Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Do Sequence Deep Learning Models Help Healthcare Provider Fraud Detection? A Two-Dataset, Multi-Split Benchmark of an Attention-Enhanced CNN-RNN against Tree Ensembles and Tabular Deep Learning

Zia Ullah

zia.ullah@valueadd.se

City University of Science and Information Technology

Wahab Khan

City University of Science and Information Technology

Research Article

Keywords: Healthcare fraud detection, Tabular data, Attention mechanism, Gradient boosting, Label leakage, Benchmark

Posted Date: November 16th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11365365/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Abstract

Attention-based sequence models are increasingly proposed for healthcare fraud detection but are rarely compared with strong tabular baselines. We test whether an attention-enhanced CNN-RNN helps on Medicare provider data and whether its attention weights indicate feature importance. We compared nine models, from an LR-RF hybrid and gradient-boosted trees to an MLP, FT-Transformer, CNN-RNN and attention CNN-RNN, over repeated random splits with natural-distribution test sets and validation-tuned thresholds. Dataset A has 100,000 Medicare provider records labelled by a statistical rule on submitted charges; we evaluated it with and without the label-defining charge features. Dataset B has 5,410 providers with fraud labels supplied with the data. We also retrained the attention model with shuffled feature order. On dataset A, gradient-boosted trees achieved a fraud F1 of 0.952. This is much better than the best neural model (0.769) and the attention CNN-RNN (0.750). When charge features were removed, every model's performance dropped sharply, with scores falling between 0.42 and 0.52. On dataset B, all nine models scored between 0.588 and 0.629. This variation is similar to what is seen between train-test splits. Attention never reliably improved the CNN-RNN, and its weights followed input position rather than features. The large advantage of tree ensembles on rule-labelled data reflects reconstruction of the labelling rule; with labels supplied with the data, architecture choice made little difference. Sequence-based models and attention did not provide reliable improvements on either dataset. Also, attention weights should not be considered as explanations for tabular fraud models.

1 Introduction

Fraud in public health insurance diverts money from patient care, and the volume of claims in programmes such as Medicare makes manual review impractical [1, 2]. Machine learning is now the standard tool for flagging suspicious providers, and recent reviews document a steady move from logistic regression and random forests towards deep architectures [3–5]. Convolutional and recurrent networks, attention layers, transformers and graph neural networks have all been proposed for the task [6–9], usually with high reported accuracy and a new architecture as the main contribution.

Earlier fraud detection studies have mainly explored deep learning, but they leave three questions open. First, it is not always clear whether these models perform better than traditional machine learning methods. LightGBM and XGBoost are important comparisons that should not be overlooked. Provider records are tabular: each is a fixed set of heterogeneous columns with no natural order. Large benchmarks outside the fraud domain have repeatedly found gradient-boosted trees to be at least as accurate as deep learning on such data [10–13], so a deep fraud model compared only with logistic regression or a random forest has not met its obvious competitor. The second is what the attention weights of such a model mean. Attention is often presented as built-in interpretability, but work in natural language processing has shown that attention weights need not identify the inputs that drive a prediction [14]. The third is specific to many public fraud datasets: the labels are not confirmed cases but are derived from a rule applied to the same features the model sees. A strong model can then score well by rediscovering the rule, a form of leakage that has inflated results across many fields [15].

This paper reports a benchmark built around these questions. It began as an attempt to show that adding Bahdanau-style attention after the GRU of a CNN-RNN improves Medicare provider fraud detection. When we added gradient boosting and a tabular transformer, repeated the experiments over several random splits and removed the label-defining features, that claim did not hold, and we report what we found instead. The contributions are as follows. We compare nine models under one protocol with natural-distribution test sets, validation-tuned thresholds and repeated splits, and we test the differences statistically. We measure how much of the apparent performance comes from the labelling rule, by repeating the comparison without the charge features and on a second public dataset whose labels are supplied with the data rather than derived from the features. We test whether the attention weights of the CNN-RNN track feature relevance, by shuffling the order in which features are presented. The full protocol is given in Section 3 so that the study can be reproduced.

