International Journal of Safety and Security Engineering
Vol. 16, No. 10, October, 2026, pp. 2439-2449
Journal homepage: http://iieta.org/journals/ijsse

Multi-Year Information, Not Recurrence: A Controlled Study of LSTM Models for Financial Statement Fraud Detection

Shatha Majeed Hameed[1*], Hala Majeed Hameed[2]

[1] Department of Financial Affairs, Al-Mustansiriya University Presidency, Baghdad 14022, Iraq
[2] College of Education, Al-Mustansiriya University, Baghdad 14022, Iraq

Corresponding Author Email: shathamajeedhameed@uomustansiriyah.edu.iq

Copyright: ©2026 The authors. This article is published by IIETA and is licensed under the CC BY 4.0 license (http://creativecommons.org/licenses/by/4.0/).

https://doi.org/10.18280/ijsse.161020

Received: 9 August 2026
Revised: 12 October 2026
Accepted: 25 October 2026
Available online: 31 October 2026

Keywords:
class imbalance, controlled simulation, financial statement fraud, long short-term memory, model comparison, permutation importance, sequence modeling, temporal attribution

ABSTRACT

Recurrent models are widely proposed for financial statement fraud detection on the premise that creative accounting unfolds across several reporting periods. This study separates two claims that are usually merged: that multi-year information helps, and that learned recurrence is the way to exploit it. Three synthetic panel generators with documented manipulation processes — cumulative drift, abrupt onset and conditional masking — are used to compare a long short-term memory (LSTM) against logistic regression on the terminal year and on concatenated five-year features, under firm-level partitioning, twenty replications, validation-selected thresholds and seven imbalance-aware metrics compared through paired tests and difference intervals. Multi-year information helps: under abrupt onset Precision-Recall Area Under the Curve (PR-AUC) rises by 0.084 over terminal-year regression ($p < 0.001$), whether the five years feed an LSTM or a concatenated logistic regression (0.084). Recurrence does not: across all three generators, six recurrent configurations and five training sizes, the LSTM never significantly exceeds concatenated-feature regression in ranking, and a non-recurrent network on the same inputs matches its threshold-metric advantage under conditional masking. Replicated attribution shows that random-donor permutation overstates the terminal year's unique contribution 1.9- to 4.5-fold relative to refitting and that splicing year blocks from other firms can push performance below chance; matched-donor permutation and refitting show early years to be informative but largely redundant. Results describe simulated data with known generators and are not claims about archival enforcement data.

1. INTRODUCTION

Financial statement fraud detection has progressed from single-period statistical scores toward flexible machine learning models, and more recently toward recurrent architectures motivated by the observation that manipulation is spread across consecutive reporting periods. That motivation bundles together two logically separate propositions. The first is that information beyond the most recent fiscal year improves detection. The second is that a recurrent architecture is the appropriate mechanism for exploiting that information. Empirical studies generally test the two jointly, by comparing a recurrent model against a single-period baseline. A favourable result is then read as support for recurrence, when it may only support the weaker proposition that multiple years matter.

Distinguishing the two requires a third model: a non-recurrent classifier that receives exactly the same multi-year inputs. If a recurrent model beats a single-period baseline but not a concatenated-feature baseline, the benefit is attributable to input scope rather than to architecture. This study is built around that three-way comparison.

Separating the propositions also requires knowing how the temporal signal was produced, which archival data cannot provide because the manipulation mechanism is unobserved. The study therefore uses synthetic panels whose generating processes are specified in advance, including one process in which the timing of a discontinuity carries information that the final year does not summarise. The purpose is not to estimate achievable detection performance, which is a property of real data, but to establish which comparisons support which conclusions.

The findings are as follows. Multi-year information does help: the long short-term memory (LSTM) significantly exceeds terminal-year regression under cumulative drift and, more markedly, under abrupt onset. Recurrence, however, is not the source of that advantage: logistic regression on concatenated five-year features realises the same gain, and across three generators, six recurrent configurations and five training-set sizes the LSTM never significantly exceeds it in ranking performance; where it does exceed it on threshold-dependent metrics, a non-recurrent nonlinear network on the same inputs matches it. Separately, temporal attribution is shown to depend on how the removed years are replaced: random-donor permutation overstates the unique importance of the terminal year and, when year blocks are spliced, can

drive performance below chance, whereas matched-donor permutation and drop-and-refit agree that early years are informative but largely redundant with later ones.

Relative to the previous version of this work, two conclusions are revised. Earlier experiments at a smaller sample size found no significant difference between the LSTM and terminal-year regression, and attributed this to a sufficiency property of the terminal state. With twenty replications at a larger sample the difference is significant, and the sufficiency claim is withdrawn. The permutation evidence that supported it is also superseded by the replicated, boundary-matched and refit-based attribution reported here.

All experiments use synthetic data. No filings, Accounting and Auditing Enforcement Releases (AAERs) or other archival records were used, and no claim is made about detection performance on real enforcement-labeled fraud, auditor workflows or deployment behaviour. Section 6 states the resulting limits on generalisation.

2. RELATED WORK

The literature is organised below by the methodological question each group of studies answers, rather than study by study, because the relevant distinction for the present work is what a given design can and cannot establish about temporal modeling.

The dominant formulation treats each firm-year independently. Statistical risk scores [1, 2] estimate manipulation likelihood from one year of accrual-quality and performance variables, and the machine learning literature that followed largely preserved this formulation while varying the classifier family: probabilistic neural networks and kernel methods [3, 4], comparative studies of trees, networks and Bayesian models [5, 6], and boosted ensembles with undersampling [7]. Two limitations of this group matter here. Multi-year context enters only through hand-engineered features, so the value of temporal information is never estimated directly; and evaluation has historically relied on single splits, whose instability under low fraud base rates has been documented [8] and which motivated the replication-based protocol.

A second group improves the representation rather than the temporal formulation. Textual disclosure signals [9, 10] add information orthogonal to ratios and yield consistent gains, but they leave the single-filing formulation intact. Work on deep models in adjacent financial prediction tasks [11] documents that such gains typically require corpora far larger than enforcement samples. This group establishes that richer inputs help; it does not establish that sequential structure is the productive axis of enrichment.

Recurrent [12, 13] and attention-based [14] architectures have been applied to multi-year ratio sequences, and the transition is surveyed in the study [15]. The characteristic design compares a recurrent model against a single-period baseline. This design cannot separate the contribution of multi-year input scope from that of the recurrent mechanism, because the two differ simultaneously between the compared models. A concatenated-feature control, which holds input scope fixed while removing recurrence, is rarely reported. This is the gap the present study addresses.

Rare positives make imbalance treatment integral to design [16-18], with resampling dominant in applied work [19, 20]. Cost-sensitive weighting is adopted here because interpolating between financial ratio observations can produce economically incoherent firms. On evaluation, ROC analysis is optimistic under imbalance relative to precision-recall analysis [21, 22], the Matthews correlation coefficient is preferable to F1 [23], model comparison requires testing across replications rather than single-split point estimates [24], and calibration is a distinct property from ranking [25]. Attribution methods [26] and arguments for interpretable models in high-stakes settings [27] motivate the permutation analysis, whose sensitivity to permutation scheme under dependent data is itself known [28].

Table 1 classifies the literature by what each design can establish about temporal modeling. The pattern is that studies with multi-year inputs do not include a non-recurrent multi-year control, so the input-scope and architecture effects remain confounded, and that replication-based statistical comparison is uncommon in the applied fraud literature even though its necessity is established in the methodological literature.

