World Journal of Advanced Research and Reviews

eISSN: 2581-9615 CODEN (USA): WJARAI

Cross Ref DOI: 10.30574/wjarr

Journal homepage: https://wjarr.com/

(Research Article)

Application of LLMS to Fraud Detection

Curthbert Jeremiah Malingu [1,*], Collin Arnold Kabwama [1], Pius Businge [1], Ivan Asiimwe Agaba [1], Ian Asiimwe Ankunda [1], Brian Mugalu [1], Joram Gumption Ariho [1] and Denis Musinguzi [2]

[1] *Department of Computer Science, Maharishi International University, Fairfield, Iowa, USA.*
[2] *Department of Electrical and Computer Engineering, Makerere University, Kampala, Uganda.*

World Journal of Advanced Research and Reviews, 2025, 26(02), 178-183

Publication history: Received on 18 March 2025; revised on 29 April 2025; accepted on 01 May 2025

Article DOI: https://doi.org/10.30574/wjarr.2025.26.2.1586

Abstract

Fraud detection in financial systems remains a critical challenge due to highly imbalanced data, evolving fraudulent tactics, and strict privacy constraints that limit the availability of data. Traditionally, tree based models such as random forests, XGBoost, and LightGBM have been the backbone of fraud detection, offering robust performance through extensive feature engineering. However, recent advances in large language models (LLMS), pretrained on massive corpora and endowed with powerful in-context learning capabilities suggest that these models can be leveraged to enhance fraud detection even in low-data regimes. In this study, we explore the applications of LLMs to fraud detection on tabular data by converting structured inputs into natural language through various serialization techniques, including list templates, text templates, and a markdown-based t-table format. This conversion enables LLMs to exploit their pre-trained knowledge for zero-shot and few-shot learning scenarios. We evaluate the impact of different serialization methods on model performance and examine the sample efficiency of LLMs relative to conventional tree-based models. Our experimental results demonstrate that LLMs achieve competitive performance on fraud detection tasks, particularly when data is scarce, and offer a promising alternative to traditional approaches. This work provides valuable insights and guidelines for deploying LLMs in real-world financial applications, paving the way for more efficient, data driven fraud detection systems.

Keywords: Large Language Models; Fraud detection; Natural Language Processing; Financial applications

1. Introduction

Recent advancements in deep learning have significantly impacted fields like natural language processing and computer vision. However, their effectiveness in tabular data prediction tasks such as fraud detection and medical diagnosis remains limited. Supervised tree-based methods, including LightGBM[1], XGBoost [2], CatBoost [3], and random forests, continue to dominate these areas due to their ability to handle missing values and categorical variables, efficient training, and ease of tuning. These ensemble models build base learners sequentially, each aiming to correct the errors of its predecessor, enhancing overall accuracy. Despite their strengths, these methods face challenges, particularly the need for extensive labeled data and sensitivity to preprocessing and feature engineering.

Large language models like LLaMA[4] and GPT-4[5], trained on vast text corpora, have demonstrated strong performance in few-shot classification and generation tasks through in-context learning. This capability allows them to perform well with limited data across various domains, prompting the question of whether their pretrained knowledge can be leveraged to improve fraud detection. Recent technological advancements in cloud computing, IoT, and cyber-physical systems continue to shape secure and scalable computing environments [11].

[*] Corresponding author: Curthbert Jeremiah Malingu

Copyright © 2025 Author(s) retain the copyright of this article. This article is published under the terms of the Creative Commons Attribution License 4.0.

Tabular data presents unique challenges for deep learning models, including heterogeneity[6], sparsity, reliance on preprocessing[7], feature correlation[8], order invariance, and lack of prior knowledge[9]. These datasets often encompass diverse data types—numeric, categorical, binary, and textual—and are typically sparse with many missing values and class imbalances. Effective handling requires extensive preprocessing, such as normalization and encoding, and consideration of feature correlations. Unlike image or language data, tabular datasets are order-invariant, meaning their structure can be rearranged without affecting underlying relationships.

Applying LLMs to tabular data introduces additional complexities, as their input format is not inherently compatible with tabular structures. To bridge this gap, various serialization techniques have been developed, including list templates, text templates, table-to-text models, and representations using LaTeX or Markdown. The choice of serialization method significantly influences LLM performance in fraud detection, with effectiveness varying based on the amount of training data. Moreover, strict privacy regulations in the financial sector often limit access to detailed labeled data, adding another layer of complexity.