2 Related Work

2.1 Machine learning for healthcare provider fraud

Early work on Medicare data used both unsupervised and supervised learners. Bauder et al. [16] applied outlier detection to Medicare Part B data and showed that statistically unusual providers could be surfaced without labels, and Bauder and Khoshgoftaar [17] studied how severe class imbalance affects supervised learners. A large body of work from the same group labels providers using the List of Excluded Individuals and Entities (LEIE), matched by National Provider Identifier: Johnson and Khoshgoftaar [6] trained neural networks on such data, and Hancock et al. [18, 19] showed that ensemble feature selection and random undersampling can improve and simplify LEIE-labelled models. Ensemble learners remain strong baselines: random forests and gradient boosting perform well on health insurance claims [20–22], and resampling methods such as SMOTE-ENN are commonly added for imbalanced Medicare data [23, 24]. Xiao et al. [25] combined a probabilistic model with interpretable machine learning for fraud prediction, and Fourkiotis and Tsadiras [26] reported that careful preprocessing and evaluation mattered more than the choice of classifier, with an MLP best on their composite index and CatBoost giving the fewest false positives. Shekhar et al. [27] took an unsupervised approach to Medicare hospital claims and validated it against hospitals named in Department of Justice fraud cases, one of the few studies to test against confirmed outcomes.

2.2 Deep and attention-based models

Convolutional and recurrent networks were developed for images and sequences [28, 29], and gated recurrent units offer a lighter alternative to the LSTM [30]. Attention was introduced to let a recurrent model weight its hidden states when producing an output [31], and transformers later built entire architectures from it [32]. In fraud detection, Sakil et al. [7] proposed a CNN-Transformer-XGBoost framework for Medicare fraud, Mardani and Moradi [8] applied graph attention networks to provider fraud, and Muhammad et al. [9] modelled medical claims as heterogeneous graphs with explanation modules. When a one-dimensional convolution or a recurrent layer is applied to a single tabular record,

however, the features are treated as a sequence, and the model sees an order that has no meaning in the data. Architectures designed for tabular data avoid this: FT-Transformer embeds each feature as a token and lets self-attention relate them without imposing an order [33], and tabular foundation models have recently been shown to rival tuned tree ensembles on small datasets [34].

2.3 Tabular benchmarks, leakage and attention as explanation

Shwartz-Ziv and Armon [10] compared deep tabular models with XGBoost [35] across many datasets and found that XGBoost usually performed better and needed far less tuning. Grinsztajn et al. [11] came to the same conclusion using a larger benchmark. They said part of the difference comes from the non-smooth functions that are common in tabular data. They also pointed to uninformative features as a factor. McElfresh et al. [12] found that the best method depends on the dataset. Still, they said that tuned boosting is usually close behind the top methods. Borisov et al. [13] reached similar results in their survey. Kapoor and Narayanan [15] documented how leakage, including features that encode the label, has inflated published results, and Bouthillier et al. [36] showed that a single train-test split can make random variation look like a method improvement. On attention, Jain and Wallace [14] showed that attention weights often disagree with other measures of feature importance, Wiegreffe and Pinter [37] argued that attention can still be informative under some conditions, and Bibal et al. [38] concluded that whether attention explains a model must be tested case by case. Our shuffling experiment is one such test.

3 Materials and Methods

3.1 Dataset A: rule-labelled provider records

We used the public Healthcare Providers dataset [39], which contains 100,000 Medicare provider records with 27 columns covering provider type and location, procedure codes, service and beneficiary counts, and average submitted charge, allowed and payment amounts. The dataset contains no fraud labels. Following the outlier-based view of provider fraud used in earlier Medicare work [16], we grouped providers by Provider Type (90 specialties) and labelled a provider as potentially fraudulent if its Average Submitted Charge Amount exceeded the specialty mean by more than two standard deviations. The statistics were computed on the raw charge values before any imputation. This produced 2,991 positive records (2.99%).

Name, street address, city and ZIP code columns were removed. Numeric columns stored as text were converted to numbers, and the remaining categorical columns were label-encoded. Four ratio features were derived: the payment-to-charge ratio (PCR), the charge-to-allowed ratio, services per beneficiary and payment per beneficiary. After median imputation, any column with an absolute correlation above 0.85 with an earlier column was dropped; this step uses no label information and removed the National Provider Identifier (a numeric identifier), the number of distinct beneficiary per day services, the average

Medicare payment and standardised amounts, and payment per beneficiary. A row-index column present in the source file was excluded. The final feature set had 19 columns (Table 1).