Table 1. What existing designs can establish about temporal modeling

Group	Input Scope	Non-Recurrent Multi-Year Control	Replicated Tests	Can Separate Scope from Architecture
Statistical scores [1, 2]	Single year	Not applicable	No	No
Classical machine learning [3-6]	Single year	Not applicable	Partly [6]	No
Ensembles on enforcement data [7, 8]	Single year	Not applicable	Partly [8]	No
Text-enriched models [9, 10]	Single filing	No	No	No
Sequence models [12-15]	Multi-year	Rarely reported	Rarely	No
This work	Multi-year	Yes (concatenated)	Yes (20 seeds)	Yes

3. METHODOLOGY

3.1 Problem formulation

Each firm *i* is a sequence of observations over *T* consecutive fiscal years, as in Eq. (1).

$$X_i = \left(x_i^{(1)}, \ldots, x_i^{(T)}\right), \quad x_i^{(t)} \in \mathbb{R}^d \tag{1}$$

where, $x_i^{(t)}$ holds *d* financial ratios for firm *i* in year *t*, with *T* = 5 and *d* = 5, and $y_i \in \{0,1\}$ records whether the generating process applied a manipulation mechanism. The task is to learn $f_\theta: \mathbb{R}^{T\times d} \to [0,1]$ producing $\hat{y}_i = f_\theta(X_i)$.

3.2 Generating processes

Three processes are specified so that the temporal structure of the signal is known. All use the same five ratios — current ratio, debt-to-equity, net income margin, asset turnover, receivables turnover — over five years, and all add independent measurement noise after generation so that

classes overlap.

Generator A applies cumulative drift: manipulated firms accumulate per-year increments in which liquidity and asset turnover decline, leverage rises, margin is inflated and receivables turnover falls, so effects compound monotonically. The base rate is 8%.

Generator B applies abrupt onset: healthy firms follow first-order autoregressive mean-reverting dynamics around a firm-specific baseline, and manipulated firms behave normally until a randomly drawn onset year, after which a step change is applied. The base rate is 5%. Here the position of the discontinuity, rather than an accumulated magnitude, carries information.

Generator C applies conditional masking: manipulation occurs only in years when the firm's current ratio falls below its own baseline, so the signal depends on a within-sequence state.

Generator A embeds the assumption implicit in the cumulative-drift argument, and its terminal observation necessarily contains much of the preceding trajectory. Generators B and C are included precisely because that property does not hold for them, and all three are used for native training and testing.

3.3 Models compared

Three models receive identical partitions, identical imbalance weighting and identical threshold-selection procedure, differing only as intended.

LSTM: a recurrent classifier over the full sequence. Linear Regression (LR)-last: logistic regression on the terminal year only, representing the single-period paradigm. LR-flat: logistic regression on all five years concatenated into a 25-dimensional vector, which receives the same information as the LSTM without learned temporal structure. LR-flat is the decisive control, since the LSTM-versus-LR-last contrast varies input scope and architecture simultaneously, whereas the LSTM-versus-LR-flat contrast isolates architecture. Because LR-flat is linear, a fourth model, Multilayer Perceptron (MLP)-flat, a feed-forward network with two hidden layers of 32 units, ReLU activations and dropout 0.3 on the same 25-dimensional vector, trained with the same loss, optimiser, epochs and threshold rule as the LSTM, is used to separate nonlinearity from recurrence.

3.4 Recurrent architecture

A recurrent network updates a hidden state as in Eq. (2), and backpropagation through time multiplies Jacobians as in Eq. (3), which for long sequences causes gradients to vanish or explode [29, 30].

$$h^{(t)} = \phi\left(W_h h^{(t-1)} + W_x x^{(t)} + b\right) \tag{2}$$

$$\frac{\partial \mathcal{L}}{\partial h^{(1)}} = \frac{\partial \mathcal{L}}{\partial h^{(T)}} \prod_{t=2}^{T} \frac{\partial h^{(t)}}{\partial h^{(t-1)}} \tag{3}$$

LSTM networks [12] regulate a cell state through multiplicative gates, given by Eqs. (4)–(9).

$$f^{(t)} = \sigma\left(W_f x^{(t)} + U_f h^{(t-1)} + b_f\right) \tag{4}$$

$$i^{(t)} = \sigma\left(W_i x^{(t)} + U_i h^{(t-1)} + b_i\right) \tag{5}$$

$$\tilde{c}^{(t)} = \tanh\left(W_c x^{(t)} + U_c h^{(t-1)} + b_c\right) \tag{6}$$

$$c^{(t)} = f^{(t)} \odot c^{(t-1)} + i^{(t)} \odot \tilde{c}^{(t)} \tag{7}$$

$$o^{(t)} = \sigma\left(W_o x^{(t)} + U_o h^{(t-1)} + b_o\right) \tag{8}$$

$$h^{(t)} = o^{(t)} \odot \tanh\left(c^{(t)}\right) \tag{9}$$

where, $\sigma(\cdot)$ is the logistic sigmoid and $\odot$ elementwise multiplication. Because the update in Eq. (7) is additive in $c^{(t-1)}$, gradients propagate with limited attenuation when the forget gate approaches unity.

3.5 Class imbalance and preprocessing

A weighted binary cross-entropy loss is used as in Eq. (10), with the positive weight set to the inverse class frequency ratio as in Eq. (11) and applied equivalently to the regression baselines.

$$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N}[w^{+} y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)] \tag{10}$$

$$w^{+} = \frac{N^{-}}{N^{+}} \tag{11}$$

Features are standardized as in Eq. (12), with statistics estimated on the training partition only. Partitioning is at the firm level, stratified on the label, in a 60/20/20 split, because within-firm ratios are strongly autocorrelated across adjacent years.

$$z = \frac{x - \mu}{\sigma} \tag{12}$$

3.6 Optimization

Parameters are optimized with adaptive moment estimation [31] as in Eqs. (13) and (14), for 20 epochs with batch size 32 and dropout [32] of 0.3 on the terminal hidden state.

$$\begin{aligned} m^{(t)} &= \beta_1 m^{(t-1)} + (1 - \beta_1) g^{(t)}, \\ v^{(t)} &= \beta_2 v^{(t-1)} + (1 - \beta_2)(g^{(t)})^2 \end{aligned} \tag{13}$$

$$\theta^{(t)} = \theta^{(t-1)} - \eta \frac{\widehat{m}^{(t)}}{\sqrt{\widehat{v}^{(t)}} + \epsilon} \tag{14}$$

3.7 Metrics and threshold selection

Threshold-dependent metrics are defined in Eqs. (15)–(17) and the Matthews correlation coefficient in Eq. (18), with specificity and balanced accuracy also reported. Ranking quality is measured by Area Under the Receiver Operating Characteristic Curve (AUC-ROC), Eq. (19), and by Precision-Recall Area Under the Curve (PR-AUC), which is more sensitive to minority-class performance [21, 22]. Calibration is measured by the Brier score, Eq. (20).

$$\text{Precision} = \frac{TP}{TP + FP} \tag{15}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{16}$$

$$F_1 = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{17}$$

$$\text{MCC} = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{18}$$

$$\text{AUC} = \int_0^1 \text{TPR}\,(\tau)\, d\,\text{FPR}(\tau) \tag{19}$$

$$\text{BS} = \frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2 \tag{20}$$

For every model the threshold is selected on the validation partition by maximising F1 and then applied unchanged to the test partition, so that threshold choice is not a confound.

4. EXPERIMENTAL DESIGN

4.1 Protocol

Epoch-level monitoring and threshold selection use the validation partition exclusively; the test partition is evaluated once per replication after training and threshold selection are complete. Every experiment is replicated with independent data-generation and split seeds. Performance levels are reported as mean ± SD across replications. Every model comparison is reported as the mean paired difference Δ together with the half-width e of its 95% t-interval across replications ($\Delta \pm e$), and is tested with a two-sided paired Wilcoxon signed-rank test; where several configurations are compared with one reference, p-values are Holm-adjusted. With n replications the smallest attainable Wilcoxon p is 2^{1-n}, which is 0.002 for $n = 10$. All tables are generated by a single released script that stores every metric for every replication. Algorithms 1 and 2 give the procedures.

