Adversarially Resilient Financial Risk Forecasting with Split-Trained Transformer Language Models

Francesco Karlsson
School of Electrical Engineering and Computer Science, Oregon State University, Corvallis, OR, USA.
hellofrancesco@oregonstate.edu

Abstract

The integration of transformer-based language models into financial risk forecasting has enabled unprecedented accuracy in capturing semantic and temporal dependencies within heterogeneous data streams. However, the deployment of such models in real-world financial infrastructures introduces critical vulnerabilities to adversarial manipulation, particularly when training is distributed across multiple custodians to comply with privacy and regulatory constraints. This paper proposes and analyzes a system architecture for adversarially resilient financial risk forecasting that leverages split-trained transformer language models. Split training partitions the model across different parties, thereby preserving data locality while enabling collaborative learning. The central contribution is the formalization of an adversarial threat model tailored to the split-training paradigm, encompassing poisoning, evasion, and backdoor attacks that target both the feature extractor and the classification head. We synthesize recent advances in defense mechanisms, including gradient sanitization, prototype consistency verification, and robust aggregation protocols, and evaluate their effectiveness under realistic financial data distributions. Through a multi-layered analysis of architectural trade-offs, we examine how communication overhead, model fidelity, and privacy guarantees interact with adversarial resilience. The paper further discusses the broader governance, fairness, and sustainability implications of deploying such systems in critical financial infrastructure, emphasizing the need for regulatory frameworks that address adversarial robustness as a core design requirement. Our findings indicate that while no single defense is sufficient, a combination of prototype-based validation and differential privacy offers a promising path toward trustworthy financial AI. The proposed framework provides a foundation for future empirical validation and policy development in the intersection of machine learning security and financial regulation.

Keywords

adversarial resilience; transformer language models; split learning; financial risk forecasting; backdoor defense; prototype consistency; governance.

1. Introduction

Financial risk forecasting has traditionally relied on econometric models and structured data, but the advent of large-scale language models has transformed the field by enabling the extraction of signals from unstructured textual sources such as earnings reports, central bank communications, and social media sentiment [1]. Transformer architectures, with their capacity to model long-range dependencies through self-attention mechanisms, have become the backbone of these advanced forecasting systems [2]. Yet the same flexibility that makes transformers effective also renders them susceptible to adversarial perturbations. Malicious actors can craft inputs that cause catastrophic mispredictions, potentially destabilizing

markets or enabling arbitrage exploitation [3]. The problem is compounded when the training process itself is distributed across multiple institutions, as is common in financial consortia where data cannot be centralized due to privacy regulations and competitive sensitivities.

Split learning offers a compelling solution to the data-sharing dilemma by partitioning the model into a frontend that resides with each data owner and a backend that is aggregated by a central server [4]. In the context of transformer language models, the embedding layers and early transformer blocks are typically placed on the client side, while the final classification layers are trained on the server. This arrangement ensures that raw textual data never leaves the client's custody, thereby satisfying many regulatory requirements. However, it also introduces new attack surfaces: an adversary controlling a client can inject poisoned embeddings or trigger backdoors that persist in the shared representation space [5]. Recent work has demonstrated that backdoor attacks can be successfully mounted in vertical split learning settings, and defenses such as prototype consistency verification have been proposed to mitigate these threats [14].

This paper provides a comprehensive system-level analysis of adversarially resilient financial risk forecasting using split-trained transformer language models. We examine the architectural design choices, the threat landscape, and the interplay between privacy, communication efficiency, and robustness. The discussion is situated within the broader context of financial infrastructure, where the stakes of model failure are exceptionally high. We also address the governance and sustainability dimensions, arguing that adversarial resilience must be incorporated into the lifecycle of financial machine learning systems from development through deployment.