Table 1
Feature sets used in the two experimental settings

Setting	Features	Count
All features	Provider attributes (middle initial, credentials, gender, entity type, state, country, provider type, Medicare participation, place of service), procedure attributes (HCPCS code, description, drug indicator), utilisation (number of services, number of beneficiaries), amounts (average allowed amount, average submitted charge amount) and ratios (PCR, charge-to-allowed ratio, services per beneficiary)	19
Charge features removed	As above, without average submitted charge amount, PCR and charge-to-allowed ratio	16

3.2 Dataset B: providers with supplied labels

To test whether the conclusions depend on rule-derived labels, we used a second public dataset, the Healthcare Provider Fraud Detection Analysis data [40]. Its training portion contains inpatient and outpatient Medicare claims and beneficiary records for 5,410 providers, each with a provider-level label (PotentialFraud); 506 providers (9.35%) are labelled as potentially fraudulent. The labels are supplied with the data and are not computed from the variables we use, although their origin is not documented by the dataset creators. The unlabelled test portion was not used. We combined the claims into a row for each provider and created twenty different features. These features include the number of claims and beneficiaries, claims per beneficiary, the portion of inpatient claims, the average, total and highest reimbursed amount, the average deductible, the average claim and admission length, the average count of diagnosis and procedure codes, the count of unique attending physicians, the portion of claims involving an operating or other physician, and the average age, mortality, number of chronic conditions, presence of renal disease and gender of the provider's beneficiaries. Dataset B is smaller, so each test split contains about 100 fraudulent providers.

3.3 Experimental settings

Because the label is a deterministic function of the submitted charge and the provider type, and both are model inputs, the task in its usual form can be solved by learning the labelling rule. We therefore ran every comparison on dataset A in two settings, and once on dataset B. In the all-features setting, models see the same inputs as in typical use of this dataset. In the charge-removed setting, the submitted charge and the two ratios computed from it are dropped, so a model must infer unusually high charges from the remaining billing and utilisation variables. Neither setting measures detection of confirmed fraud. The first measures how well each model reconstructs the rule, and the second how well the rule can be predicted from indirect evidence.

3.4 Evaluation protocol

Each run used a stratified 80/20 train-test split. A further 15% of the training portion was held out at the natural class distribution as a validation set. Features were standardised using statistics from the remaining training data, and SMOTE [41] was applied to that training data only, so the validation and test sets kept the natural fraud rate (about 3% for dataset A and 9% for dataset B). For every model, the decision threshold was chosen on the validation set by maximising fraud-class F1 and then applied unchanged to the test set. The search grid for the LR-RF hybrid, tree models and MLPs ranged from 0.01 to 0.99 in steps of 0.01. For the CNN-RNN, attention CNN-RNN and FT-Transformer, the grid was narrower, from 0.10 to 0.79. Every threshold selected for these three models fell strictly within this range, between 0.42 and 0.75, so the narrower grid did not limit their choices. We report fraud-class precision, recall and F1 scores. We also report the area under the ROC curve and average precision, which is the area under the precision-recall curve. Because positives are rare, fraud F1 and average precision are the primary metrics. Each configuration was repeated over five independent random splits (seeds 1, 2, 3, 4 and 42), and every model was trained and tested on the same five splits, so that models can be compared split by split. Results are reported as mean ± standard deviation across splits. To compare two models we computed the per-split difference in F1 and its 95% confidence interval from the t distribution over the splits shared by both models, following the recommendation to account for split-to-split variance [36].

3.5 Models

The LR-RF hybrid and the two CNN-RNN models follow the architectures of our earlier, not yet published work on this dataset; here they are retrained and re-evaluated under the common protocol above, and none of the results below appear elsewhere.

LR-RF hybrid. We combined logistic regression and random forest by averaging the probabilities predicted by the two models. Logistic regression used the liblinear solver with C set to 1. The random forest had 100 trees, with the maximum depth set to 10. We used balanced class weights for both models.

Gradient-boosted trees. We tested XGBoost and LightGBM. The first XGBoost model used 400 trees, a maximum depth of 6, a learning rate of 0.1, and row and column subsampling rates of 0.9 [35]. We also tested a tuned XGBoost model. For XGBoost, we ran a random search with 20 different parameter combinations. These included changes to the number of trees, tree depth, learning rate, subsampling, and minimum child weight. We kept the configuration that achieved the highest average precision on the validation set. LightGBM was run with 400 trees, 63 leaves, and a learning rate of 0.05 [42].

