithm of total liabilities
X112	Logarithm of total assets
X113	Logarithm of (total assets / GDP)
X114	Logarithm of total operating revenue
Sector-Relative Indicators *Represents the difference between the company's ratio and the industry sector average for that year.*	
X115	Cash conversion cycle (sector-relative)
X116	(Current liabilities $\times 365$) / revenue (sector-relative)
X117	Current assets / current liabilities (sector-relative)
X118	EBITDA margin (sector-relative)
X119	(Inventories $\times 365$) / revenue (sector-relative)
X120	Net profit / absolute equity (sector-relative)
X121	Net profit / assets (sector-relative)
X122	Net profit / current assets (sector-relative)
X123	Net profit / fixed assets (sector-relative)
X124	Net profit / sales (sector-relative)
X125	Operating cycle (sector-relative)
X126	(Receivables $\times 365$) / revenue (sector-relative)
X127	Revenue / assets (sector-relative)
X128	Revenue / fixed assets (sector-relative)
X129	Short-term financial assets / current liabilities (sector-relative)
X130	Short-term receivables investments / current liabilities (sector-relative)
X131	Working capital / assets (sector-relative)

C Model training and evaluation details

Data preprocessing. For each dataset, we impute missing numeric values with training dataset medians and standardize using training dataset statistics (scaling is not performed for data fetched by LLM). Categorical columns such as *Country*, *State/Region*, and *Legal form* are label-encoded.

C.1 Llama-3.3

Prompt design. In our prompt template, we provide guidance for bankruptcy prediction and impose a financial analyst persona on LLM. The template includes detailed explanations of financial ratio categories, risk interpretation guidelines, and strict output formatting requirements. When using in-context learning, an `**Examples**` section with 20 examples is inserted before the target company details.

The complete prompt structure is shown below.

LLM Prompt Template for Bankruptcy Prediction

```
You are a financial analyst specializing in bankruptcy prediction for
companies.  Analyze the provided financial data and predict [TIMEFRAME]
bankruptcy risk.

**Task**:  Based on the financial metrics below, predict if this company will
go bankrupt within the next [X] years.

**Prediction Timeframe**:  This is a [X]-year ahead prediction.
[TIMEFRAME-SPECIFIC GUIDANCE]

**Key Financial Ratios Explained**:
- **Liquidity Ratios**:  Measure ability to pay short-term debts
- Current_assets/ current_liabilities:  Current ratio (>1.0 = good liquidity)
- Current_assets-inventories/ current_liabilities:  Acid-test ratio (removes
less liquid inventories)
- Current_assets-inventories-receivables/ current_liabilities:  Quick ratio
(most conservative liquidity measure)
- Working_capital:  Current assets minus current liabilities (positive = good)
- Working_capital/total_assets:  Working capital efficiency (higher = better
liquidity management)
- Cash/total_assets:  Cash position relative to total assets (higher = safer)
- **Profitability Ratios**:  Measure earnings performance
- Net_profit/total_assets:  Return on assets (ROA, higher = better)
- EBIT/total_assets:  Operating return on assets (excludes financial structure
effects)
- EBITDA/total_assets:  Cash-based return on assets (excludes depreciation)
- Net_profit/total_operating_revenue:  Net profit margin (higher = better)
- EBIT/total_operating_revenue:  Operating margin (higher = better)
- EBITDA/total_operating_revenue:  EBITDA margin (cash flow efficiency)
- EBIT/equity:  Return on equity (higher = better)
- EBIT/total_costs:  Operating efficiency ratio
- Net_profit/fixed_assets:  Asset utilization efficiency
- EBITDA/fixed_assets:  Cash generation from fixed assets
- **Leverage Ratios**:  Measure debt levels and financial risk
- EBIT/total_liabilities:  Earnings coverage of total debt (higher = better)
- Net_profit+depreciation/total_liabilities:  Cash flow coverage of debt
- Total_liabilities/total_assets:  Debt-to-asset ratio (lower = better)
- Equity/total_assets:  Equity ratio (higher = better)
- Equity/fixed_assets:  Equity financing of fixed assets (higher = better)
- Long_term_liabilities/equity:  Long-term debt burden
- Current_liabilities/current_assets:  Short-term debt pressure (lower =
better)
- **Efficiency Ratios**:  Measure operational performance
- Operating_cycle:  Days to convert inventory to cash (lower = better)
- Cash_conversion_cycle:  Net days to convert investments to cash
```