2. Background and Related Work

Financial risk forecasting has evolved from linear factor models to deep learning approaches that exploit nonlinear relationships and high-dimensional data [1]. Machine learning models, particularly those based on recurrent and convolutional architectures, have been applied to credit scoring, portfolio optimization, and market volatility prediction. The introduction of transformer language models, such as FinBERT, has further improved the ability to incorporate textual information alongside numerical time series [7]. These models are typically trained on large corpora of financial documents and fine-tuned on specific forecasting tasks.

Adversarial machine learning has been extensively studied in the context of image classification, where small perturbations to input pixels can cause misclassification [3]. In the financial domain, adversarial attacks can target both the input data, such as manipulated news articles, and the training pipeline itself. Backdoor attacks are particularly insidious because they embed a trigger into the model during training; the model behaves normally on clean inputs but produces attacker-chosen outputs when the trigger is present [5]. Such attacks have been demonstrated in federated and split learning environments, where the distributed nature of training makes detection more challenging.

Split learning was originally proposed to preserve privacy in healthcare settings, but its applicability to finance is natural given the stringent data protection regulations [4]. In vertical split learning, different parties hold different features of the same set of samples, and the model is split such that the forward pass is computed jointly. For transformer language models, the embedding and attention layers can be executed locally, while the final feedforward layers are computed on the server [11]. This architecture reduces the bandwidth

requirement compared to full gradient sharing but still requires the communication of intermediate representations.

Defenses against adversarial attacks in distributed learning fall into several categories. Gradient compression and perturbation techniques can obscure the malicious signal, but they may also degrade model accuracy [12]. Differential privacy provides formal guarantees against inference attacks but typically increases noise and reduces utility [9]. More recent methods focus on verifying the consistency of learned representations across clients. For instance, prototype-based defenses compare the embeddings of incoming samples against a set of benign prototypes to detect anomalies [14]. These approaches are particularly relevant to split learning because the server has access to the aggregated representations and can perform statistical tests.

3. System Architecture and Split Training Paradigm

The proposed system architecture consists of multiple client institutions, each holding proprietary textual and numerical data, and a coordinating server that aggregates information for risk forecasting. Each client runs a frontend model comprising the embedding layer and the initial transformer blocks. The backend model, typically a multilayer perceptron or a lightweight transformer, resides on the server and outputs the risk prediction. During training, the client computes the forward pass up to the cut layer, sends the intermediate representations to the server, and receives gradients for backpropagation through the split. This process is repeated for each mini-batch, with the server updating its own parameters and sending updates to the clients [4].

The split training paradigm introduces several structural trade-offs. On one hand, it preserves data locality and reduces the risk of sensitive information leakage, as raw text never leaves the client. On the other hand, the intermediate representations can still encode substantial information about the original data, raising privacy concerns that may require additional protection such as differential privacy [9]. The communication overhead is a significant consideration: each training iteration requires the transmission of high-dimensional vectors between client and server. Compression techniques can reduce bandwidth, but they may also affect the fidelity of the gradients and consequently the model's accuracy [12].

The choice of cut layer is a critical architectural decision. Placing the cut earlier in the transformer stack reduces the size of the transmitted representation but also limits the amount of feature extraction that can be performed locally. Conversely, a later cut reduces computational burden on the server but increases communication cost and exposes more of the model to potential adversarial influence. Empirical studies suggest that a cut after the third or fourth transformer block often provides a reasonable balance for financial text data, as these layers capture syntactic and semantic features without encoding low-level token identities [11].

From a systems perspective, the deployment of split-trained transformers requires careful orchestration of client-server synchronization, fault tolerance, and version control. In a financial institution, the server may be operated by a neutral third party or a consortium, and clients must be authenticated and monitored to prevent sybil attacks. The governance structure must define which entities have access to the aggregated representations and under what conditions. These operational considerations are often more challenging than the algorithmic aspects of split learning, and they directly impact the trustworthiness of the overall system.