Multilayer perceptron (MLP). The default MLP had three dense layers with 128, 64, and 32 units. We used ReLU activation and a dropout rate of 0.3. We also tried a tuned version, testing eight configurations with different layer widths (64–512 units), depths (two to four layers), dropout rates, learning rates, and batch sizes. The configuration with the highest validation average precision was selected.

FT-Transformer. We followed the approach described by Gorishniy et al. [33]. Each feature was mapped to a 64-dimensional token using a separate linear embedding. We added a learnable [CLS] token before passing the sequence through three pre-norm Transformer blocks, each with eight attention heads. The output for the [CLS] token was then passed to a dense layer and a sigmoid output.

CNN-RNN. We treated the feature vector as a one-channel sequence with 19, 16, or 20 positions, depending on the setting. The model began with a one-dimensional convolution using 64 filters and a kernel size of 3. We applied dropout at a rate of 0.3 before passing the output to a GRU with 64 units [30]. The GRU returned its final hidden state. After another dropout layer with a rate of 0.3, the output passed through a dense layer with 32 units and a final sigmoid output.

Attention CNN-RNN. As above, but the GRU returns its hidden state at every position, and an additive self-attention layer [31] computes weights $\alpha = \mathrm{softmax}(V \tanh(W h))$ over positions and passes the weighted sum of hidden states to the dense layer. W has 64 units. The attention weights are also extracted for analysis.

The default MLP, FT-Transformer and both CNN-RNNs were trained with focal loss ($\gamma = 2$, $\alpha = 0.25$) [43], as in the original attention model; the tuned MLP used binary cross-entropy. All neural models used the Adam optimiser, up to 100 epochs (FT-Transformer: up to 30 epochs with batch size 1024, to keep its training time on a CPU manageable), and early stopping on a 10% slice of the resampled training data (patience 8, best weights restored). They were implemented in TensorFlow 2.21; the tree models used scikit-learn, XGBoost 3.4 and LightGBM 4.7. All experiments ran on a CPU.

3.6 Additional analyses

To test whether attention weights reflect feature relevance or input position, we retrained the CNN-RNN and attention CNN-RNN with the feature columns in a random fixed order (a different permutation for each split) and compared the resulting attention profiles with those from the original order, both by position and by feature (Spearman rank correlation of the per-feature weights). We also trained the attention model with binary cross-entropy instead of focal loss, and evaluated every model at the default threshold of 0.5 to measure the effect of threshold tuning.

3.7 Use of AI tools

AI tools were used to support parts of the research workflow and manuscript preparation. The authors reviewed the outputs and take full responsibility for the study, its results, and the final manuscript.

4 Results

4.1 Main comparison

Table 2
Test-set performance with all features (fraud rate about 3%), mean ± standard deviation over five random splits. Precision, recall and F1 refer to the fraud class; AP is average precision; ROC AUC is the area under the ROC curve; time is mean training time in seconds on a CPU, including hyperparameter search for the tuned models

Model	Precision	Recall	F1	AP	ROC AUC	Time (s)
LR-RF hybrid	0.609 ± 0.028	0.676 ± 0.048	0.639 ± 0.015	0.649 ± 0.017	0.983 ± 0.001	13
XGBoost (default)	0.931 ± 0.015	0.944 ± 0.021	0.938 ± 0.008	0.984 ± 0.003	0.999 ± 0.000	4
XGBoost (tuned)	0.935 ± 0.012	0.945 ± 0.016	0.940 ± 0.008	0.985 ± 0.004	0.999 ± 0.000	68
LightGBM	0.951 ± 0.023	0.955 ± 0.021	0.952 ± 0.006	0.992 ± 0.002	1.000 ± 0.000	5
MLP (default)	0.781 ± 0.041	0.762 ± 0.050	0.769 ± 0.010	0.848 ± 0.017	0.990 ± 0.002	117
MLP (tuned)	0.772 ± 0.032	0.733 ± 0.020	0.752 ± 0.012	0.833 ± 0.013	0.989 ± 0.002	1006
FT-Transformer	0.759 ± 0.059	0.781 ± 0.063	0.769 ± 0.053	0.855 ± 0.056	0.993 ± 0.003	1796
CNN-RNN	0.736 ± 0.056	0.756 ± 0.039	0.744 ± 0.029	0.829 ± 0.024	0.988 ± 0.003	891
Attention CNN-RNN	0.733 ± 0.047	0.773 ± 0.039	0.750 ± 0.009	0.831 ± 0.008	0.988 ± 0.002	1081