Algorithm 1. Preprocessing and sequence tensor construction

Require: panel P; horizon $T = 5$; dimension $d = 5$
Ensure: tensors of shape $(\cdot, T, d)$ and labels
$L \leftarrow$ firm-level label set from P
$(I\text{tr}, I\text{tmp}) \leftarrow \text{StratifiedSplit}(L, 0.6/0.4)$
$(I\text{va}, I\text{te}) \leftarrow \text{StratifiedSplit}(I\text{tmp}, 0.5/0.5)$
$(\mu, \sigma) \leftarrow$ statistics over Itr only, per Eq. (12)
for $S \in \{\text{tr}, \text{va}, \text{te}\}$ **do**
$Z_s \leftarrow$ standardize features of I_s, sorted by (i, t)
$X_s \leftarrow \text{Reshape}(Z_s, (|I_s|, T, d))$
end for
assert shape(Xtr) = $(|I\text{tr}|, T, d)$
return partitioned tensors and labels

Algorithm 2. Replicated training, threshold selection and testing

Require: generator G; seeds S; $E = 20$; $B = 32$; η; H; $p = 0.3$
Ensure: per-seed metrics for LSTM, LR-last, LR-flat
for each seed $s \in S$ **do**
$P \leftarrow G(N, \text{seed } s)$; partitions $\leftarrow$ Algorithm 1
$w^+ \leftarrow N^- / N^+$ per Eq. (11)
for epoch = 1 **to** E **do**
for each mini-batch (X_β, y_β) **do**
$\hat{z} \leftarrow \text{Linear}(\text{Dropout}(\text{RNN}\theta(X_\beta)))$ per Eqs. (4)-(9)
$\theta \leftarrow \text{Adam}(\theta, \nabla\theta\ \text{WeightedBCE}(\hat{z}, y_\beta; w^+); \eta)$
end for
end for
fit LR-last on year T; fit LR-flat on concatenated years
$\tau^* \leftarrow$ argmax F1 on validation, per model
evaluate test partition once at τ^* per Eqs. (15)-(18)
end for
return means, SDs, CIs and Wilcoxon tests over S

4.2 Experiments

E1 trains and tests the three models, and the MLP-flat control, natively within each generator, with 20 seeds and 2,000 firms. E2 sweeps six recurrent configurations, varying hidden size (16, 32, 64), depth (1, 2), cell type (LSTM, GRU) and bidirectionality, on generators A and B with 10 seeds, each compared with LR-flat by paired differences with Holm adjustment. E3 varies the training population from 500 to 8,000 firms with 10 seeds. E4 estimates feature and temporal importance on generators A and B inside every E1 replication, so that intervals reflect data generation and model training rather than permutation noise alone: 30 permutations are averaged within each replication, and summary statistics are taken across the 20 replications. Feature importance permutes a feature's whole firm trajectory. Temporal importance, the loss in test AUC, is estimated in three ways: random-donor permutation, in which a single year or the suffix block of years $k..5$ is copied from a randomly chosen test firm; matched-donor permutation, in which the donor is drawn at random from the ten test firms closest to the target in the adjacent unpermuted year or years, which keeps trajectories continuous at the splice boundary [28]; and drop-and-refit, in which the model is retrained without the year or block. Because suffix blocks are nested and all contain year 5, the contribution of year k is reported as the increment $\Delta(k) = I(k..5) - I(k{+}1..5)$, not as the block total $I(k..5)$.

5. RESULTS AND DISCUSSION

5.1 Native training and testing on each generator

Figure 1 shows the comparison of the four models trained and tested within each generator. Tables 2 and 3 report all seven metrics, Table 4 presents the paired comparisons for PR-AUC and MCC, and Table A1 in the Appendix provides the paired comparisons for every metric.

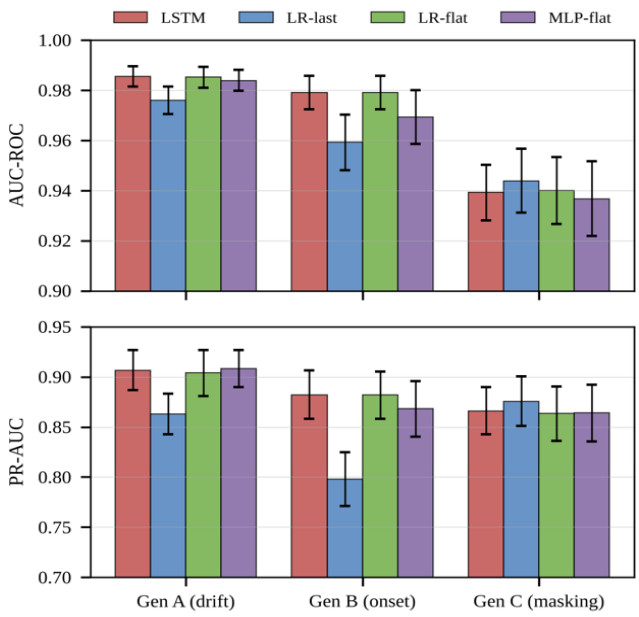

Figure 1. Native performance of the four models by generator, with 95% confidence intervals

Table 2. Native ranking and calibration performance by generator (20 seeds, N = 2,000, mean ± SD)

Gen.	Model	AUC-ROC	PR-AUC	Brier
A	LSTM	0.986 ± 0.009	0.907 ± 0.043	0.038 ± 0.011
	LR-last	0.976 ± 0.012	0.863 ± 0.043	0.058 ± 0.012
	LR-flat	0.985 ± 0.009	0.904 ± 0.049	0.040 ± 0.010
	MLP-flat	0.984 ± 0.009	0.908 ± 0.040	0.030 ± 0.007
B	LSTM	0.979 ± 0.014	0.882 ± 0.052	0.028 ± 0.011
	LR-last	0.959 ± 0.024	0.798 ± 0.057	0.065 ± 0.012
	LR-flat	0.979 ± 0.014	0.882 ± 0.050	0.032 ± 0.012
	MLP-flat	0.969 ± 0.023	0.868 ± 0.059	0.026 ± 0.009
C	LSTM	0.939 ± 0.024	0.866 ± 0.050	0.040 ± 0.007
	LR-last	0.944 ± 0.027	0.875 ± 0.053	0.052 ± 0.010
	LR-flat	0.940 ± 0.029	0.863 ± 0.058	0.052 ± 0.010
	MLP-flat	0.937 ± 0.032	0.864 ± 0.061	0.035 ± 0.009

Note: AUC-ROC =Area Under the Receiver Operating Characteristic Curve; PR-AUC = Precision-Recall Area Under the Curve; LR = Linear Regression; LSTM = long short-term memory; MLP = Multilayer Perceptron.

