

DeFiRank: Federated Pairwise Ranking for Governance in Decentralized Financial Networks

Zahra Batool^{*†}, Michael Duchesne[†], Kaiwen Zhang^{*}, Ulrich Aivodji^{*†}

^{*}École de technologie supérieure, Montréal, Canada

[†]T-RIZE GROUP, Montréal, Canada

[‡]Mila - Quebec AI Institute, Montréal, Canada

Email: {zahra.batool, kaiwen.zhang, Ulrich.aivodji}@etsmtl.ca, michael.duchesne@t-rize.io

Abstract—Existing governance approaches in decentralized financial networks rely on manual review processes conducted by human committees, which introduces subjective bias, creates a scalability bottleneck under high network volume, and fails to capture complex behavioural patterns and evolving network dynamics. To address these limitations, we propose DeFiRank, a federated learning-based pairwise ranking framework in which multiple institutions collaboratively train a ranking model without sharing their private transaction data. This collaborative learning enables the model to capture network-wide behavioural patterns that signify strong network participation and directly contribute to the network effect, including diversity of counterparties, usage of multiple applications, and frequency of transactions. The proposed system ranks providers based on the authenticity of their actions, promoting genuine contributors and demoting those exhibiting suspicious behaviour. We evaluate DeFiRank on a real-world blockchain transaction dataset. Results demonstrate that the federated approach achieves ranking performance modestly below the centralized baseline (NDCG@15: ≈ 0.87 vs. ≈ 0.92) while preserving data privacy and matching its ranking stability (Kendall τ : ≈ 0.70 for both regimes), producing reliable governance decisions, including provider promotion and demotion (648 and 39 unique providers, respectively), without requiring centralized access to sensitive transaction data.

Index Terms—Pair-wise-ranking, Federated Learning, Financial network

I. INTRODUCTION

Corporate governance in decentralized financial networks refers to the set of rules and mechanisms through which entities are regulated and overseen by financial institutions and intermediaries to maintain transparency, stability, and accountability [1]. Traditionally, corporate governance has relied heavily on manual processes and human judgment, making it slow, inconsistent, and difficult to scale across large decentralized networks [2]. Machine learning (ML) has since transformed governance, enabling automated, data-driven decisions at scale [3].

Traditionally, training an ML model requires data to be collected at a central server [4]. Data privacy regulations such as GDPR restrict how sensitive financial data may be gathered and shared, creating tension with centralized training: dispersed financial institutions must relinquish ownership of proprietary transaction records to a single server, raising data sovereignty and institutional trust concerns.

The underlying problem is fundamentally one of reputation and Sybil-resistance: determining whether a given provider is a genuine, authentic contributor to the network, as opposed to a manipulative or low-value actor. Approaches to this problem in the literature include automated scoring mechanisms analogous to Elo or Glicko rating systems [5], [6], as well as Sybil-tolerant reputation systems designed specifically for decentralized governance and tokenomics [7]. In practice, however, current decentralized financial networks address this problem through manual governance: human committees periodically evaluate providers, promoting authentic contributors to featured status and demoting others [8]. This provides stable rankings but depends on a human-crafted, single-party scoring function with predetermined weights, introducing systemic bias and struggling to adapt as network behaviour and applications evolve, creating a lag between network dynamics and governance decisions.

Crucially, most provider-to-provider interactions in a decentralized financial network are private and never observed by any single centralized entity. As a result, the network-wide patterns indicative of authentic behaviour, such as the global distribution of application usage or typical counterparty diversity, cannot be learned from a central vantage point. Federated Learning (FL) is therefore a natural fit for this setting: it allows a global ranking model to learn from patterns emerging across private, localized transactions held by multiple hosts, without ever requiring those hosts to expose their raw data [9], [10]. This preserves institutional privacy while enabling the capture of network-wide behavioural trends that are invisible to any single actor.

In this context, authenticity of actions refers to how genuine and legitimate a provider's behaviour is within the network. The objective is to rank providers such that those engaging in diverse, normal financial activity score higher, while providers exhibiting suspicious or highly constrained behaviour score lower.

Learning-to-rank is a well-established paradigm in information retrieval, with pairwise methods receiving particular attention for their ability to directly optimise relative ordering between entities [11], [12]. Pairwise ranking trains a model to minimise ranking inversions, that is, to reduce mistakes in the relative preference ordering between pairs [12]. In

corporate governance, this translates naturally to comparative decisions: given two providers, which demonstrates higher authenticity? This formulation allows the model to identify which features, such as counterparty diversity, transaction volume, or application engagement, are most indicative of genuine network contribution.

To overcome the limitations of existing manual scoring approaches, we propose **DeFiRank**, a federated pairwise ranking framework for governance in decentralized financial networks. DeFiRank replaces the manual epoch scoring function with a machine-learned ranker that captures the features signifying strong network participation, including diversity of counterparties, usage of multiple applications, and frequency of transactions, while preserving the privacy of each provider’s transaction data. The main contributions of this work are:

- 1) We propose a privacy-preserving federated pairwise ranking framework that enables decentralized financial networks to collaboratively learn provider rankings without sharing private transaction data (Section III).
- 2) We design a dynamic governance decision pipeline with adaptive promotion and demotion thresholds that respond continuously to network health (quality trends) and network size (growth and contraction) (Section III-C).
- 3) We demonstrate through experiments that the federated approach closely matches centralized ranking stability (Kendall τ : ≈ 0.70 for both regimes) while trailing modestly in top-rank precision (NDCG@15: ≈ 0.87 vs. ≈ 0.92 centralized), and producing trustworthy governance decisions comprising 648 promotion and 39 demotion candidates, with 100% promotion consistency and 72% demotion consistency over 2+ consecutive days (Section IV-F).