Table 2 and Fig. 1a show the main result. The gradient-boosted trees were the best models by a wide margin: LightGBM reached a fraud F1 of 0.952 ± 0.006 and an average precision of 0.992 ± 0.002, and the three tree models were within a few points of each other. On dataset A, the tree models performed better than the neural models. LightGBM showed a clear advantage over the default MLP, with an F1-score difference of + 0.183 (95% CI + 0.167 to + 0.199). Tuning did not improve the MLP: its score dropped from 0.769 ± 0.010 to 0.752 ± 0.012. The results suggest that tuning the neural network did not close the gap between it and the tree models. The LR-RF hybrid was weakest at 0.639 ± 0.015, mainly because of low precision. Every model had a ROC AUC above 0.98 (Table 2), which shows how little that metric separates models when positives are rare; the precision-recall curves in Fig. 2 separate them clearly.

4.2 Attention versus the plain CNN-RNN

Table 3
Fraud-class F1 of the plain and attention CNN-RNN on the same random splits (all features)

Split seed	CNN-RNN	Attention CNN-RNN	Difference
1	0.769	0.743	-0.026
2	0.757	0.739	-0.019
3	0.734	0.760	+ 0.025
4	0.763	0.757	-0.006
42	0.698	0.753	+ 0.056
Mean (95% CI)	0.744	0.750	+ 0.006 (-0.036 to + 0.048)

Table 3 compares the two sequence models split by split. Attention gave the higher F1 in 2 of 5 splits, and the mean difference of + 0.006 had a 95% confidence interval that included zero (-0.036 to + 0.048). On a single split (seed 42, the split used in our original experiments), attention appeared to give a clear gain; over all splits, that gain disappeared. Neither sequence model beat the default MLP, which processes the same features without imposing an order (MLP minus attention CNN-RNN: +0.019 (95% CI -0.004 to + 0.043)).

4.3 Dataset A without the label-defining features

Table 4
Test-set performance with the submitted charge, PCR and charge-to-allowed ratio removed. Columns as in Table 2, without training time

Model	Precision	Recall	F1	AP	ROC AUC
LR-RF hybrid	0.388 ± 0.046	0.483 ± 0.086	0.423 ± 0.020	0.357 ± 0.018	0.936 ± 0.006
XGBoost (default)	0.453 ± 0.023	0.546 ± 0.056	0.493 ± 0.014	0.526 ± 0.016	0.953 ± 0.003
XGBoost (tuned)	0.461 ± 0.013	0.566 ± 0.028	0.508 ± 0.014	0.542 ± 0.011	0.955 ± 0.004
LightGBM	0.491 ± 0.018	0.545 ± 0.026	0.516 ± 0.015	0.551 ± 0.013	0.956 ± 0.004
MLP (default)	0.429 ± 0.036	0.505 ± 0.057	0.461 ± 0.010	0.483 ± 0.017	0.936 ± 0.008
MLP (tuned)	0.420 ± 0.023	0.472 ± 0.039	0.443 ± 0.015	0.465 ± 0.011	0.936 ± 0.009
FT-Transformer	0.398 ± 0.024	0.469 ± 0.031	0.430 ± 0.014	0.436 ± 0.021	0.934 ± 0.004
CNN-RNN	0.435 ± 0.066	0.439 ± 0.048	0.432 ± 0.016	0.447 ± 0.021	0.925 ± 0.006
Attention CNN-RNN	0.435 ± 0.066	0.450 ± 0.055	0.436 ± 0.022	0.465 ± 0.022	0.928 ± 0.007

Without the charge features, every model lost a large part of its performance (Table 4, Fig. 1b). Fraud F1 fell to between 0.42 and 0.52; LightGBM, for example, fell from 0.95 to 0.52. Tree ensembles remained the best family (LightGBM versus MLP (default): +0.056 (95% CI + 0.042 to + 0.069)), and the neural models were close to one another, all between 0.43 and 0.46. The size of the drop shows that much of the performance in the all-features setting comes from reconstructing the labelling rule from the charge amount and provider type.