Table 3. Native threshold-dependent performance by generator (20 seeds, N = 2,000, mean ± SD)

Gen.	Model	F1	MCC	Specificity	Balance Accuracy
A	LSTM	0.810 ± 0.066	0.796 ± 0.070	0.980 ± 0.015	0.908 ± 0.042
	LR-last	0.754 ± 0.060	0.739 ± 0.065	0.976 ± 0.016	0.872 ± 0.049
	LR-flat	0.805 ± 0.050	0.791 ± 0.053	0.979 ± 0.011	0.909 ± 0.039
	MLP-flat	0.813 ± 0.057	0.801 ± 0.057	0.983 ± 0.011	0.901 ± 0.049
B	LSTM	0.801 ± 0.066	0.796 ± 0.068	0.990 ± 0.010	0.890 ± 0.041
	LR-last	0.732 ± 0.071	0.721 ± 0.076	0.989 ± 0.007	0.844 ± 0.035
	LR-flat	0.801 ± 0.069	0.793 ± 0.072	0.989 ± 0.007	0.901 ± 0.042
	MLP-flat	0.789 ± 0.085	0.783 ± 0.089	0.991 ± 0.008	0.877 ± 0.047
C	LSTM	0.848 ± 0.052	0.841 ± 0.053	0.995 ± 0.005	0.889 ± 0.038
	LR-last	0.835 ± 0.055	0.826 ± 0.059	0.990 ± 0.009	0.893 ± 0.039
	LR-flat	0.812 ± 0.068	0.804 ± 0.070	0.991 ± 0.009	0.874 ± 0.049
	MLP-flat	0.843 ± 0.041	0.835 ± 0.041	0.992 ± 0.007	0.894 ± 0.038

Notes: 1. Thresholds selected on the validation partition. 2. (Multi MLP-flat is the non-recurrent nonlinear control of Section 3.3. 3. Per-seed values for every metric are released with the implementation. MCC = Matthews Correlation Coefficient; LR = Linear Regression; LSTM = long short-term memory; MLP = Multilayer Perceptron.

Table 4. Paired comparisons of PR-AUC and MCC (20 seeds)

Gen.	Comparison	ΔPR-AUC ± *e*	*p*	ΔMCC ± *e*	*p*
A	LSTM − LR-last	+0.044 ± 0.016	<0.001	+0.057 ± 0.025	<0.001
	LSTM − LR-flat	+0.003 ± 0.011	0.812	+0.006 ± 0.027	0.778
	LR-flat − LR-last	+0.041 ± 0.014	<0.001	+0.051 ± 0.026	0.001
	LSTM − MLP-flat	−0.002 ± 0.009	0.729	−0.004 ± 0.024	0.717
B	LSTM − LR-last	+0.084 ± 0.025	<0.001	+0.075 ± 0.040	0.002
	LSTM − LR-flat	+0.000 ± 0.017	0.452	+0.003 ± 0.034	0.845
	LR-flat − LR-last	+0.084 ± 0.024	<0.001	+0.072 ± 0.036	<0.001
	LSTM − MLP-flat	+0.014 ± 0.023	0.165	+0.013 ± 0.044	0.421
C	LSTM − LR-last	−0.010 ± 0.010	0.064	+0.016 ± 0.018	0.098
	LSTM − LR-flat	+0.003 ± 0.011	0.756	+0.037 ± 0.021	0.003
	LR-flat − LR-last	−0.012 ± 0.011	0.033	−0.022 ± 0.026	0.102
	LSTM − MLP-flat	+0.002 ± 0.010	0.812	+0.007 ± 0.020	0.469

Notes: Δ ± *e* is the mean paired difference, first model minus second, with the half-width of its 95% confidence interval across replications; *p* is from the two-sided Wilcoxon signed-rank test. Table A1 reports all seven metrics with the test statistic *W*. PR-AUC = Precision-Recall Area Under the Curve; MCC= Matthews Correlation Coefficient; LR = Linear Regression; LSTM = long short-term memory; MLP = Multilayer Perceptron.

Against LR-last, the LSTM is significantly better on generators A and B on every metric except specificity, which is close to its ceiling (Table A1); the largest margin is under abrupt onset, where PR-AUC improves by 0.084 and MCC by 0.075 ($p \leq 0.002$). The decisive comparison is LR-flat against LR-last. On generator B it yields the same PR-AUC gain, +0.084 ± 0.024 ($p < 0.001$), and on generator A 0.041 against 0.044 for the LSTM, so the multi-year gain is available to a linear model without recurrence. Against LR-flat the LSTM shows no significant difference in AUC or PR-AUC on any generator; on generator B the PR-AUC difference is +0.000 ± 0.017, so the 95% interval excludes an LSTM advantage larger than 0.018.

Two exceptions concern threshold-dependent metrics and calibration. Under conditional masking the LSTM exceeds LR-flat in F1 (+0.036 ± 0.019, p = 0.003), MCC (+0.037 ± 0.021, p = 0.003) and balanced accuracy (+0.015 ± 0.013, p = 0.017), and on generators B and C its Brier score is lower than that of LR-flat (Table A1). Because generator C manipulates only when a within-firm condition holds, this advantage could reflect either recurrence or merely nonlinearity. MLP-flat separates the two: it matches the LSTM on generator C in F1 (+0.005 ± 0.018, p = 0.445), MCC (+0.007 ± 0.020, p = 0.469) and balanced accuracy (−0.006 ± 0.009, p = 0.355), and its Brier score is lower than the LSTM's on generators A and C. On no generator and no metric does the LSTM significantly

exceed both non-recurrent multi-year controls. On generator C the terminal year is itself competitive: LR-last does not differ significantly from the LSTM in AUC or PR-AUC, and exceeds LR-flat in PR-AUC (p = 0.033).

The previous version reported no significant LSTM advantage over LR-last and attributed this to approximate sufficiency of the terminal state. That conclusion was drawn at a smaller sample with lower power, and it does not survive replication at 2,000 firms: the sufficiency claim is withdrawn. What does survive is the narrower and better-identified claim that the advantage comes from multi-year input scope rather than from recurrence, because it is fully realised by a linear model on concatenated years.

5.2 Architecture and hyperparameter sensitivity

Figure 2 and Table 5 report six recurrent configurations against LR-flat as paired differences.

On generator A every configuration is below LR-flat, with differences from −0.029 to −0.010 PR-AUC, and none is significant after Holm adjustment (smallest adjusted p = 0.117). On generator B the best configuration, LSTM H16, exceeds LR-flat by +0.027 ± 0.029 PR-AUC (unadjusted p = 0.084, adjusted p = 0.387); no configuration differs significantly, and every AUC difference on generator B lies within ±0.003. Varying hidden size by a factor of four, adding depth or bidirectionality and substituting a GRU cell leave the comparison unchanged.

This is a bounded result and is stated as such. Six configurations within one architecture family cannot establish that no recurrent model would benefit, and the sweep does not cover attention architectures, longer sequences, or substantially larger datasets. The claim supported is that the absence of an advantage is not an artifact of one arbitrary configuration.

5.3 Training-set size

Table 6 and Figure 3 report performance from 500 to 8,000 firms.

The LSTM separates from LR-last as N grows, significantly so from 2,000 firms upward (Δ = +0.040, +0.055, +0.044 PR-AUC at 2,000, 4,000, 8,000 firms; p = 0.049, p = 0.002, p = 0.002), which is again consistent with multi-year information being valuable. It does not overtake LR-flat at any size: the LSTM − LR-flat difference is negative at every size and significantly so at 4,000 and 8,000 firms (−0.009 ± 0.007, p = 0.027; −0.015 ± 0.009, p = 0.004). The number of replications per size was raised from six to ten, which lowers the smallest attainable p from 0.031 to 0.002, but the intervals at 500 and 1,000 firms remain wide. The claim made here is correspondingly bounded: over this range these data do not support the hypothesis that the LSTM would overtake the concatenated baseline with more data. The stronger assertion made in an earlier version, that the sample-size explanation was falsified, overstated what an experiment of this size can establish and is withdrawn.