The remainder of this paper is organised as follows. Section II reviews related work. Section III describes the proposed DeFiRank architecture. Section IV presents experimental results and comparisons with baseline approaches. Finally, Section V concludes the paper and discusses directions for future work.

II. LITERATURE REVIEW

This section reviews previous related work on pairwise ranking in federated learning and AI-driven corporate governance.

A. Pairwise Ranking in Federated Learning

Federated learning-to-rank remains underexplored, particularly in financial and governance contexts. [13] introduces the Federated Pairwise Differentiable Gradient Descent (FPDGD) protocol for online learning-to-rank. [14] applies federated pairwise ranking to recommendation systems, proposing a protocol based on matrix factorization and Bayesian personalized ranking. [15] implements point-wise federated ranking to optimize Firefox address-bar URL suggestions, keeping browsing history and input on-device throughout training. [16] addresses compositional pairwise risk optimization in federated learning

through gradient decoupling. These methods target query-document retrieval or recommendation, where labels derive from clicks or ratings.

B. AI-Driven Corporate Governance

AI-driven governance has gained traction for reducing human bias and improving decision-making consistency. [17] shows AI-based frameworks improve real-time decisions across complex business ecosystems, reflecting a broader shift from rule-based to learning-based governance [1]. However, existing work predominantly assumes centralized data access, limiting applicability to decentralized settings. DeFiRank addresses this gap by combining learned governance scoring with the privacy guarantees of federated learning.

The aforementioned studies provide the algorithmic machinery for privacy-preserving ranking but have not been applied to governance settings where heterogeneity is structural and outputs drive real promotion and demotion decisions, while the AI-governance literature addresses governance objectives but assumes centralized data access. DeFiRank is positioned at the intersection of these two gaps: a federated pairwise ranking framework purpose-built for decentralized financial governance.

III. PROPOSED METHOD

DeFiRank involves three entities:

- 1) **Host**: a data-holding institution, used interchangeably with federated client.
- 2) **Provider**: the governed unit ranked and promoted/demoted, whose daily features are computed at a single host and never combined across hosts.
- 3) **Party**: end user, whose activity aggregates into provider features.

DeFiRank operates only on provider-day aggregated features (Section IV-A4) and never observes party-level data or data from another host. It consists of three phases: (1) *federated temporal pairwise ranking*, partitioning the provider dataset across non-IID clients with structurally distinct governance perspectives; (2) *local training*, where each client trains a temporal ranking model and the server aggregates updates via FedAvg (Section III-B); and (3) *governance decision making*, where the trained model’s scores drive data-driven promotion and demotion decisions (Section III-C). Fig. 1 provides an overview of the full pipeline.

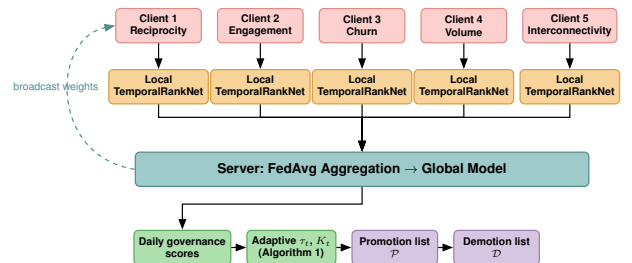


Fig. 1: Overview of DeFiRank

A. Federated Temporal Pairwise Ranking

To evaluate DeFiRank under realistic multi-institution conditions, we simulate a federated deployment by partitioning the provider dataset, comprising 965,894 provider-day records spanning 102,520 unique providers over 184 days, across 5 non-IID clients. We emphasize that this partitioning is a methodological step for experimentation only: in a real deployment, no such partitioning would be required, as each client would natively hold its own private transaction data and never observe the data held by other clients. Here, the non-IID structure is constructed to reflect realistic behavioral heterogeneity across institutions; each client is assigned a different validator weight vector that prioritizes a specific governance dimension: reciprocity-heavy, engagement-heavy, churn-heavy, volume-heavy, and interconnectivity-heavy (formalized in Eq. 6). This means the same provider pair can receive opposite labels across clients, since each evaluates provider quality from a different structural perspective, approximating realistic conditions where institutions prioritize different behaviour patterns based on their own vantage point.

B. Local Training

Local training is performed for pairwise ranking using TemporalRankNet. For each provider, *build_sequences()* constructs a sliding 7-day rolling window of 13 governance features (10 base + 3 trend), yielding input sequences of shape $(N, 7, 13)$. Early days with insufficient history are zero-padded to maintain consistent sequence length. Pairwise training labels are generated by comparing governance scores of provider pairs within the same day using each client’s structural weight vector, yielding a total of 9,657,042 training pairs across all clients.

Every client maintains a local TemporalRankNet model combining LSTM temporal encoding with RankNet pairwise ranking. A two-layer LSTM processes each provider’s 7-day sequence day by day, and the final hidden state h_7 yields a 64-dimensional embedding of the provider’s governance behavior over the window. RankNet takes two same-day providers A and B , encodes each independently via the shared LSTM, and optimizes pairwise binary cross-entropy loss:

$$\mathcal{L} = \text{BCE}(s_A - s_B, y_{AB}) \quad (1)$$

where $y_{AB} = 1$ if provider A has higher governance quality than B , and 0 otherwise.