4.4 Dataset B: labels supplied with the data

Table 5
Test-set performance on dataset B (5,410 providers, fraud rate about 9%), mean ± standard deviation over 5 random splits. Columns as in Table 2, without training time

Model	Precision	Recall	F1	AP	ROC AUC
LR-RF hybrid	0.592 ± 0.078	0.683 ± 0.072	0.629 ± 0.034	0.690 ± 0.034	0.946 ± 0.009
XGBoost (default)	0.596 ± 0.044	0.642 ± 0.078	0.614 ± 0.026	0.670 ± 0.041	0.936 ± 0.011
XGBoost (tuned)	0.602 ± 0.056	0.638 ± 0.079	0.616 ± 0.037	0.686 ± 0.045	0.939 ± 0.013
LightGBM	0.588 ± 0.056	0.685 ± 0.067	0.628 ± 0.011	0.696 ± 0.037	0.941 ± 0.010
MLP (default)	0.557 ± 0.059	0.709 ± 0.046	0.622 ± 0.045	0.713 ± 0.038	0.940 ± 0.015
MLP (tuned)	0.548 ± 0.055	0.697 ± 0.037	0.612 ± 0.038	0.689 ± 0.037	0.934 ± 0.008
FT-Transformer	0.499 ± 0.054	0.719 ± 0.054	0.588 ± 0.045	0.623 ± 0.092	0.929 ± 0.020
CNN-RNN	0.553 ± 0.071	0.705 ± 0.089	0.615 ± 0.051	0.711 ± 0.054	0.942 ± 0.013
Attention CNN-RNN	0.568 ± 0.029	0.693 ± 0.069	0.623 ± 0.040	0.695 ± 0.063	0.937 ± 0.016

Dataset B tells a different story (Table 5, Fig. 1c). On dataset B, the nine models performed quite similarly. Their F1-scores ranged from 0.588 for FT-Transformer to 0.629 for the LR-RF hybrid. The difference was small (0.041, against a typical standard deviation of 0.036 for a single model across splits), and no model showed a clear advantage over the others (best tree model minus best neural model: +0.005 (95% CI -0.031 to + 0.042); best tree model minus LR-RF hybrid: -0.001 (95% CI -0.031 to + 0.030)). Attention also failed to produce a reliable improvement in the CNN-RNN model (+ 0.008 (95% CI -0.013 to + 0.029)). With labels that are not derived from the model inputs, neither tree ensembles nor any neural architecture had a measurable advantage over the others.

4.5 What the attention weights show

With the original feature order, the attention CNN-RNN placed 50% of its weight, on average, on the last three of the 19 positions (Fig. 3a). These happen to hold the derived ratios PCR, charge-to-allowed ratio and services per beneficiary, so the profile could be read as the model attending to billing ratios. Shuffling the feature order rules this reading out. With shuffled order, the weight again concentrated on the final positions (50% on the last three, Fig. 3b), whatever features those were. In one split, the final position held Number of Services and received 47% of the weight on its own. Performance under shuffled order was similar to that under the original order (Table 6). On dataset A, then, the attention

weights favour the final positions, which is consistent with later GRU hidden states summarising more of the input, and the most heavily weighted features are whichever ones happen to be placed at the end. The Spearman correlation between the per-feature weights in the original and the shuffled order was −0.01 on average, so a feature's weight in one order says almost nothing about its weight in the other. On dataset B the profile is different but equally positional (Fig. 3c, d): the weight concentrated at the start of the sequence (44% on the first three positions in the original order), and the per-feature weights again bore little relation across orders (mean Spearman correlation 0.13). The GRU gave more weight to the start of the sequence in one dataset and to the end in the other. We do not know why this happened. However, the pattern suggests that the weights are affected by feature order. They do not necessarily show which features matter most, since the same feature could receive a different weight if it appeared in another position.

Table 6
Fraud-class F1 under alternative settings (dataset A, all features). Compare with the F1 column of Table 2

Model	Change	F1	Splits
Attention CNN-RNN	Shuffled feature order	0.743 ± 0.016	5
CNN-RNN	Shuffled feature order	0.728 ± 0.011	5
Attention CNN-RNN	Binary cross-entropy instead of focal loss	0.744 ± 0.017	5
LR-RF hybrid	Default threshold 0.5 instead of tuned	0.494 ± 0.009	5
LightGBM	Default threshold 0.5 instead of tuned	0.956 ± 0.008	5
MLP (default)	Default threshold 0.5 instead of tuned	0.715 ± 0.011	5
FT-Transformer	Default threshold 0.5 instead of tuned	0.688 ± 0.071	5
CNN-RNN	Default threshold 0.5 instead of tuned	0.723 ± 0.025	5
Attention CNN-RNN	Default threshold 0.5 instead of tuned	0.703 ± 0.047	5