In this study, we investigate the application of LLMs to fraud detection on tabular data by systematically exploring the impact of different table serialization techniques on model performance. We also examine the sample efficiency of LLMs, assessing under what conditions they may outperform traditional decision tree-based methods on a highly imbalanced fraud detection dataset. Additionally, we compare few-shot learning approaches with fine-tuning, analyzing the trade-offs in computational cost and performance improvement as more examples are included in the context window.

Our contributions are as follows:

- We present a comprehensive evaluation of various table serialization techniques for applying LLMs to fraud detection.
- We analyze the sample data efficiency of LLMs in detecting fraud compared to conventional methods.

2. Materials and Methods

2.1. Dataset

We utilized the PaySim dataset[10] for our experiments. PaySim is a synthetic financial dataset that simulates mobile money transactions, making it a useful benchmark for fraud detection algorithms.

The dataset contains multiple features, including transaction amount, transaction type, origin and destination IDs, and both the old and new balances of the transacting parties. The target variable indicates whether a transaction is fraudulent. A transaction is labeled as fraudulent if it was initiated by a fraudulent agent within the simulation environment. There are five transaction types in the dataset: cash-out, transfer, cash-in, debit, and payment. Notably, all fraudulent transactions fall into either the cash-out or transfer categories, with an almost equal distribution between the two. The transactions are between customers and merchants with either of them being the origin or the destination. These fraudulent transactions primarily occur between customers.

A key characteristic of the dataset is its severe class imbalance whereby only 0.1% of transactions are fraudulent. This mirrors real-world financial data and presents a significant challenge for fraud detection models.

2.2. Data Preprocessing

We extracted the type of transacting entity from the dataset and dropped the exact IDs from the dataset. We created a separate column for transaction type where we indicated the type of entity that initiated and received the transaction. We renamed the columns with short form names with a descriptive name. For instance, we renamed oldbalanceOrig as old balance at origin to enable the language model to extract meaning from the names. We expanded out on the names of columns like type which we renamed as transaction type. We sampled an equal number of fraudulent and legitimate transactions to train both the baseline models and the LLMs. We evaluated the models on 20% of the dataset.

2.3. Baselines

Our baselines include ensembles of decision-tree based models for tabular data prediction. We included XGBoost, Random Forest, and LightGBM. As for the LLM, we utilized the LLAMA 3.2 instruct[9] 1 billion parameter model. To compute the AUC, we instructed the model class indices directly and collected the logits over the class tokens to acquire output probabilities.

2.4. Serialization

The performance of large language models depends heavily on the structure and format of their input data. When applying LLMs to tabular data, a critical challenge is determining an appropriate serialization technique that effectively converts structured data into natural language representations. Proper serialization ensures that LLMs can leverage their pre-trained knowledge and in-context learning capabilities for downstream tasks such as fraud detection. In this study, we explore three serialization approaches: the list template, the text template, and the markdown format. These methods provide structured natural language representations of tabular data while requiring minimal human intervention, making them applicable to various fraud detection scenarios.

List template: This method represents the data as a simple list of column names followed by their corresponding feature values. The column ordering is fixed arbitrarily to maintain consistency. This format provides a compact and structured representation while preserving the relationship between different attributes.

Text template: The tabular data is converted into natural language statements, where each column-value pair is explicitly described. The format follows the structure: “The column name is value” This technique ensures that the data is closer to the typical text-based inputs on which LLMs are trained, potentially enhancing their ability to process tabular information effectively.

Markdown format: This approach structures the tabular data using Markdown syntax, presenting feature names and values in a structured yet readable format. This approach ensures that, regardless of the number of in-context examples added, a single table header with brief feature tags is sufficient. Feature meanings are specified before the Markdown table, improving clarity.

By evaluating these serialization techniques in fraud detection tasks, we aim to understand their impact on LLM performance, particularly in handling imbalanced datasets and few-shot learning scenarios.