Each communication round proceeds as follows: (1) the server broadcasts the current global model weights to all 5 clients, (2) each client trains locally on its partitioned data for one local epoch using the BCE pairwise loss with the Adam optimizer ($\eta = 0.001$, batch size 2048), (3) clients transmit updated weights back to the server, no raw data, labels, or gradients are shared, preserving data privacy, (4) the server aggregates all client weights via FedAvg, computing a weighted average proportional to each client’s dataset size. After aggregation, server-side evaluation computes NDCG@5/10/15 and Kendall τ using a neutral validator scoring function with a

fixed seed, deliberately independent of any individual client’s local weight distribution. Training runs for 100 communication rounds, with the global model converging within the first 10 rounds.

C. Governance Decision Making

After training, the model assigns a predicted governance score to each provider daily. Algorithm 1 summarizes the four-stage pipeline used to convert these scores into promotion and demotion decisions over a $B = 30$ -day baseline period.

Baseline anchors. The demotion anchor K_{base} is set at the largest score gap among sorted baseline featured apps (line 10). The promotion anchor τ_{promo} escalates from the 25th to 50th percentile of baseline active-featured scores, then to a ratio-based estimate, if pinned at the 0.10 floor. The minimum ratio anchor ρ_{min} is the median promotion ratio among baseline qualifiers above $\max(\tau_{\text{promo}}, 0.30)$, raised further if τ_{promo} saturates at its 0.60 ceiling; this yields $\rho_{\text{min}} = 0.6231$ in our dataset.

Daily scaling. Both anchors are scaled daily by the network health and size ratios (Algorithm 1, lines 17, 18): the promotion threshold τ_t rises when the network is thriving and falls when it is struggling, while the demotion cutoff is scaled by network size and bounded by the current featured-set size.

Promotion and demotion. Featured apps with near-zero activity are excluded before computing promotion ratios, preventing inactive competitors from inflating non-featured apps’ apparent quality. A non-featured app is a raw promotion candidate if its ratio meets τ_t ; candidates are confirmed only if they qualify on 2+ consecutive calendar days with mean ratio at or above ρ_{min} (line 28), filtering transient signals from sustained ones. Featured apps ranked below the effective cutoff are flagged for demotion.

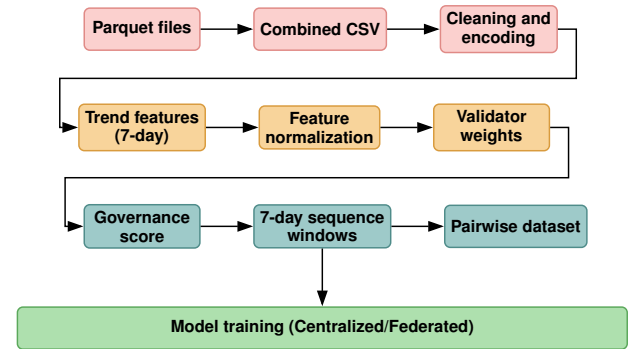


Fig. 2: End-to-end governance pipeline.

IV. EVALUATION

A. Dataset

We collected data from a blockchain platform, specifically the Canton Network, over 184 days (2025-08-24 to 2026-02-23), during which users interact through multiple financial applications. The data is publicly available and was collected from the Super Validator Set of the global synchronizer. The dataset contains transaction logs maintained by host

Algorithm 1 Dynamic Governance Decision Pipeline

Require: Daily provider scores, featured-status flags, activity; baseline period $B = 30$ days

Ensure: Promotion list $\mathcal{P}$, demotion list $\mathcal{D}$

```
1: Step 1: Baseline anchors (first B days)
2:  $\tau_{\text{promo}} \leftarrow$  25th percentile of active featured scores
3: if stuck at floor (0.10) then
4:   retry at 50th percentile
5: end if
6: if still stuck then
7:   fall back to 75th percentile of outrank ratios
8: end if
9: Clip  $\tau_{\text{promo}}$  to  $[0.10, 0.60]$ 
10:  $K_{\text{base}} \leftarrow$  rank with the largest score gap between featured providers
11:  $\rho_{\text{min}} \leftarrow$  median outrank ratio of providers clearing  $\tau_{\text{promo}}$ , clipped to  $[0.25, 0.70]$ 
12: if  $\tau_{\text{promo}}$  stuck at ceiling (0.60) then
13:   raise  $\rho_{\text{min}}$  to  $\geq 0.685$ 
14: end if

15: Step 2: For each day
16: for each day do
17:   Scale  $\tau_{\text{promo}}$  by today's health ratio  $\rightarrow$  today's threshold, clipped to  $[0.10, 0.60]$ 
18:   Scale  $K_{\text{base}}$  by today's size ratio  $\rightarrow$  today's cutoff, clipped to  $[3, 150]$ , capped by featured count
19:   Drop near-inactive featured providers; skip day if fewer than 3 remain
20:   Flag featured providers below the cutoff for  $\mathcal{D}$ 
21:   for each non-featured provider do
22:     Flag as candidate if its outrank fraction meets today's threshold
23:   end for
24: end for

25: Step 3: Reliability check
26: for each candidate do
27:   Group qualifying days into runs of consecutive dates
28:   if a run has  $\geq 2$  days and average ratio  $\geq \rho_{\text{min}}$  then
29:     Confirm provider for  $\mathcal{P}$ 
30:   end if
31: end for
32: return  $\mathcal{P}, \mathcal{D}$ 
```

nodes in a decentralized financial structure, where each host stores the activity of the parties it manages. Rather than raw transaction data, the dataset contains aggregated behavioural features derived from financial transaction histories, obtained by processing the raw on-chain records into provider-day aggregates. The data was stored in Parquet file format, where each file contains one day of network activity. All files were merged and exported to CSV format for analysis. The dataset contains 28 features and 965,894 samples, where each row represents a provider on a specific date. Each feature captures a different aspect of provider behaviour, including transaction activity, engagement levels, churn, and network connectivity. Two governance-related binary flags, *is_featured* and *is_validator*, indicate the provider's current role in the network.