- Receivables_turnover_days: Days to collect receivables (lower = better)
- **Growth Metrics**: Measure company expansion
- Operating_revenue_growth: Revenue growth rate
- Total_assets_growth: Asset growth rate
- Net_profit_growth: Profit growth rate
- **Asset Composition**: Measure asset structure and efficiency
- Fixed_assets/total_assets: Asset structure (higher = more capital intensive)
- Working_capital/fixed_assets: Working capital relative to fixed investment
- Current_assets/total_liabilities: Asset coverage of liabilities
- **Risk Flags**: Binary indicators
- Insolvency_flag: 1 if company is technically insolvent
- Loss_flag: 1 if company has consecutive losses

Bankruptcy Risk Indicators:
- **High Risk**: Negative working capital, low liquidity ratios (<1.0), high debt ratios (>0.7), declining profits, insolvency/loss flags
- **Medium Risk**: Declining growth, moderate debt levels (0.4-0.7), industry volatility, operational issues
- **Low Risk**: Strong liquidity (>1.5), positive growth, low debt (<0.4), consistent profitability, stable industry

Ratio Interpretation Guidelines:
- Current ratio < 1.0: Severe liquidity problems
- Quick ratio < 1.0: Potential liquidity problems
- Working capital/total_assets < 0: Negative working capital (high risk)
- Debt-to-asset ratio > 0.7: High financial risk
- EBIT/total_liabilities < 0.1: Poor debt coverage
- Fixed_assets/total_assets > 0.8: High capital intensity (industry dependent)
- Negative growth rates: Declining business performance
- Operating cycle > 120 days: Inefficient operations
- Loss flags = 1: Immediate bankruptcy risk

Company Structure Features:
- **Industry Codes**: NAICS classification system
- primary_naics_encoded: Main industry code (higher numbers = more specific industries)
- naics_2digit: Broad sector (11-99, e.g., 23=Construction, 31-33=Manufacturing)
- naics_3digit: Subsector (e.g., 236=Construction of Buildings)
- has_multiple_industries: 1 if company operates in multiple industries (higher risk)
- secondary_naics_encoded: Secondary industry if diversified
- **Incorporation Date**:
- incorporation_date_1: 0-2y, 3-4y, 5-24y, >24y
- incorporation_date_2: 0-1y, 1-2y, 3-5y, 6-9y, 10-19y, >19y
- **Operational Status**: Liquidation, Under Legal Investigation, Closed, Active

Response Format: Respond with exactly two numbers separated by a comma: [prediction],[probability]

Where: - prediction: 1 if the company will likely go bankrupt, 0 if not - probability: a decimal between 0.0 and 1.0 representing the probability of bankruptcy

Example responses:
- "1,0.85" (high bankruptcy risk)
- "0,0.15" (low bankruptcy risk)
- "1,0.65" (moderate-high bankruptcy risk)

RETURN ONLY THE REQUIRED NUMBERS, NO OTHER TEXT

Examples:

Example 1: Company Info: country=Hungary, state=Bacs-Kiskun, number_of_employees=1-9 employees, legal_form=Limited Liability

```
Partnership, primary_naics_encoded=42512, naics_2digit=42, naics_3digit=425,
has_multiple_industries=Single Industry, secondary_naics_encoded=0,
sector_1=Wholesale Trade, year=2,019.00
Liquidity:  Working_capital=-25.170, Cash/total_assets=0.000,
Inventories/working_capital=0.000...
[Additional financial metrics...]
Risk Flags:  Insolvency_flag=1.000, Loss_flag=0.000 Prediction:  0

Example 2:  Company Info:  country=Poland, state=Podlaskie,
number_of_employees=10-49 employees, legal_form=Limited Liability Company...
[Financial metrics...]
Prediction:  0

...

**Now analyze this company**:
Company Info:  country=Poland, state=Lodzkie, number_of_employees=10-49
employees, legal_form=Limited Liability Partnership, primary_naics_encoded=236.0,
naics_2digit=23.0, naics_3digit=236.0...
[Complete set of financial indicators...]
Prediction:
```

In-context learning examples selection. Our procedure for selecting $2k$ examples for in-context learning looks as follows:

- we train a proxy XGBoost model on the training data and score the validation dataset to obtain probabilities,
- we choose k samples from bankruptcy class with lowest probabilities and k samples from non-bankruptcy class with highest probabilities.

In other words, we try to choose *hard* samples in terms of prediction for in-context learning. For all tasks, we choose $k = 10$.

Calibrating threshold Threshold for calculating metrics is selected by maximizing F_1 score on a dataset of 5,000 samples sub-sampled with stratification from the validation dataset.

C.2 TabPFN approaches

We implement two strategies to handle our large-scale datasets with TabPFN, which is optimized for datasets under 10,000 samples.

Decision Tree Partitioning (TabPFN-DT) We partition the entire training dataset using shallow decision tree, setting the minimum number of samples required to split an internal node to 10,000. The decision tree partitions the training set into smaller, more manageable subsets. During inference, a test instance is first passed through the decision tree to a leaf node and then predicted by the corresponding TabPFN model.

Bootstrap Ensemble (TabPFN-Ensemble) We also implement a bootstrap ensemble approach where we iteratively sample m subsets, each containing $n < N$ randomly selected samples. For each subset, we leverage TabPFN's ability to handle the reduced dataset size to obtain predictions. This divide-and-conquer strategy aggregates outputs using majority voting for classification. The approach's performance is determined by the number of datasets bootstrapped.

Performance comparison Table 5 compares the performance of both TabPFN approaches for the 4-year horizon prediction. Based on the obtained results, we use the TabPFN-DT approach in the main text analysis.

Threshold calibration The threshold for calculating metrics on the test dataset is selected to maximize the F_1-score on the validation dataset.

Table 5: TabPFN approaches performance comparison for $h = 4$ (ROC-AUC / F_1-score).

Model	ROC-AUC	F_1-score
TabPFN-DT	0.771	**0.024**
TabPFN-Ensemble-8	0.797	0.018
TabPFN-Ensemble-16	0.794	0.013
TabPFN-Ensemble-24	**0.814**	0.013

C.3 Classical methods hypertuning

Evaluation protocol. For each prediction horizon, metrics are computed on a common stratified test *subset* of 20,000 samples (computational parity across models). We do not use class weights or resampling; class imbalance is handled via F_1-score-based model selection and threshold calibration.

Models and search spaces. Hyperparameters are selected using grid search by maximizing the F_1-score on the validation dataset. After selection, a decision threshold is calibrated on the validation dataset to maximize the F_1-score; this threshold is stored and used at test time when evaluating metrics.

The complete configuration of the search grid is given in Table 6.

Table 6: Hyperparameter ranges used in grid search.