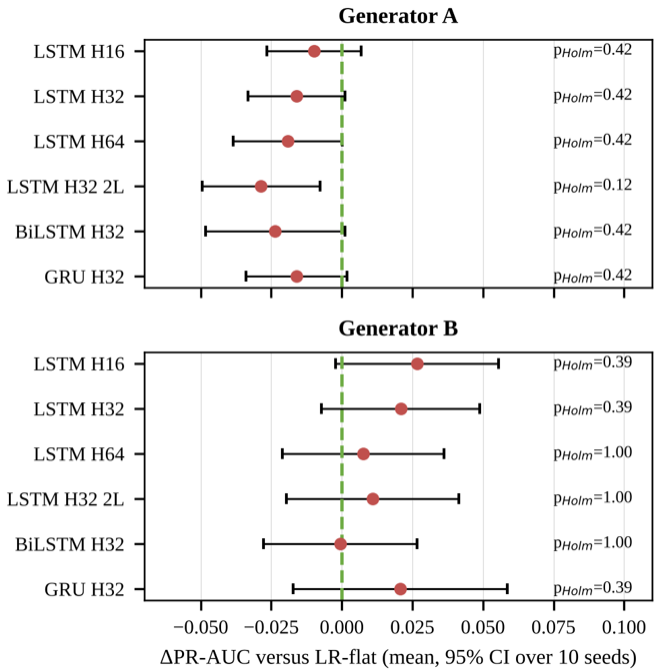

Figure 2. Paired PR-AUC difference of each recurrent configuration from LR-flat, with 95% confidence intervals and Holm-adjusted p-values

Note: PR-AUC = Precision-Recall Area Under the Curve; LR = Linear Regression.

Table 5. Architecture sweep: PR-AUC (mean ± SD) and paired difference from LR-flat (10 seeds, N = 2,000)

Configuration	PR-AUC	Δ vs LR-flat ± e	p	p_{Holm}
Generator A				
LSTM H16	0.894 ± 0.033	−0.010 ± 0.017	0.322	0.422
LSTM H32	0.888 ± 0.038	−0.016 ± 0.017	0.105	0.422
LSTM H64	0.885 ± 0.043	−0.019 ± 0.019	0.084	0.420
LSTM H32 2L	0.875 ± 0.040	−0.029 ± 0.021	0.020	0.117
BiLSTM H32	0.880 ± 0.042	−0.024 ± 0.025	0.105	0.422
GRU H32	0.888 ± 0.038	−0.016 ± 0.018	0.131	0.422
LR-flat (ref.)	0.904 ± 0.023	—	—	—
Generator B				
LSTM H16	0.893 ± 0.049	+0.027 ± 0.029	0.084	0.387
LSTM H32	0.887 ± 0.050	+0.021 ± 0.028	0.064	0.387
LSTM H64	0.874 ± 0.054	+0.008 ± 0.029	0.695	1.000
LSTM H32 2L	0.877 ± 0.058	+0.011 ± 0.031	0.375	1.000
BiLSTM H32	0.865 ± 0.059	−0.001 ± 0.027	0.695	1.000
GRU H32	0.887 ± 0.059	+0.021 ± 0.038	0.064	0.387
LR-flat (ref.)	0.866 ± 0.079	—	—	—

Notes: Δ ± e and p as in Table 4; p_{Holm} adjusts for the six configurations compared with LR-flat within each generator. The sweep uses its own data seeds, so LR-flat values differ slightly from Table 2. Note: PR-AUC = Precision-Recall Area Under the Curve; LR = Linear Regression; LSTM = long short-term memory.

Table 6. PR-AUC by training population (10 seeds, mean ± SD) and paired difference from LR-flat

N	LSTM	LR-Last	LR-Flat	LSTM − LR-Flat	*p*
500	0.848 ± 0.109	0.838 ± 0.129	0.880 ± 0.080	−0.032 ± 0.038	0.098
1,000	0.877 ± 0.093	0.868 ± 0.078	0.902 ± 0.074	−0.025 ± 0.050	0.322
2,000	0.900 ± 0.037	0.860 ± 0.057	0.904 ± 0.038	−0.005 ± 0.013	0.625
4,000	0.883 ± 0.026	0.828 ± 0.027	0.893 ± 0.027	−0.009 ± 0.007	0.027
8,000	0.897 ± 0.019	0.852 ± 0.028	0.912 ± 0.016	−0.015 ± 0.009	0.004

Note: PR-AUC = Precision-Recall Area Under the Curve; LR = Linear Regression; LSTM = long short-term memory.

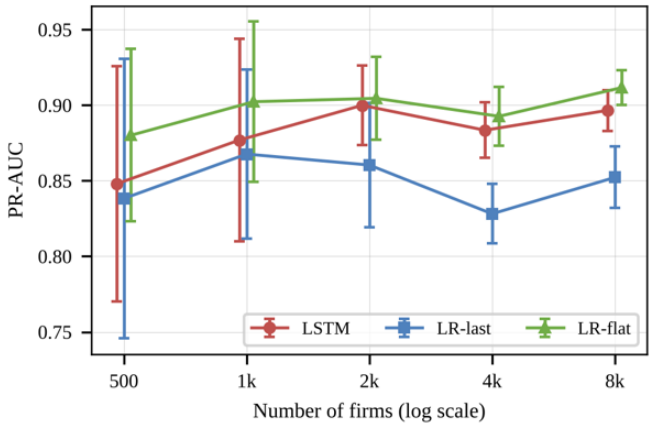

Figure 3. Precision-Recall Area Under the Curve (PR-AUC) against training population size (10 seeds), with 95% confidence intervals

5.4 Temporal attribution

Tables 7, 8 and Figure 4 report temporal attribution for generators A and B, estimated inside the 20 native replications with intervals taken across replications. Table 7 gives the unique contribution of each single year given the others, and Table 8 the nested suffix blocks $k..5$ together with the increments $\Delta(k)$.

Random-donor block permutation produces a splice artefact. Replacing years 2 to 5 of generator A with another firm's years lowers AUC by 0.514 ± 0.011, more than replacing the entire firm (0.483 ± 0.005; paired $p < 0.001$), which takes AUC below chance; the same occurs for years 3 to 5 of generator B (0.511 ± 0.018 against 0.479 ± 0.008, $p < 0.001$). A loss larger than the removal of all information can arise only if the spliced trajectories actively mislead the model, through the discontinuity created where an unrelated firm's years are appended to the target firm's earlier years. Matched donors remove the artefact (years 2 to 5 of generator A: 0.463 ± 0.020) and lower the block importance by 0.051 to 0.135 AUC on generator A and by up to 0.103 on generator B. The block totals reported in the previous version, including the 0.487 drop for years 3 to 5, were obtained with random donors and are withdrawn.

Table 7. Single-year temporal importance, AUC drop (LSTM, 20 replications, mean ± *e*)

Year	Random Donor	Matched Donor	Drop and Refit
Generator A (base AUC 0.986)			
1	0.007 ± 0.002	0.006 ± 0.002	0.005 ± 0.003
2	0.000 ± 0.001	0.001 ± 0.001	0.001 ± 0.002
3	0.006 ± 0.002	0.004 ± 0.002	−0.001 ± 0.002
4	0.045 ± 0.009	0.025 ± 0.006	0.007 ± 0.004
5	0.128 ± 0.022	0.070 ± 0.012	0.028 ± 0.006
Generator B (base AUC 0.979)			
1	0.005 ± 0.003	0.004 ± 0.002	0.002 ± 0.004
2	0.012 ± 0.005	0.008 ± 0.004	0.006 ± 0.005
3	0.009 ± 0.005	0.005 ± 0.003	0.001 ± 0.004
4	0.009 ± 0.004	0.005 ± 0.003	0.000 ± 0.005
5	0.282 ± 0.015	0.197 ± 0.016	0.148 ± 0.028

Note: AUC = Area Under the Curve; LSTM = long short-term memory.