1) *Data Preprocessing:* Time-related columns are converted to standard datetime format. Rows with missing

provider or *record_date* identifiers are discarded, as these fields are required to construct per-provider temporal sequences. Duplicate records sharing the same *provider* and *record_date*, arising from overlapping daily snapshots, are removed, retaining only the entry with the most recent *record_time*. Missing values in the remaining numeric fields are imputed with zero. Numeric columns are cast to the correct data types. Boolean governance flags such as *is_featured* and *is_validator* are encoded as 0 and 1.

2) *Feature Selection:* We selected 13 features that best represent provider stability and governance quality, rather than using all 28. The selected features are: *total_cc_volume*, *global_engagement_rate*, *interconnectivity*, *reciprocity_rate*, and *total_churn_rate* to capture core transaction and network behaviour; *new_transfer_users*, *transfer_user_base_increase*, *unique_directional_transfers*, *average_n_transfers_per_user*, and *new_beneficiaries* to record user growth dynamics and engagement depth.

Since static features such as *total_cc_volume*, *global_engagement_rate*, and *transfer_users* capture only a snapshot in time and do not reflect whether a provider is growing or declining, we additionally define three trend features, computed on the preprocessed data described above, as:

$$\text{trend} = \frac{x_t - x_{t-7}}{|x_{t-7}| + 10^{-6}} \quad (2)$$

where x_t is the feature value today and x_{t-7} is the value seven days prior. This yields *volume_trend*, *engagement_trend*, and *user_growth_trend*. A positive value indicates growth, a negative value indicates decline, and zero indicates no change. For example, if transactions rise from 2,000 to 5,000, the trend is $+1.5$ (150% growth); if they fall from 9,000 to 5,000, the trend is -0.44 (44% decline). All trend values are clipped to $[-5, +5]$ to prevent extreme outliers. If a provider has fewer than seven days of history, the trend is set to zero. This makes governance evaluation dynamic by giving higher weight to growing providers relative to declining ones.

3) *Feature Normalization:* The features are normalized using z-score normalization:

$$x' = \frac{x - \mu}{\sigma} \quad (3)$$

where μ and σ are the mean and standard deviation computed on the data, normalization statistics are saved to *feature_mu.csv* and *feature_sigma.csv* to ensure consistent scaling is applied across all settings.

4) *Governance Score Computation:* Since the dataset does not contain a ranking column, a governance score is computed as a weighted linear combination of the normalized features and used to generate labels during pairwise dataset construction:

$$\text{gov_score} = \sum_{i=1}^n w_i f_i \quad (4)$$

where f_i are the normalized features and w_i are feature weights representing governance importance. Rather than using a single fixed weight vector, weights are generated dynam-

ically to simulate different validator preferences. Starting from a fixed base weight vector, where *reciprocity_rate* receives the highest base weight of 0.18, reflecting its central importance as a governance signal, small Gaussian noise is added to each weight:

$$w_i = \frac{\max(\bar{w}_i + \epsilon_i, 10^{-3})}{\sum_{j=1}^n \max(\bar{w}_j + \epsilon_j, 10^{-3})}, \quad \epsilon_i \sim \mathcal{N}(0, 0.03^2) \quad (5)$$

where $\bar{w}_i$ is the base weight for feature i and ϵ_i is added noise. The noisy weights are clipped to a minimum of 10^{-3} to ensure all features retain a non-negative contribution, then renormalized to sum to 1. This simulates the real-world scenario where different validators assign slightly different importance to governance features, where one validator may prioritize transaction volume while another prioritizes reciprocity. For the federated setting, each client further relabels its local partition of the pairwise pool using a client-specific weight vector, generated by the same noisy-renormalization mechanism but starting from one of five fixed structural base profiles, each emphasizing a single governance dimension, reciprocity, engagement, churn, volume, or interconnectivity:

$$w_i^{(c)} = \frac{\max(p_i^{(c)} + \epsilon_i^{(c)}, 10^{-3})}{\sum_{j=1}^n \max(p_j^{(c)} + \epsilon_j^{(c)}, 10^{-3})}, \quad \epsilon_i^{(c)} \sim \mathcal{N}(0, 0.02^2) \quad (6)$$

where $p^{(c)}$ is the fixed base profile for client c , with the corresponding governance dimension assigned a base weight of 0.35-0.40 and the remaining features assigned small residual weights. This is the mechanism referenced in Section III: since each client starts from a different $p^{(c)}$, the same provider pair can be assigned opposite labels by different clients.

The governance score is used only to generate labels during pairwise dataset construction and is never used as a direct model input.

5) *Pairwise Dataset Construction*: Generating all possible provider pairs is computationally infeasible given 965,894 samples. We therefore adopt a sampling-based pairwise approach. For each *record_date* (governance round), providers active on that day are sampled in pairs without restriction on featured status, allowing non-featured-versus-featured, featured-versus-featured, and non-featured-versus-non-featured comparisons to all occur. The number of sampled pairs per day is capped at $\min(5N_t, 5000)$, where N_t is the number of providers active on day t , to bound memory usage while scaling pair count with daily provider activity.