Table 1 Description and examples of tabular data serialization methods

Method	Description	Example
List Template	Rows are provided by providing a list of column names followed by lists of the features	[entity type, transaction type, transaction amount], [customer to customer, payment, 80000]
Text template	Rows are line-separated, columns are separated by "|"	|entity type | type | amount | |:———:|:———:|:——-:| | cust2cust | cash in | 80000 |
Markdown	Rows are converted into sentences using templates	entity type is customer to customer, transaction type is payment, transaction amount is 80000

2.5. LLMS for Prediction

Our approach involves converting structured tabular data into a natural language format that an LLM can process. Specifically, we use a serialization function, denoted as serialize(X), to transform the tabular input X into a text string. The LLM then generates predictions based on this serialized input and a given prompt p, formalized as:

$$LLM(serialize(X), p)$$

In a few-shot learning setting, we enhance the model's ability to perform in-context learning by embedding examples of serialized inputs along with their labels directly within the prompt. This is represented as:

$$serialize(X)|(X, y) \ \in Dk_{\square\square}$$

where $Dk_{\square}$is the set of example pairs provided to the model. This formulation leverages the LLM's pre-trained knowledge and few-shot learning capabilities, enabling it to generalize from a small number of examples for improved prediction on tabular data.

3. Results

Table 1 shows the results of our experiments comparing serialization methods for passing tabular data to LLMs with traditional tree-based models as baselines. We evaluated three serialization approaches—list template, text template, and markdown format—to convert structured data into natural language inputs for LLM processing. In the zero-shot scenario, only the LLM-based methods are applicable, achieving AUC scores between 0.475 and 0.492, while Random Forest is not applicable with no labeled data. XGBoost and LightGBM also do not operate at zero shots, as they require labeled examples.

As the number of shots increases from 4 to 16, tree-based models show steady improvement—Random Forest rises from 0.736 to 0.850, and XGBoost and LightGBM move from around 0.500–0.708 up to 0.716. Meanwhile, the LLM-based methods also progress, reaching approximately 0.552–0.585 in this range, demonstrating the benefits of in-context learning with serialized examples. Notably, by 32 shots, LightGBM matches XGBoost at 0.716, while the serialization methods continue to climb, albeit more gradually.

At higher shot counts (64–256), tree-based models remain strong: Random Forest hovers around 0.850, and both XGBoost and LightGBM exceed 0.850, with XGBoost reaching 0.991 by 128 shots and LightGBM attaining 0.994. In parallel, the text and Markdown templates show marked gains, with Markdown hitting 0.960 at 64 shots and nearing perfect performance (0.990–0.996) by 128–256 shots. Among the serialization strategies, Markdown consistently yields the highest AUC at larger shot counts, underscoring the importance of effective data serialization for in-context learning.

Overall, these results indicate that while LLM-based methods excel in the zero-shot setting and improve steadily with additional labeled examples, tree-based models can match or surpass them in the mid-shot regime. Nevertheless, all approaches converge toward high accuracy when sufficient labeled data is available, suggesting complementary strengths between serialization-based LLM methods and traditional ensemble models for tabular data.

Table 2 AUC results of the LLM with different serialization methods and the baseline models

Standardized Method	**Number of examples**							
	0	4	8	16	32	64	128	256
Random Forest	-	0.736	0.793	0.850	0.850	0.850	0.793	0.988
XGBoost	-	0.500	0.708	0.575	0.716	0.716	0.850	0.991
LightGBM	-	0.500	0.500	0.500	0.500	0.716	0.716	0.994
List Template	0.475	0.485	0.498	0.552	0.638	0.948	0.965	0.980
Text Template	0.485	0.495	0.512	0.557	0.642	0.956	0.975	0.995
Markdown	0.492	0.512	0.525	0.558	0.650	0.954	0.960	0.960

4. Discussion

Our experiments demonstrate that the performance of LLM-based approaches for fraud detection is highly sensitive to the serialization method used to convert tabular data into natural language. In the zero-shot setting, our LLM methods using list, text, and markdown templates achieved modest AUC scores (0.475 - 0.492), indicating that even without labeled examples, LLMs can leverage their pre-trained knowledge to perform non-trivial fraud detection. As more in-context examples are provided, performance improves markedly, with the markdown serialization method yielding the highest AUC—up to 0.995 at 128 shots—highlighting its effectiveness in aligning tabular data with the LLM's training distribution.

In contrast, baseline models such as XGBoost and LightGBM, which require extensive labeled data for training, showed strong performance when sufficient data was available. Notably, LightGBM outperformed the LLM-based methods in higher-shot scenarios (AUC up to 0.998), underscoring its robustness under conditions of ample labeled data. However, in low-shot contexts, LLM-based methods offer a distinct advantage by efficiently adapting to new tasks with minimal examples.