Table 8. Suffix-block temporal importance and increments, AUC drop (LSTM, 20 replications, mean ± *e*)

k	*I*(*k*..5) Random	*I*(*k*..5) Matched	Δ(*k*) Matched	Δ(*k*) Refit
Generator A				
1	0.483 ± 0.005	=random	+0.020 ± 0.019	—
2	0.514 ± 0.011	0.463 ± 0.020	+0.081 ± 0.026	+0.139 ± 0.019
3	0.506 ± 0.030	0.383 ± 0.024	+0.146 ± 0.023	+0.098 ± 0.019
4	0.372 ± 0.045	0.237 ± 0.031	+0.167 ± 0.022	+0.063 ± 0.010
5	0.128 ± 0.022	0.070 ± 0.012	+0.070 ± 0.012	+0.028 ± 0.006
Generator B				
1	0.479 ± 0.008	=random	+0.001 ± 0.009	—
2	0.475 ± 0.011	0.478 ± 0.014	−0.005 ± 0.015	+0.078 ± 0.029
3	0.511 ± 0.018	0.483 ± 0.016	+0.097 ± 0.019	+0.118 ± 0.030
4	0.488 ± 0.021	0.386 ± 0.017	+0.190 ± 0.021	+0.117 ± 0.035
5	0.283 ± 0.016	0.195 ± 0.018	+0.195 ± 0.018	+0.148 ± 0.028

Notes: *e* is the half-width of the 95% confidence interval across the 20 replications, each averaging 30 permutations. Block $k = 1$ replaces the whole firm and has no splice boundary; its increment uses the matched block 2..5. Refit is undefined for $k = 1$, which leaves no input.
AUC = Area Under the Curve; LSTM = long short-term memory.

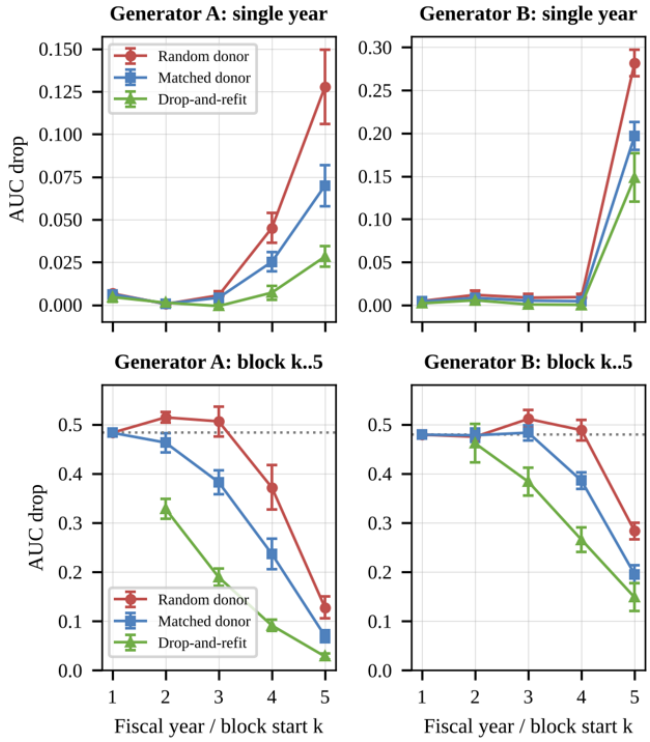

Figure 4. Temporal attribution by random-donor permutation, matched-donor permutation and drop-and-refit, for single years (top) and suffix blocks (bottom); the dotted line marks replacement of the whole firm

Because every suffix block contains year 5, a block total says nothing on its own about the earlier years inside it; the increments $\Delta(k)$ do. On generator A the matched-donor increments for years 2, 3 and 4 are 0.081, 0.146 and 0.167, and the refit increments 0.139, 0.098 and 0.063, all significantly positive ($p < 0.001$). Early years therefore carry information when the later years are absent: an LSTM refitted on years 1 and 2 alone loses 0.189 AUC, not the 0.483 that complete removal would cost. Generator B provides a check with a known answer. Onset never occurs before year 2, so year 1 holds no information, and an LSTM refitted on year 1 alone loses 0.462 AUC, which is at chance.

Single-year attribution answers a different question, the unique contribution of a year given all the others. Here the three estimators agree on the ranking, with year 5 first and year 4 second on generator A and year 5 alone on generator B, but not on magnitude (Table 7). For year 5, random-donor permutation gives 0.128 on generator A and 0.282 on generator B, matched-donor permutation 0.070 and 0.197, and drop-and-refit 0.028 and 0.148. Random-donor permutation thus overstates the unique contribution of the terminal year by a factor of 4.5 on generator A and 1.9 on generator B relative to refitting, which is the extrapolation effect described in the study [28]. Under refitting no single year among years 1 to 4 contributes more than 0.007 AUC on either generator: early-year information is real but redundant with the later years, which is why it can be removed one year at a time at almost no cost and still carries substantial signal as a block.

Drop-and-refit also compares how the two multi-year models use time. LR-flat loses almost exactly what the LSTM loses when the same years are withheld: across the nine refit conditions on each generator the difference never exceeds 0.014 AUC, and 1 of the 18 differences reach nominal significance at 0.05 without correction for multiple testing. The recurrent model does not extract temporal information that the linear model on concatenated years misses.

Feature-level attribution, computed by permuting whole firm trajectories within each replication, matches the channels each generator manipulates. On generator A the largest drops are debt-to-equity (0.072 ± 0.009), current ratio (0.037 ± 0.006) and net income margin (0.036 ± 0.008), with receivables turnover (0.019 ± 0.005) and asset turnover (0.010 ± 0.003) smaller. On generator B, which manipulates only margin, receivables turnover and leverage, net income margin (0.103 ± 0.015), receivables turnover (0.048 ± 0.008) and debt-to-equity (0.036 ± 0.007) carry the signal, while current ratio (−0.001 ± 0.002) and asset turnover (0.000 ± 0.002) contribute nothing. This serves as a check that the models learn the intended signal.

5.5 Interpretation

The three experiments converge. Multi-year information improves detection relative to a single-period model, most clearly when manipulation begins abruptly, and this holds across generators, sample sizes and architectures. The recurrent mechanism, however, contributes nothing detectable beyond what non-recurrent models extract from the same years supplied as separate features: a linear model matches its ranking performance on every generator, a feed-forward network matches its threshold-dependent performance where the linear model falls short, and both multi-year models lose the same performance when years are withheld. For the generating processes examined, the practical implication is that a concatenated-feature model is preferable on grounds of simplicity, training cost and interpretability, and that recurrent architectures should be justified by evidence against a matched multi-year control rather than against a single-period baseline.

6. CONCLUSIONS

This study separated two claims usually merged in the fraud detection literature. Multi-year information helps: the LSTM significantly exceeded terminal-year logistic regression on two of three generators, most strongly under abrupt onset, and increasingly with sample size. Learned recurrence does not help: across three generators, six recurrent configurations and five training sizes, the LSTM never significantly exceeded logistic regression on concatenated five-year features in ranking performance, and where it exceeded it on threshold-dependent metrics a non-recurrent feed-forward network on the same inputs did equally well. A methodological result accompanies these: temporal attribution depends on how withheld years are replaced. Random-donor permutation overstates the unique importance of the terminal year, and splicing year blocks from unrelated firms can push performance below chance; matched-donor permutation and drop-and-refit avoid both effects, and block importance must be read through increments because nested blocks all contain the final year.

Two conclusions from the previous version are revised. The claim that the terminal state is approximately sufficient is withdrawn, since it was based on a comparison lacking power and is contradicted by replication at a larger sample. The claim that the sample-size explanation was falsified is also withdrawn as stronger than the evidence supports; the weaker statement, that these data do not support it over the range

tested, is retained. The random-donor block-permutation figures used in the previous version to argue for the importance of early years are also withdrawn, since they were inflated by splice boundaries and confounded by the presence of year 5 in every block.