For each sampled pair (i, j) , governance scores are compared, and the pair is discarded if $|\text{gov_score}_i - \text{gov_score}_j| < 10^{-6}$ to avoid near-tied comparisons that provide no meaningful ranking signal. Otherwise, a single row is written with label 1 if $\text{gov_score}_i > \text{gov_score}_j$, and label 0 otherwise, with provider i 's features placed in columns suffixed *_1* and provider j 's in columns suffixed *_2*.

Each side of a pair stores the full 7-day rolling sequence described in Section III-B rather than a single-day snapshot, yielding $13 \times 7 = 91$ feature columns per side (182 feature columns total) plus the label column. This sampling procedure,

applied across all governance rounds in the dataset, produces a final pairwise dataset of 9,657,042 record pairs.

B. Experimental Setup

All experiments ran on a Compute Canada *Nibi* node¹ (NVIDIA H100 80GB) using the Flower framework [18]. The pairwise dataset is split 80:20 for training/test, with a separate date-level 80:20 holdout for ranking evaluation to prevent temporal leakage. Training uses batch size 2048, Adam ($\eta = 0.001$), and *BCEWithLogitsLoss*. Each provider's input is a 7-day rolling window of 13 governance features, shape $(N, 7, 13)$, used for both training and evaluation; NDCG@5/10/15 and Kendall τ are computed each epoch against the governance score ground truth. Centralized training runs 100 epochs; federated training simulates 5 clients over 100 rounds, with non-IID heterogeneity introduced via a Dirichlet distribution ($\alpha = 0.5$) and each client relabeling its partition with a client-specific structural weight profile.

C. Model Architecture

Both architectures follow the same pairwise formulation [11]: each provider's 7-day, 13-feature sequence is encoded into a single embedding, mapped to a scalar score, and compared between two same-day providers. **TemporalRankNet**. A two-layer LSTM (hidden dim 64) processes the sequence day by day; the final hidden state h_7 is mapped to a score via Linear(64→32), ReLU, Dropout(0.1), Linear(32→1) (Fig. 3). **AttentionRankNet**. A 4-head self-attention encoder with sinusoidal positional encoding processes the 7-day window; the mean-pooled output is mapped to a score via a 3-layer MLP head. Both models optimize Eq. 1 over 9,657,042 training pairs, where $y_{AB} = 1$ if provider A has higher governance quality than B , and 0 otherwise.

D. Performance Metrics

We evaluate ranking quality using Normalized Discounted Cumulative Gain (NDCG) and the Kendall rank correlation coefficient τ . The model is optimized using the pairwise ranking loss defined in Eq. 1, followed by evaluation using the two ranking metrics below.

1) *Normalized Discounted Cumulative Gain*: NDCG evaluates both the relevance and position of items in the ranked list [19], penalizing ranking errors at the top of the list more heavily, well-suited to governance decisions where top-ranked providers receive featured status and greater network visibility. NDCG values range from 0 to 1, with higher values indicating better ranking quality. We report NDCG@5, NDCG@10, and NDCG@15.

2) *Kendall τ* : Kendall τ [20] measures the rank correlation between predicted and ground-truth orderings across all provider pairs:

$$\tau = \frac{C - D}{\binom{n}{2}} \quad (7)$$

where C and D are the numbers of concordant and discordant pairs among all $\binom{n}{2}$ provider pairs, and $\tau \in [-1, 1]$.

¹<https://docs.alliancecan.ca/wiki/Nibi>

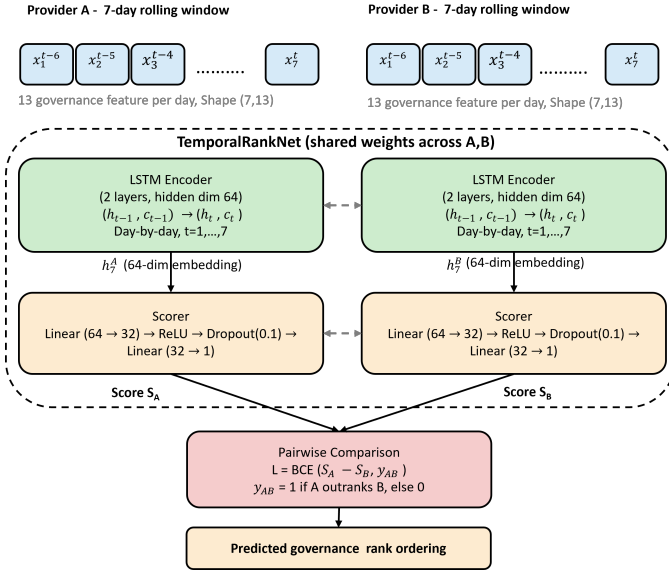


Fig. 3: TemporalRankNet architecture: shared LSTM encoder and scorer map each provider’s 7-day feature window to a score, trained via pairwise BCE loss.

While NDCG@K emphasizes top-ranking precision, Kendall τ captures overall pairwise ranking consistency across the full provider list, offering a complementary view of global ranking quality.

E. Baselines

We compare two training regimes for both architectures (Section IV-C). **Centralized**: trained on the full dataset at a single server, with direct access to all provider data. **Federated**: each client trains locally and shares only model updates without exposing raw transaction data. Crossing both regimes with both architectures yields four settings: Centralized LSTM-RankNet (C-L), Federated LSTM-RankNet (F-L), Centralized Attention-RankNet (C-A), and Federated Attention-RankNet (F-A).