4.6 Loss function and threshold tuning

Table 6 summarises the remaining checks. When we trained the attention model using binary cross-entropy instead of focal loss, the attention model achieved an F1 score of 0.744 ± 0.017, which is very close to the score the attention model obtained with focal loss. Therefore, focal loss does not appear to be the source of the attention model’s performance. Threshold tuning mattered a great deal for some models. With the default threshold of 0.5, the LR-RF hybrid dropped to 0.494 ± 0.009, because training on SMOTE-balanced data pushes its probabilities upward and produces many false positives at the natural fraud rate, whereas LightGBM changed little (0.956 ± 0.008). Comparing models at a fixed threshold of 0.5 would therefore have exaggerated the gap between the hybrid and the other models.

5 Discussion

5.1 Sequence models on tabular data

The two datasets together give a clear picture. On dataset A, the ranking agrees with the broader tabular-learning literature [10–13]. Treating a provider record as a sequence and passing it through a convolution and a GRU did not help: the order of columns carries no information, and the convolution mixes neighbouring columns that happen to be adjacent in the file. An MLP, which makes no such assumption, did at least as well, and FT-Transformer, which relates features through attention without imposing an order, did not reach the tree models either. The likely reason is the form of the target. The label is a threshold on the charge amount that depends on the provider type, a sharp, axis-aligned rule. Trees represent such rules directly, while smooth neural networks approximate them with difficulty [11]. On dataset B, where the labels are not a rule on the inputs, that advantage disappeared and all nine models performed alike. The large gaps on dataset A therefore tell us more about the labels than about the models.

For practitioners the implication is simple. A new deep architecture for provider-level fraud detection should be compared with a tuned gradient-boosting model on the same splits before any gain is claimed, and the comparison should be repeated over several splits [36]. In our own earlier experiments, a single split suggested that attention improved the CNN-RNN; with more splits that difference disappeared.

5.2 What rule-based labels can and cannot show

The fall in performance when the charge features were removed, and the disappearance of the gaps between models on dataset B, are the most important results for anyone using rule-labelled fraud data. With all features, the best model is mostly learning how the labels were made. That is not useless: it shows which models can capture a specialty-specific threshold, and the charge-removed setting shows how far other billing variables can predict an unusual charge. But neither setting measures fraud detection in the sense that matters to investigators, because no label here is a confirmed case. Studies on this and similar public datasets should state this clearly and, where possible, report a setting without the label-defining features, in line with the leakage checklist of Kapoor and Narayanan [15]. Validation against confirmed outcomes, such as LEIE exclusions [6, 19] or enforcement actions [27], is the necessary next step.

5.3 Attention is not an explanation here

Attention layers are often motivated as a source of interpretability. In this model the weights followed input position, and the shuffling experiment shows that a plausible-looking attention profile can arise for reasons unrelated to feature relevance. This supports the view that attention must be tested before it is read as an explanation [14, 37, 38]. Where explanations are needed for audit or regulatory reasons, model-agnostic attribution methods applied to a strong tabular model are a better starting point than attention weights from a sequence model applied to tabular data.

5.4 Limitations

The study has several limitations. The labels are rule-derived and are not confirmed fraud, as discussed above. Dataset B adds labels that are not derived from the features we use, but their origin is not documented, and with about 100 fraudulent providers per test split its estimates are noisy; small differences between models on dataset B should not be over-interpreted. Both datasets are public Medicare samples, and private claims data may behave differently. FT-Transformer was trained with a reduced epoch budget because of the cost of CPU training, and on dataset A with all features its results varied more across splits than those of any other model; a longer budget might raise its scores somewhat, although the gap to the tree models on dataset A is several times larger than this variation. Hyperparameter search was limited to random search over a small number of configurations for XGBoost and the MLP, and the sequence models and FT-Transformer used fixed architectures; more extensive tuning could change the ranking among neural models, although tuning is known to benefit tree ensembles as much as neural networks on tabular data [11, 12]. Finally, label encoding and the correlation filter were applied to the dataset before splitting; neither step uses the labels, so label encoding and the correlation filter are unlikely to affect the comparison.