The constructive implication is a design requirement rather than a recommendation for or against recurrence. Studies proposing sequence models for fraud detection should report a non-recurrent multi-year control, because without it the contribution of input scope cannot be separated from that of architecture, and the evidence here indicates that input scope accounts for the observed gains in these regimes.

The scope of these conclusions is limited in ways that should be stated plainly. All evidence comes from three synthetic generators with five ratios and complete five-year histories, so the results describe those processes and not archival enforcement data; absolute performance levels are artifacts of generators whose signals are cleaner and more homogeneous than real manipulation, and no statement is made about auditor workflows, regulatory cost trade-offs or behaviour under real distribution shift. The feature set does not reproduce established constructs such as the M-Score or F-Score, nor accrual-quality measures. The architecture sweep covers one family and cannot exclude attention-based or substantially larger models, and the sample sizes examined reach 8,000 firms with limited replication at the upper end.

Future work should test whether the conclusion survives on archival panels, where manipulation is heterogeneous and reporting histories irregular; construct generators in which ordering information cannot be recovered from any fixed-length concatenation, which is the regime where recurrence should show an advantage if it has one; and extend the attribution comparison to model-based conditional samplers and Shapley-type methods that respect the dependence structure of panel data.

REFERENCES

[1] Beneish, M.D. (1999). The detection of earnings manipulation. Financial Analysts Journal, 55(5): 24-36. https://doi.org/10.2469/faj.v55.n5.2296

[2] Dechow, P.M., Ge, W., Larson, C.R., Sloan, R.G. (2011). Predicting material accounting misstatements. Contemporary Accounting Research, 28(1): 17-82. https://doi.org/10.1111/j.1911-3846.2010.01041.x

[3] Ravisankar, P., Ravi, V., Rao, G.R., Bose, I. (2011). Detection of financial statement fraud and feature selection using data mining techniques. Decision Support Systems, 50(2): 491-500. https://doi.org/10.1016/j.dss.2010.11.006

[4] Cecchini, M., Aytug, H., Koehler, G.J., Pathak, P. (2010). Detecting management fraud in public companies. Management Science, 56(7): 1146-1160. https://doi.org/10.1287/mnsc.1100.1174

[5] Dawood, R.N. (2026). Federated deep learning for intrusion detection in Internet of Things networks using software-defined networking. Ingénierie des Systèmes d'Information, 31(2): 625-637. https://doi.org/10.18280/isi.310229

[6] Hajek, P., Henriques, R. (2017). Mining corporate annual reports for intelligent detection of financial statement fraud – A comparative study of machine learning methods. Knowledge-Based Systems, 128: 139-152. https://doi.org/10.1016/j.knosys.2017.05.001

[7] Bao, Y., Ke, B., Li, B., Yu, Y.J., Zhang, J. (2020). Detecting accounting fraud in publicly traded U.S. firms using a machine learning approach. Journal of Accounting Research, 58(1): 199-235. https://doi.org/10.1111/1475-679X.12292

[8] Perols, J. (2011). Financial statement fraud detection: An analysis of statistical and machine learning algorithms. Auditing: A Journal of Practice & Theory, 30(2): 19-50. https://doi.org/10.2308/ajpt-50009

[9] Craja, P., Kim, A., Lessmann, S. (2020). Deep learning for detecting financial statement fraud. Decision Support Systems, 139: 113421. https://doi.org/10.1016/j.dss.2020.113421

[10] Purda, L., Skillicorn, D. (2015). Accounting variables, deception, and a bag of words: Assessing the tools of fraud detection. Contemporary Accounting Research, 32(3): 1193-1223. https://doi.org/10.1111/1911-3846.12089

[11] Kim, A., Yang, Y., Lessmann, S., Ma, T., Sung, M.C., Johnson, J.E.V. (2020). Can deep learning predict risky retail investors? A case study in financial risk behavior forecasting. European Journal of Operational Research, 283(1): 217-234. https://doi.org/10.1016/j.ejor.2019.11.007

[12] Hochreiter, S., Schmidhuber, J. (1997). Long short-term memory. Neural Computation, 9(8): 1735-1780. https://doi.org/10.1162/neco.1997.9.8.1735

[13] Cho, K., van Merrienboer, B., Gulcehre, C., et al. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. In Conference on Empirical Methods in Natural Language Processing, Doha, pp. 1724-1734. https://doi.org/10.3115/v1/D14-1179

[14] Vaswani, A., Shazeer, N., Parmar, N., et al. (2017). Attention is all you need. In Advances in Neural Information Processing Systems, Long Beach, pp. 5998-6008.

[15] Ashtiani, M.N., Raahemi, B. (2022). Intelligent fraud detection in financial statements using machine learning and data mining: A systematic literature review. IEEE Access, 10: 72504-72525. https://doi.org/10.1109/ACCESS.2021.3096799

[16] He, H., Garcia, E.A. (2009). Learning from imbalanced data. IEEE Transactions on Knowledge and Data Engineering, 21(9): 1263-1284. https://doi.org/10.1109/TKDE.2008.239

[17] Japkowicz, N., Stephen, S. (2002). The class imbalance problem: A systematic study. Intelligent Data Analysis, 6(5): 429-449. https://doi.org/10.3233/IDA-2002-6504

[18] Fernandez, A., Garcia, S., Galar, M., Prati, R.C., Krawczyk, B., Herrera, F. (2018). Learning from imbalanced data sets. Cham: Springer. https://doi.org/10.1007/978-3-319-98074-4

[19] Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P. (2002). SMOTE: Synthetic minority over-sampling technique. Journal of Artificial Intelligence Research, 16: 321-357. https://doi.org/10.1613/jair.953

[20] Zhao, Z., Bai, T. (2022). Financial fraud detection and prediction in listed companies using SMOTE and machine learning algorithms. Entropy, 24(8): 1157. https://doi.org/10.3390/e24081157

[21] Davis, J., Goadrich, M. (2006). The relationship between precision-recall and ROC curves. In International

Conference on Machine Learning, Pittsburgh, USA, pp. 233-240. https://doi.org/10.1145/1143844.1143874
[22] Saito, T., Rehmsmeier, M. (2015). The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. PLOS ONE, 10(3): e0118432. https://doi.org/10.1371/journal.pone.0118432
[23] Chicco, D., Jurman, G. (2020). The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. BMC Genomics, 21(1): 6. https://doi.org/10.1186/s12864-019-6413-7
[24] Demsar, J. (2006). Statistical comparisons of classifiers over multiple data sets. Journal of Machine Learning Research, 7: 1-30.
[25] Dawood, R.N., Fawzi, N.A., Hassan, H.A.A. (2025). Multi cyber attacks detection system in internet of things based on machine learning. Journal Européen des Systèmes Automatisés, 58(6): 1179-1187. https://doi.org/10.18280/jesa.580608
[26] Lundberg, S.M., Lee, S.I. (2017). A unified approach to interpreting model predictions. In Advances in Neural Information Processing Systems, Long Beach, pp. 4765-4774.
[27] Rudin, C. (2019). Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. Nature Machine Intelligence, 1(5): 206-215. https://doi.org/10.1038/s42256-019-0048-x
[28] Hooker, G., Mentch, L., Zhou, S. (2021). Unrestricted permutation forces extrapolation: Variable importance requires at least one more model, or there is no free variable importance. Statistics and Computing, 31(6): 82. https://doi.org/10.1007/s11222-021-10057-z
[29] Hochreiter, S. (1991). Untersuchungen zu dynamischen neuronalen Netzen (Diploma thesis). Technische Universität München, Munich, Germany.
[30] Bengio, Y., Simard, P., Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. IEEE Transactions on Neural Networks, 5(2): 157-166. https://doi.org/10.1109/72.279181
[31] Kingma, D.P., Ba, J. (2015). Adam: A method for stochastic optimization. In International Conference on Learning Representations, San Diego.
[32] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. Journal of Machine Learning Research, 15: 1929-1958.