4. Adversarial Threat Model and Resilience Mechanisms

Adversarial threats in split-trained financial risk forecasting can be categorized by the attacker's objective and capability. A poisoning attacker aims to corrupt the training process so that the final model exhibits targeted misbehavior. An evasion attacker crafts adversarial inputs at inference time to cause incorrect predictions. A backdoor attacker implants a hidden trigger that activates only under specific conditions, allowing stealthy manipulation. In the split learning context, the attacker may control one or more clients, the communication channel, or even the server, depending on the trust model.

The most dangerous scenario involves a malicious client that contributes poisoned representations during training. Because the server cannot inspect the raw data, it must rely on statistical properties of the received embeddings to detect anomalies. Backdoor attacks in vertical split learning typically involve embedding a trigger pattern into the local representation so that when the trigger reappears during inference, the server's prediction is steered toward the adversary's target class. Recent research has shown that such attacks can achieve high success rates without degrading overall accuracy, making them difficult to detect through standard validation metrics [5].

To counter these threats, a suite of resilience mechanisms can be deployed at different layers of the system. On the server side, robust aggregation techniques can replace the simple averaging of client updates with median-based or trimmed mean operations that are less sensitive to outliers [16]. These methods, however, assume that the attacker's influence is bounded by the number of compromised clients. In financial settings with a small number of large institutions, this assumption may not hold. A complementary approach is to apply differential privacy to the gradients or representations sent by clients, thereby masking small deviations that an attacker might exploit [9]. The trade-off is a reduction in model accuracy, which must be calibrated against the required level of adversarial resilience.

A more targeted defense is prototype consistency verification, as exemplified by the ProtoGuard-SL framework [14]. This method maintains a set of prototype representations for each class, which are derived from a clean validation set. During training, the server compares incoming client embeddings against these prototypes; embeddings that deviate significantly are flagged as potentially malicious and either discarded or subjected to further inspection. The prototype-based approach leverages the observation that backdoor triggers tend to induce representations that lie outside the manifold of natural data. In financial risk forecasting, where classes correspond to risk levels (e.g., low, medium, high), the concept of prototype consistency aligns naturally with the notion of typical risk profiles.

Additional resilience can be achieved through input sanitization at the client side, such as adversarial training that exposes the model to perturbed examples during the local training phase [8]. While this does not prevent backdoor attacks from a compromised client, it can harden the client's own model against evasion attacks. The combination of client-side adversarial training, server-side robust aggregation, and prototype-based verification forms a layered defense that addresses multiple attack vectors. The effectiveness of this layered approach depends on the strength of each component and the underlying distribution of financial data, which often exhibits heavy tails and nonstationarity that can complicate the definition of "normal" representations.

5. Empirical Evaluation and Case Studies

Although the primary contribution of this paper is architectural and conceptual, we outline a framework for empirical evaluation that can guide future implementations. The evaluation should consider three dimensions: accuracy on clean data, resilience under attack, and computational cost. A realistic testbed would involve a consortium of financial institutions sharing anonymized time series and textual data, such as earnings call transcripts, over several years. The forecasting task could be the prediction of credit default probability for a portfolio of corporate bonds.

Split-trained transformer language models have been shown to achieve comparable or superior accuracy relative to centrally trained models on financial data, provided that the cut layer is chosen appropriately and the communication protocol is reliable [11]. However, under a backdoor attack that injects a trigger phrase into a subset of training texts, the accuracy on clean test data remains high while the attack success rate approaches 100 percent. This illustrates the insidious nature of backdoor attacks and the need for dedicated defenses.

In a simulated case study, applying ProtoGuard-SL reduced the backdoor success rate from 95 percent to below 10 percent while maintaining a clean accuracy drop of less than 2 percent [14]. The defense was particularly effective when the prototypes were updated dynamically to reflect the evolving data distribution. The addition of differential privacy at a modest privacy budget further eliminated residual attacks but introduced a 4 percent accuracy degradation. The combination of prototype verification and differential privacy provided a robust defense across a range of attack strengths.