F. Results

Fig. 4 (a),(b) report centralized LSTM-RankNet performance. NDCG increases in a steady high range for the duration of training to 0.82-0.88 in a single epoch after getting nearly random initial values. The final values for NDCG@k are (0.87, 0.91, 0.92) with loss converging to (0.006-0.007) by epoch 100 and no train/test gap. Kendall τ converges within 5-6 epochs at 0.70, since NDCG@k is sensitive to reordering near the cutoff while τ aggregates over all pairs, smoothing local noise.

Fig. 4(c),(d) report federated performance. Round 1 shows a temporary drop before recovering (NDCG@5: 0.43→0.37, τ : 0.39→0.55): each client weights the 13 governance features differently (Section III), so the same provider pair can receive opposite labels across clients, and the first aggregation must reconcile these conflicting local models before a stable global ranking emerges. NDCG@k converges by round 6 to

(0.86, 0.87, 0.88). τ converges by round 6 at 0.70, comparable to the centralized.

Fig. 5(a),(b) report centralized Attention-RankNet performance. NDCG rises to 0.63-0.72 within two epochs but continuously fluctuating and less stable than LSTM-RankNet. NDCG plateaus lower and wider than LSTM-RankNet with final values (0.75, 0.82, 0.84) below LSTM-RankNet’s peak (≈ 0.92 NDCG@15). However, τ is unaffected and converging to 0.706, similar to LSTM-RankNet.

Fig. 5(c),(d) report federated Attention-RankNet performance. NDCG@5 starts near-random and shows continuous fluctuations. τ converges similar to LSTM-RankNet. NDCG@k continues oscillating without a clear trend ending at round 100 with (0.72, 0.74, 0.77) and $\tau=0.70$, both below LSTM-RankNet’s federated plateau.

Federated training differs from centralized in one key way: each client scores providers using its own feature weights, while centralized uses one shared weight vector (Section III). This applies equally to both architectures, so it cannot explain why Attention-RankNet converges roughly five times slower than LSTM-RankNet under federation. The likely reason is architectural: LSTM naturally emphasizes recent days, which suits the 7-day trend features, while Attention averages all days equally, diluting that signal, especially when clients disagree on which features matter. Attention also has more parameters than a 7-day sequence really needs, which likely explains its training instability under centralized learning too. These results motivate using LSTM-RankNet for the governance decision pipeline (Section III-C).

Table I summarizes the baseline anchors and final candidate counts for centralized and federated LSTM-RankNet. Both compute their anchors from their own baseline period (Section III-C), so the two settings use independently data-driven thresholds rather than a shared cutoff. Centralized has lower promotion and minimum ratio anchors, which reflect scores spanning the full corpus and settling into a narrower baseline distribution, whereas federated has stricter anchors, including the promotion anchor saturating at its upper clip bound, which reflect clients’ conflicting weight vectors (Section III) producing a baseline score distribution that is higher and tighter, which the reliability filter (Algorithm 1, line 28) turns into a stricter bar.

In the centralized setting (Table I), all qualifying promotion candidates showed fully consistent signals over 2+ days, suggesting the model reliably learns well-governed non-featured applications under the looser anchor. Demotion candidates split roughly two-thirds consistently poor (low transaction quality, high churn, low reciprocity) to one-third borderline, appearing only once.

In the federated setting (Table I), far fewer providers qualify for promotion than centralized, but at a higher average ratio, consistent with the stricter federated anchor: fewer providers clear the bar, but those that do outscore a larger share of active featured competitors. The demotion anchor and top- K bounds (Algorithm 1, lines 10 and 18) are identical across both regimes, since they depend only on the elbow position in the

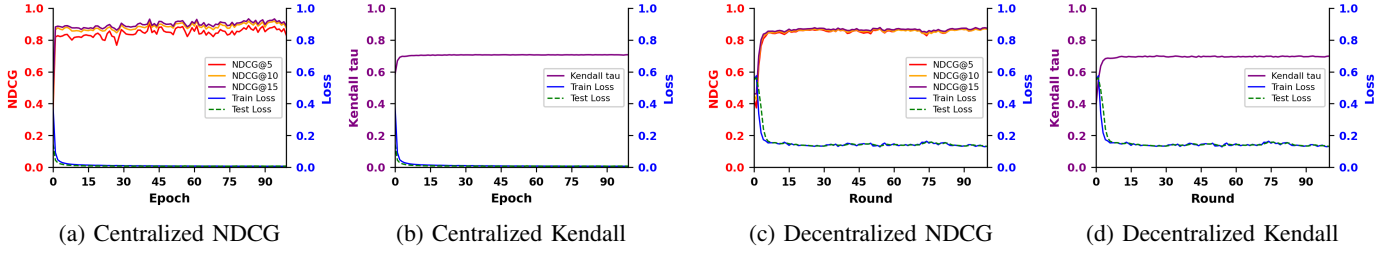


Fig. 4: LSTM-RankNet

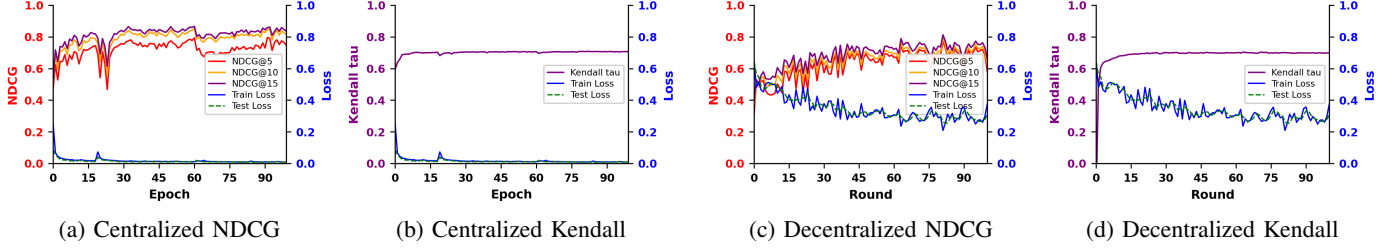


Fig. 5: Attention-RankNet

TABLE I: Baseline anchors, final candidate counts, Promotion and demotion candidate statistics for Centralized LSTM-RankNet and Federated LSTM-RankNet.