NOMENCLATURE

b	bias vector
B	mini-batch size
c	cell state
d	number of financial ratio features
e	half-width of a 95% confidence interval
E	number of training epochs
f	forget gate activation
g	gradient of the loss
h	hidden state
H	recurrent hidden dimension
i	input gate activation; firm index
I	temporal importance, AUC drop
k	first fiscal year of a withheld block
m	first moment estimate
N	number of firms
o	output gate activation
p	dropout probability
t	fiscal year index
T	sequence length, years
U, W	learned weight matrices
v	second moment estimate
w^+	positive class weight, dimensionless
W	Wilcoxon signed-rank statistic
x	input feature vector
X	firm sequence tensor
y	binary fraud label
z	standardized feature

Greek symbols

β	moment decay rate, dimensionless
Δ	paired mean difference between models
ε	numerical stability constant
η	learning rate
θ	model parameters
μ	training-partition feature mean
σ	standard deviation; logistic sigmoid
τ	decision threshold, dimensionless
φ	nonlinear activation function

Subscripts and superscripts

i	firm index
t	time step
$+$	positive (fraud) class
$-$	negative (non-fraud) class

APPENDIX

Table A1. Paired comparisons on every metric, all generators (20 seeds; Δ = first model minus second; lower Brier is better)

Comparison	Metric	Δ ± *e*	*W*	*p*
Generator A				
LSTM − LR-last	AUC-ROC	+0.010 ± 0.003	3.0	<0.001
	PR-AUC	+0.044 ± 0.016	10.0	<0.001
	F1	+0.055 ± 0.024	6.0	<0.001
	MCC	+0.057 ± 0.025	10.0	<0.001
	Brier	−0.020 ± 0.006	6.0	<0.001
	Specificity	+0.003 ± 0.008	60.0	0.434
	Bal. accuracy	+0.036 ± 0.018	16.0	<0.001
LSTM − LR-flat	AUC-ROC	+0.000 ± 0.002	95.0	0.709
	PR-AUC	+0.003 ± 0.011	98.0	0.812
	F1	+0.005 ± 0.027	93.0	0.936
	MCC	+0.006 ± 0.027	88.0	0.778
	Brier	−0.001 ± 0.005	85.0	0.475
	Specificity	+0.001 ± 0.008	78.5	0.760
	Bal. accuracy	−0.001 ± 0.015	91.0	0.872
LR-flat − LR-last	AUC-ROC	+0.009 ± 0.003	2.0	<0.001
	PR-AUC	+0.041 ± 0.014	4.0	<0.001
	F1	+0.051 ± 0.024	22.0	0.001
	MCC	+0.051 ± 0.026	24.0	0.001
	Brier	−0.018 ± 0.004	0.0	<0.001
	Specificity	+0.002 ± 0.007	78.0	0.744
	Bal. accuracy	+0.037 ± 0.017	16.0	<0.001
LSTM − MLP-flat	AUC-ROC	+0.002 ± 0.003	65.0	0.143
	PR-AUC	−0.002 ± 0.009	95.0	0.729
	F1	−0.003 ± 0.023	90.0	0.841

	MCC	−0.004 ± 0.024	86.0	0.717
	Brier	+0.008 ± 0.005	20.0	<0.001
	Specificity	−0.004 ± 0.005	50.5	0.218
	Bal. accuracy	+0.007 ± 0.014	72.5	0.365
Generator B				
LSTM − LR-last	AUC-ROC	+0.020 ± 0.010	10.0	<0.001
	PR-AUC	+0.084 ± 0.025	4.0	<0.001
	F1	+0.070 ± 0.038	27.0	0.002
	MCC	+0.075 ± 0.040	25.0	0.002
	Brier	−0.037 ± 0.004	0.0	<0.001
	Specificity	+0.001 ± 0.005	64.5	0.570
	Bal. accuracy	+0.046 ± 0.026	23.5	0.002
LSTM − LR-flat	AUC-ROC	+0.000 ± 0.005	81.0	0.388
	PR-AUC	+0.000 ± 0.017	84.0	0.452
	F1	+0.000 ± 0.032	81.0	0.845
	MCC	+0.003 ± 0.034	81.0	0.845
	Brier	−0.005 ± 0.005	50.0	0.040
	Specificity	+0.001 ± 0.005	65.5	0.602
	Bal. accuracy	−0.011 ± 0.023	54.0	0.170
LR-flat − LR-last	AUC-ROC	+0.020 ± 0.009	5.0	<0.001
	PR-AUC	+0.084 ± 0.024	2.0	<0.001
	F1	+0.070 ± 0.034	19.0	<0.001
	MCC	+0.072 ± 0.036	20.0	<0.001
	Brier	−0.033 ± 0.004	0.0	<0.001
	Specificity	+0.000 ± 0.004	75.5	0.962
	Bal. accuracy	+0.056 ± 0.018	5.0	<0.001
LSTM − MLP-flat	AUC-ROC	+0.010 ± 0.009	46.0	0.027
	PR-AUC	+0.014 ± 0.023	67.0	0.165
	F1	+0.012 ± 0.042	69.0	0.472
	MCC	+0.013 ± 0.044	75.0	0.421
	Brier	+0.002 ± 0.004	90.0	0.596
	Specificity	−0.001 ± 0.005	79.5	0.793
	Bal. accuracy	+0.013 ± 0.020	59.0	0.147

Generator C				
LSTM − LR-last	AUC-ROC	−0.005 ± 0.009	71.0	0.216
	PR-AUC	−0.010 ± 0.010	55.0	0.064
	F1	+0.013 ± 0.017	49.5	0.117
	MCC	+0.016 ± 0.018	47.5	0.098
	Brier	−0.012 ± 0.004	5.0	<0.001
	Specificity	+0.004 ± 0.005	36.5	0.103
	Bal. accuracy	−0.004 ± 0.012	71.5	0.542
LSTM − LR-flat	AUC-ROC	−0.001 ± 0.008	101.0	0.898
	PR-AUC	+0.003 ± 0.011	96.0	0.756
	F1	+0.036 ± 0.019	21.0	0.003
	MCC	+0.037 ± 0.021	21.0	0.003
	Brier	−0.012 ± 0.003	2.0	<0.001
	Specificity	+0.004 ± 0.004	39.0	0.074
	Bal. accuracy	+0.015 ± 0.013	35.5	0.017
LR-flat − LR-last	AUC-ROC	−0.004 ± 0.008	76.0	0.294
	PR-AUC	−0.012 ± 0.011	48.0	0.033
	F1	−0.022 ± 0.022	35.0	0.049
	MCC	−0.022 ± 0.026	42.0	0.102
	Brier	+0.000 ± 0.002	102.0	0.927
	Specificity	+0.001 ± 0.006	50.0	0.568
	Bal. accuracy	−0.019 ± 0.013	15.5	0.004
LSTM − MLP-flat	AUC-ROC	+0.002 ± 0.008	83.0	0.430
	PR-AUC	+0.002 ± 0.010	98.0	0.812
	F1	+0.005 ± 0.018	76.0	0.445
	MCC	+0.007 ± 0.020	77.0	0.469
	Brier	+0.005 ± 0.004	37.0	0.009
	Specificity	+0.003 ± 0.003	44.5	0.129
	Bal. accuracy	−0.006 ± 0.009	72.0	0.355