Model	**Search space**
LR	$C \in$ logspace$(0.001, 1, 5)$; penalty = L2
MLP	hidden_layer_sizes $\in \{(32,32), (64,64), (128,128), (256,256), (512,512)\}$; $\alpha \in \{10^{-4}, 10^{-3}, 10^{-2}\}$; learning_rate_init $\in \{10^{-3}, 10^{-2}\}$
XGBoost	n_estimators $\in \{100, 200, 500\}$; max_depth $\in \{3, 5, 7\}$; learning_rate $\in \{0.01, 0.05, 0.1, 0.2\}$; eval_metric = logloss
LightGBM	n_estimators $\in \{100, 200\}$; max_depth $\in \{-1, 5, 10\}$; learning_rate $\in \{0.01, 0.05, 0.1, 0.2\}$
CatBoost	iterations $\in \{100, 200\}$; depth $\in \{4, 6, 8\}$; learning_rate $\in \{0.01, 0.05, 0.1\}$; silent=True

Software. Implementations rely on `scikit-learn`, `xgboost`, `lightgbm`, and `catboost`. All runs use `random_state` $= 42$.

D Complete results

Table 7 gathers complete results for all metrics (accuracy, precision, recall, F_1-score, ROC-AUC) evaluated for all models for each specific dataset.

Table 7: Complete performance results across all prediction horizons and metrics.

Prediction Horizon	Model	Accuracy	Precision	Recall	F_1-score	ROC-AUC
$h = 0$	XGBoost	0.995	0.368	0.630	**0.465**	**0.996**
	CatBoost	0.994	0.336	0.603	0.431	**0.996**
	LightGBM	**0.995**	**0.382**	0.575	0.459	**0.996**
	MLP	0.993	0.300	0.616	0.404	0.994
	LR	0.990	0.123	0.260	0.167	0.983
	TabPFN	**0.995**	0.336	0.493	0.400	0.987
	Llama-3.3	0.973	0.080	0.616	0.141	0.945
	Llama-3.3 (ICL)	0.964	0.063	**0.644**	0.114	0.966
$h = 1$	XGBoost	0.993	0.150	0.298	0.200	**0.968**
	CatBoost	0.993	0.172	**0.386**	**0.238**	0.964
	LightGBM	**0.994**	**0.174**	0.333	0.229	0.965
	MLP	0.992	0.116	0.246	0.157	0.959
	LR	0.993	0.114	0.210	0.148	0.952
	TabPFN	**0.994**	0.156	0.263	0.196	0.951
	Llama-3.3	0.991	0.064	0.158	0.091	0.914
	Llama-3.3 (ICL)	**0.994**	0.059	0.070	0.064	0.932
$h = 2$	XGBoost	0.993	**0.145**	0.309	**0.198**	**0.894**
	CatBoost	**0.995**	**0.145**	0.182	0.161	0.886
	LightGBM	0.993	0.134	0.273	0.180	0.878
	MLP	0.992	0.044	0.091	0.059	0.848
	LR	0.993	0.049	0.091	0.064	0.853
	TabPFN	0.992	0.094	0.218	0.131	0.800
	Llama-3.3	0.804	0.010	**0.727**	0.020	0.796
	Llama-3.3 (ICL)	0.991	0.016	0.036	0.022	0.817
$h = 3$	XGBoost	0.995	0.046	0.067	0.054	0.896
	CatBoost	0.990	0.043	0.156	0.067	**0.901**
	LightGBM	0.988	0.036	0.178	0.060	0.888
	MLP	**0.996**	**0.075**	0.067	**0.071**	0.895
	LR	0.992	0.030	0.089	0.046	0.858
	TabPFN	0.993	0.044	0.111	0.063	0.823
	Llama-3.3	0.602	0.005	**0.889**	0.010	0.823
	Llama-3.3 (ICL)	0.938	0.010	0.267	0.019	0.807
$h = 4$	XGBoost	**0.996**	0.025	0.022	0.024	**0.891**
	CatBoost	0.988	0.038	0.178	0.062	0.883
	LightGBM	0.990	**0.044**	0.156	**0.069**	0.877
	MLP	0.991	0.014	0.044	0.021	0.877
	LR	0.992	0.008	0.022	0.012	0.850
	TabPFN	0.992	0.016	0.044	0.024	0.771
	Llama-3.3	0.783	0.006	**0.600**	0.012	0.782
	Llama-3.3 (ICL)	0.948	0.008	0.178	0.015	0.780