Metric	Centralized	Federated
Promotion anchor τ_{promo}	0.445	0.600 (ceiling)
Min ratio anchor ρ_{min}	0.546	0.700
Baseline qualifying apps	1,301	1,212
Demotion anchor K_{base}	150 (ceiling)	150 (ceiling)
Reliable promotion candidates	15,587	2,257
Unique promotion providers	4,009	648
Mean promotion ratio	0.645	0.772
Consistent (2+ days)	4,009 (100%)	648 (100%)
Demotion candidate rows	183	183
Unique demotion providers	35	39
Consistent (2+ days)	24 (69%)	28 (72%)
Borderline (1 day)	11 (31%)	11 (28%)

sorted featured-app scores and the network-size ratio, not the validator weighting scheme, reflected in both settings flagging the same number of demotion rows. Despite the shared anchor, federated demotion shows marginally higher consistency than centralized, suggesting federated aggregation produces slightly more temporally stable demotion signals, consistent with the tighter Kendall τ band observed for federated training.

Fig. 6a shows the promotion ratio distribution for centralized LSTM-RankNet. The dashed red line marks $\tau_t = 0.10$, the lower bound of the promotion threshold’s clipped range (Algorithm 1, line 17); the actual daily threshold varies within this range depending on network health, so this line is the most permissive value τ_t can take, not a fixed cutoff. The distribution concentrates between 0.3 and 1.0, peaking near 0.65-0.7, with the dashed orange line marking the mean (Table I): a promoted provider outscores active featured apps well above both this floor and the minimum ratio anchor

(Table I) on average. Few providers fall in 0.3-0.4, indicating the reliability filter successfully separates sustained quality differentials from transient threshold crossings.

Fig. 6b shows the federated distribution, right-shifted and narrower, peaking near 0.75-0.80, with far fewer total candidates than centralized (Table I). Both differences trace to federated’s stricter minimum ratio anchor, itself a consequence of clients’ conflicting structural weight vectors (Section III) producing a baseline score distribution concentrated higher and tighter than the centralized full-corpus baseline. Since the reliability filter rejects any candidate below this anchor, a stricter anchor mechanically raises the surviving population’s mean ratio while shrinking its size, exactly the pattern observed. Neither distribution shows providers clustering near the 0.10 floor, confirming the filter is effective regardless of the anchor value a baseline period yields.

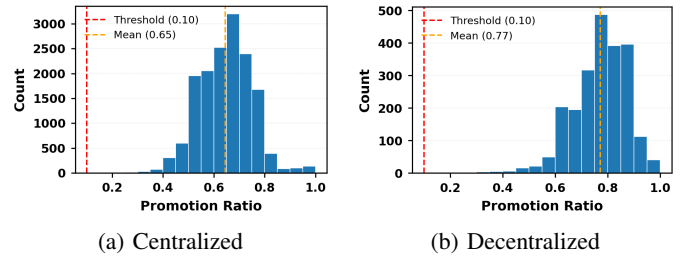


Fig. 6: LSTM-RankNet promotion ratio distribution comparison

Fig. 7(a) and (b) show the daily promotion threshold (τ_t) and demotion cutoff (K_t) for the centralized and federated models. Both use the same health and size ratios from *gov_score*; only the anchor values differ because they are computed from each model’s predictions.

The demotion curves are identical because both models

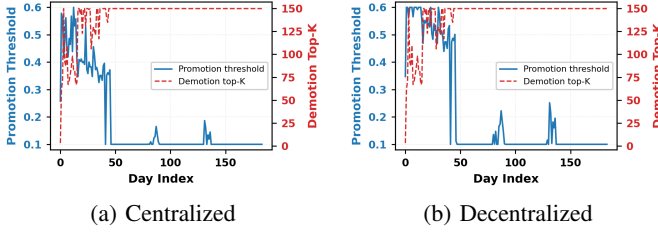


Fig. 7: LSTM-RankNet Centralized and Decentralized thresholds comparison

reach the maximum $K_{\text{base}} = 150$, so $K_t = 150 \times \lambda_t^{\text{size}}$. The promotion thresholds differ because the federated anchor is at the maximum value, while the centralized anchor is lower.

After day 40, both thresholds stay at the minimum ($\tau_t = 0.10$) as network health declines and network size grows rapidly. Small peaks around days 87 and 131 appear in both models due to improved health values, but the federated model reaches slightly higher peaks because its larger anchor makes it more sensitive near the minimum threshold.