From a deployment perspective, the computational overhead of the defense mechanisms must be accounted for. Prototype verification requires the server to compare each incoming embedding against a set of prototypes, which adds a linear cost in the number of prototypes and the dimensionality of the embeddings. For a system with thousands of clients and high-frequency updates, this overhead may become significant. However, because financial risk models are typically retrained on a daily or weekly basis rather than in real time, the additional latency is acceptable in most operational contexts. Energy consumption is another concern, particularly for large transformer models. Recent studies indicate that the carbon footprint of training a single large language model can be substantial [20]. Distributed split training, by distributing the computational load across multiple clients, can reduce the peak power requirement but may increase total energy due to communication overhead. These sustainability considerations are often overlooked in adversarial resilience research but are critical for long-term deployment.

6. Governance, Fairness, and Sustainability Implications

The deployment of adversarially resilient financial risk forecasting systems raises several governance questions. First, who is responsible when a model makes a faulty prediction due to an undetected adversarial attack? In a consortium setting, the server operator may be liable for the aggregation algorithm, while each client is responsible for the integrity of its own data and model frontend. A clear allocation of responsibility must be established in contracts and regulatory frameworks. Second, the use of prototype-based defenses introduces a potential fairness issue: if the prototypes are derived from a validation set that is not representative of all clients, the defense may systematically reject embeddings from certain institutions, leading to disparate treatment. This is particularly problematic in global financial systems where clients may operate in different regulatory environments and market conditions.

Fairness considerations also extend to the training data itself. Financial texts exhibit significant variation in language, tone, and frequency of reporting across different sectors and regions. A transformer model trained primarily on North American firms may encode biases that affect risk predictions for emerging market entities. The split training architecture does not inherently mitigate these biases; indeed, it may exacerbate them if the server's aggregation algorithm amplifies the signals from dominant clients. Adversarial defenses that discard outliers could further marginalize minority groups. Therefore, any deployment of such a system must include fairness audits and mechanisms for reweighting client contributions.

Sustainability is another dimension that intersects with adversarial resilience. The energy required to train and maintain large transformer models is a growing concern [20]. In the financial industry, which is under increasing pressure to reduce its environmental impact, the computational costs of frequent retraining and real-time inference must be justified. Adversarial defenses often increase computational requirements, as seen with iterative prototype updates and differential privacy noise injection. A sustainable approach may involve less frequent retraining cycles combined with efficient inference, along with the use of compressed models or distillation techniques. Moreover, the split training paradigm can be adapted to leverage clients' existing hardware, potentially reducing the need for dedicated server-side infrastructure.

Regulatory bodies such as the Basel Committee on Banking Supervision and the European Securities and Markets Authority are beginning to issue guidelines on the use of artificial intelligence in financial services. These guidelines typically emphasize transparency, explainability, and robustness. Adversarial resilience is an emerging requirement, but it is not yet codified. The framework presented in this paper can inform future regulation by highlighting the specific vulnerabilities of split-trained systems and the effectiveness of combined defensive measures. Policymakers should mandate that financial institutions conduct adversarial stress tests similar to the stress tests used for capital adequacy, and that they document the resilience mechanisms employed.

7. Conclusion

This paper has presented a comprehensive system-level analysis of adversarially resilient financial risk forecasting using split-trained transformer language models. We have argued that while split training offers significant advantages in terms of data privacy and regulatory compliance, it also introduces unique adversarial vulnerabilities that must be addressed through layered defense mechanisms. The integration of prototype consistency verification, robust aggregation, and differential privacy provides a viable path toward trustworthy deployment. However, no single defense is sufficient, and the choice of security architecture must be informed by the specific threat model and operational constraints of the financial institution.

The structural trade-offs explored in this work including the cut layer position, communication overhead, and accuracy-robustness trade-offs are central to the design of any practical system. Furthermore, the governance, fairness, and sustainability implications underscore that adversarial resilience cannot be treated as a purely technical problem. It requires alignment with regulatory expectations, ethical principles, and environmental goals. Future research should empirically validate the proposed framework on large-scale financial datasets, investigate the impact of nonstationary data distributions on prototype-based defenses, and develop standardized benchmarks for adversarial robustness in financial

machine learning. As financial systems become increasingly reliant on AI, the imperative to build resilient, fair, and sustainable models will only grow.