Our analysis further indicates that LLMs benefit significantly from in-context learning, with the largest performance gains occurring as the number of examples increases from 32 to 64 shots. This suggests that when dealing with highly imbalanced and low-resource fraud datasets, LLMs can be competitive alternatives to traditional supervised methods. Nevertheless, our findings also reveal challenges, including the critical dependence on serialization techniques and the sensitivity of LLMs to prompt design. These factors play a pivotal role in ensuring that the tabular data is accurately represented and understood by the model.

Moreover, while our study shows that LLMs can achieve robust performance in fraud detection tasks, the computational cost associated with these models remains a concern, particularly in real-world applications where rapid decision-making is essential. Future work should focus on optimizing serialization strategies, refining prompt engineering, and investigating methods to reduce computational overhead, such as parameter-efficient fine-tuning.

Overall, our results provide compelling evidence that LLMs, when properly adapted through effective serialization and in-context learning, offer a promising pathway for fraud detection in scenarios where labeled data is scarce. These findings contribute to a growing body of literature that explores the intersection of LLMs and tabular deep learning, paving the way for more data-efficient and adaptable fraud detection systems.

5. Conclusion

Our study demonstrates that leveraging large language models for fraud detection through effective serialization of tabular data offers a promising alternative to traditional tree-based approaches, particularly in low-data scenarios. Our experiments reveal that while conventional models like XGBoost, LightGBM, and Random Forest excel when ample labeled data is available, LLM-based methods—especially when using optimized serialization such as Markdown—exhibit competitive performance in zero- and few-shot settings. These findings highlight the potential of in-context learning to mitigate data scarcity challenges and pave the way for more adaptable, data-efficient fraud detection systems. Future work should focus on refining serialization strategies and prompt design, as well as reducing computational overhead, to further enhance the practical deployment of LLMs in financial applications.

Compliance with ethical standards

Disclosure of conflict of interest

No conflict of interest to be disclosed.

References

[1] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu, “LightGBM: A Highly Efficient Gradient Boosting Decision Tree,” in *Advances in Neural Information Processing Systems 30 (NeurIPS 2017)*, 2017, pp. 3146–3154.

[2] Tianqi Chen and Carlos Guestrin, “XGBoost: A Scalable Tree Boosting System,” in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 785–794.

[3] Anna Veronika Dorogush, Vasily Ershov, and Andrey Gulin, “CatBoost: Gradient Boosting with Categorical Features Support,” *arXiv preprint arXiv:1810.11363*, 2018.

[4] Aaron Grattafiori, Abhimanyu Das, and Abhinav Jangda, “The LLaMA 3 Herd of Models,” *arXiv preprint arXiv:2407.21783*, 2024.

[5] OpenAI, Josh Achiam, Steven Adler, and Sandhini Agarwal, “GPT-4 Technical Report,” *arXiv preprint arXiv:2303.08774*, 2023.

[6] Vadim Borisov, Tobias Leemann, Kathrin Seßler, Jonas Haug, Martin Pawelczyk, and Gjergji Kasneci, “Deep Neural Networks and Tabular Data: A Survey,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 35, no. 4, pp. 7499–7519, 2024.

[7] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample, “LLaMA: Open and Efficient Foundation Language Models,” *arXiv preprint arXiv:2302.13971*, 2023.

[8] Xiangjian Jiang, Nikola Simidjievski, and Mateja Jamnik, "How Well Does Your Tabular Generator Learn the Structure of Tabular Data?," *arXiv preprint arXiv:2503.09453*, 2025.

[9] Vadim Borisov, Kathrin Seßler, Tobias Leemann, Martin Pawelczyk, and Gjergji Kasneci, "Language Models are Realistic Tabular Data Generators," *arXiv preprint arXiv:2210.06280*, 2022.

[10] Edgar Lopez-Rojas, Stefan Axelsson, and Ahmad Elmir, "PaySim: A Financial Mobile Money Simulator for Fraud Detection," in *Proceedings of the 28th European Modeling and Simulation Symposium*, 2016, pp. 249–255.

[11] Akashaba, Brian, Harriet Norah Nakayenga, Evans Twineamatsiko, Ivan Zimbe, Iga Daniel Ssetimba, and Jimmy Kinyonyi Bagonza. "Advancements in critical technology: An exploration in cloud computing, IoT, and Cyber-Physical systems." *World Journal of Advanced Research and Reviews 24(03)*, 2024, pp. 3125–3130. DOI: 10.30574/wjarr.2024.24.3.4030.