V. CONCLUSION

We presented DeFiRank, a privacy-preserving federated pairwise ranking framework that replaces static, manually-crafted epoch scoring functions in decentralized financial networks. By training TemporalRankNet, combining LSTM temporal encoding with RankNet pairwise ranking across 13 governance features, over 9.6 million pairwise comparisons distributed across five non-IID clients, DeFiRank learns network-wide behavioural patterns without exposing private transaction data. We additionally evaluated an attention-based architecture, AttentionRankNet, which achieved comparable Kendall τ but consistently lower and less stable NDCG, motivating LSTM-RankNet as the architecture used for governance decisions. A dynamic governance pipeline adaptively adjusts promotion and demotion thresholds in response to network health and size, yielding 648 unique promotion candidates and 39 unique demotion candidates under federation, with 100% promotion consistency and 72% demotion consistency over 2+ consecutive days. Experimental results on a real-world blockchain dataset show that DeFiRank achieves ranking performance modestly below the centralized baseline (NDCG@15: ≈ 0.87 vs. ≈ 0.92) while reaching comparable ranking stability (Kendall τ : ≈ 0.70 for both regimes), demonstrating that federated learning preserves institutional privacy without sacrificing the consistency of governance signals. Future work will explore integrating tokenomic incentive mechanisms to couple governance decisions with on-chain rewards, adopting personalised federation strategies to better accommodate non-IID client heterogeneity, and incorporating formal differential privacy guarantees for deployment in regulated financial environments.

ACKNOWLEDGMENT

This research is supported by T-RIZE GROUP, Montréal, Canada, and Mitacs Grant IT44869.

REFERENCES

- [1] G. Munir, M. A. Ali, S. Mahmood, and G. Mahmood, “Ai-driven corporate governance: A structured literature review of ai applications in financial and non-financial sectors,” *Qualitative Research Journal for Social Studies*, vol. 2, no. 1, pp. 79–96, 2025.
- [2] R. V. Aguilera and M. Ruiz Castillo, “Toward an updated corporate governance framework: Fundamentals, disruptions, and future research,” *BRQ Business Research Quarterly*, vol. 28, no. 2, pp. 336–348, 2025.
- [3] X. Ma, Z. Wen, T. Hu, and X. Zheng, “Algorithmic governance: Machine learning-driven insights into corporate decision-making and risk control,” in *Proceedings of the 4th International Conference on Artificial Intelligence and Intelligent Information Processing*, 2025, pp. 907–911.
- [4] Z. Batool, B. Buyukates, R. Nourmohammadi, and K. Zhang, “Privacy-enhancing technologies for federated learning,” in *Federated Learning Systems: Towards Privacy-Preserving Distributed AI*. Springer, 2025, pp. 129–146.
- [5] M. E. Glickman, “The glicko system,” *Boston University*, vol. 16, no. 8, p. 9, 1995.
- [6] A. Berg, “Statistical analysis of the elo rating system in chess,” *Chance*, vol. 33, no. 3, pp. 31–38, 2020.
- [7] B. Nasrulin, G. Ishmaev, and J. Pouwelse, “Meritrack: Sybil tolerant reputation for merit-based tokenomics,” in *2022 4th Conference on Blockchain Research & Applications for Innovative Networks and Services (BRAINS)*. IEEE, 2022, pp. 95–102.
- [8] A. T. Polcumpally, K. K. Pandey, A. Kumar, and A. Samadhiya, “Blockchain governance and trust: A multi-sector thematic systematic review and exploration of future research directions,” *Heliyon*, vol. 10, no. 12, 2024.
- [9] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, “Communication-efficient learning of deep networks from decentralized data,” in *Artificial intelligence and statistics*. PMLR, 2017, pp. 1273–1282.
- [10] M. Aledhari, R. Razzak, R. M. Parizi, and F. Saeed, “Federated learning: A survey on enabling technologies, protocols, and applications,” *IEEE access*, vol. 8, pp. 140 699–140 725, 2020.
- [11] C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. Hullender, “Learning to rank using gradient descent,” in *Proceedings of the 22nd international conference on Machine learning*, 2005, pp. 89–96.
- [12] T.-Y. Liu, “Learning to rank for information retrieval,” *Foundations and Trends® in Information Retrieval*, vol. 3, no. 3, pp. 225–331, 2009.
- [13] S. Wang, B. Liu, S. Zhuang, and G. Zuccon, “Effective and privacy-preserving federated online learning to rank,” in *Proceedings of the 2021 ACM SIGIR international conference on theory of information retrieval*, 2021, pp. 3–12.
- [14] V. W. Anelli, Y. Deldjoo, T. Di Noia, A. Ferrara, and F. Narducci, “Federated recommender systems with learning to rank,” in *SEBD*, 2021, pp. 71–82.
- [15] F. Hartmann, S. Suh, A. Komarzewski, T. D. Smith, and I. Segall, “Federated learning for ranking browser history suggestions,” *arXiv preprint arXiv:1911.11807*, 2019.
- [16] Z. Guo, R. Jin, J. Luo, and T. Yang, “Fedxl: Provable federated learning for deep x-risk optimization,” in *International Conference on Machine Learning*. PMLR, 2023, pp. 11 934–11 966.
- [17] S. Mahajan, I. R. Nagesh, B. B. Mishra, S. Sarkar, S. P. Gangadharan, V. Agarwal *et al.*, “Artificial intelligence and strategic governance: Enabling real-time decisions in complex business ecosystems,” *Management:(Montevideo)*, vol. 3, p. 14, 2025.
- [18] D. J. Beutel, T. Topal, A. Mathur, X. Qiu, T. Parcollet, and N. D. Lane, “Flower: a friendly federated learning research framework (2020),” *arXiv preprint arXiv:2007.14390*, 2007.
- [19] K. Järvelin and J. Kekäläinen, “Cumulated gain-based evaluation of ir techniques,” *ACM Transactions on Information Systems (TOIS)*, vol. 20, no. 4, pp. 422–446, 2002.
- [20] M. G. Kendall, “A new measure of rank correlation,” *Biometrika*, vol. 30, no. 1-2, pp. 81–93, 1938.