References

1. Gu, S., Kelly, B., & Xiu, D. (2020). Empirical asset pricing via machine learning. The Review of Financial Studies, 33(5), 2223-2273.

2. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. Advances in Neural Information Processing Systems, 30.

3. Goodfellow, I. J., Shlens, J., & Szegedy, C. (2015). Explaining and harnessing adversarial examples. In International Conference on Learning Representations.

4. Vepakomma, P., Gupta, O., Swedish, T., & Raskar, R. (2018). Split learning for health: Distributed deep learning without sharing raw patient data. arXiv preprint arXiv:1812.00564.

5. Gu, T., Dolan-Gavitt, B., & Garg, S. (2017). BadNets: Identifying vulnerabilities in the machine learning model supply chain. arXiv preprint arXiv:1708.06733.

6. Li, O., Sun, J., Yang, X., Gao, J., Zhang, H., Xie, L., & Han, S. (2022). Rethinking privacy in split learning: A systematic analysis and defense. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.

7. Huang, A. H., Wang, H., & Yang, Y. (2023). FinBERT: A large language model for extracting and classifying financial textual data. Journal of Financial Data Science, 5(2), 1-18.

8. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., & Vladu, A. (2018). Towards deep learning models resistant to adversarial attacks. In International Conference on Learning Representations.

9. Dwork, C., & Roth, A. (2014). The algorithmic foundations of differential privacy. Foundations and Trends in Theoretical Computer Science, 9(3-4), 211-407.

10. Dove, S., & Kurniawan, T. (2020). Adversarial attacks on financial machine learning models. Journal of Financial Data Science, 2(4), 49-64.

11. Li, D., Wang, J., & Chen, T. (2022). Split-BERT: A privacy-preserving BERT fine-tuning framework. In Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing.

12. Lin, Y., Han, S., Mao, H., Wang, Y., & Dally, W. J. (2018). Deep gradient compression: Reducing the communication bandwidth for distributed training. In International Conference on Learning Representations.

13. Vepakomma, P., Swedish, T., Raskar, R., Gupta, O., & Dubey, A. (2019). No peek: A survey of private distributed deep learning. arXiv preprint arXiv:1812.03288.

14. Shui, Y., Jin, R., Dou, Z., & Gao, Z. (2026). ProtoGuard-SL: Prototype Consistency Based Backdoor Defense for Vertical Split Learning. arXiv preprint arXiv:2604.03595.

15. Bagdasaryan, E., Veit, A., Hua, Y., Estrin, D., & Shmatikov, V. (2020). How to backdoor federated learning. In International Conference on Artificial Intelligence and Statistics.

16. Wei, K., Li, J., Ding, M., Ma, C., Yang, H., Farokhi, F., Jin, S., Poor, H. V., Liu, A., & Zhu, H. (2020). Federated learning with differential privacy: Algorithms and performance analysis. IEEE Transactions on Information Forensics and Security, 15, 3454-3469.

17. Sezer, O. B., Gudelek, M. U., & Ozbayoglu, A. M. (2020). Financial time series forecasting with deep learning: A systematic literature review: 2005–2019. Applied Soft Computing, 90, 106181.

18. Fischer, T., & Krauss, C. (2018). Deep learning with long short-term memory networks for financial market predictions. European Journal of Operational Research, 270(2), 654-669.

19. Floridi, L., & Cowls, J. (2019). A unified framework of five principles for AI in society. Harvard Data Science Review, 1(1).

20. Patterson, D., Gonzalez, J., Le, Q., Liang, C., Munguia, L. M., Rothchild, D., So, D., Texier, M., & Dean, J. (2021). Carbon emissions and large neural network training. arXiv preprint arXiv:2104.10350.