6 Conclusion

On dataset A, LightGBM scored higher than the neural models. Dataset B was different: the nine models had very similar F1-scores, ranging from 0.588 to 0.629. Adding attention to the CNN-RNN did not produce a clear improvement. These results held across five different data splits. On dataset A, some input features are closely related to the fraud labels, which explains why LightGBM scored so high there. The models have not yet been tested on confirmed Medicare fraud cases, so their performance on real fraud cases remains unknown.

Declarations

Competing interests:

The authors declare no competing interests.

Ethics approval:

Not applicable. Before analysis, we removed provider identifiers from dataset A, including National Provider Identifiers, names, and addresses. Dataset B contains pseudonymous beneficiary identifiers in the public release. No identifiable patient data were used, and no human participants took part in the study.

Consent to participate:

Not applicable.

Consent for publication:

Not applicable.

Funding:

This study received no funding.

Author Contribution

Z.U.: conceptualization, methodology, software, formal analysis, and writing of the original draft. W.K.: supervision, validation, and review and editing of the manuscript. Both authors reviewed and approved the final manuscript.

Acknowledgements

The authors thank the Department of Computer Science, City University of Science and Information Technology, Peshawar, for supporting this research.

Data Availability

Both datasets analysed in this study are publicly available on Kaggle: "Healthcare Providers Data for Anomaly Detection" (https://www.kaggle.com/datasets/tamilsel/healthcare-providers-data) and "Healthcare Provider Fraud Detection Analysis" (https://www.kaggle.com/datasets/rohitrox/healthcare-provider-fraud-detection-analysis). The labels and features can be reproduced from them by following Section 3 of the manuscript.

Code availability:

The experiment code is available from the corresponding author on reasonable request.

References

1. National Health Care Anti-Fraud Association: The challenge of health care fraud. NHCAA, Washington, DC. https://www.nhcaa.org (2022). Accessed 15 March 2024
2. Joudaki, H., Rashidian, A., Minaei-Bidgoli, B., Mahmoodi, M., Geraili, B., Nasiri, M., Arab, M.: Using data mining to detect health care fraud and abuse: a review of literature. Glob. J. Health Sci. 7(1), 194–202 (2015). https://doi.org/10.5539/gjhs.v7n1p194

3. du Preez, A., Bhattacharya, S., Beling, P., Bowen, E.: Fraud detection in healthcare claims using machine learning: a systematic review. Artif. Intell. Med. 160, 103061 (2025). https://doi.org/10.1016/j.artmed.2024.103061
4. Razzaq, K., Shah, M.: Next-generation machine learning in healthcare fraud detection: current trends, challenges, and future research directions. Information 16(9), 730 (2025). https://doi.org/10.3390/info16090730
5. Curtis, E.D., Billion-Polak, P., Khoshgoftaar, T.M., Furht, B.: A review of distinct machine learning classifiers for healthcare fraud detection. J. Big Data 12, 238 (2025). https://doi.org/10.1186/s40537-025-01295-3
6. Johnson, J.M., Khoshgoftaar, T.M.: Medicare fraud detection using neural networks. J. Big Data 6, 63 (2019). https://doi.org/10.1186/s40537-019-0225-0
7. Sakil, M.B.H., Hasan, M.A., Mozumder, M.S.A., Hasan, M.R., Opee, S.A., Mridha, M.F., Aung, Z.: Enhancing Medicare fraud detection with a CNN-Transformer-XGBoost framework and explainable AI. IEEE Access 13, 79609–79622 (2025). https://doi.org/10.1109/ACCESS.2025.3562577
8. Mardani, S., Moradi, H.: Using graph attention networks in healthcare provider fraud detection. IEEE Access 12, 132786–132800 (2024). https://doi.org/10.1109/ACCESS.2024.3425892
9. Muhammad, R., Tbaishat, D., Nazir, A., Yacoub, S., AbdulRazek, M., Abo El-Enen, M.A., Sahlol, A.T.: Fraud detection and explanation in medical claims using GNN architectures. Sci. Rep. 15, 41734 (2025). https://doi.org/10.1038/s41598-025-22910-6
10. Shwartz-Ziv, R., Armon, A.: Tabular data: deep learning is not all you need. Inf. Fusion 81, 84–90 (2022). https://doi.org/10.1016/j.inffus.2021.11.011
11. Grinsztajn, L., Oyallon, E., Varoquaux, G.: Why do tree-based models still outperform deep learning on typical tabular data? In: Advances in Neural Information Processing Systems 35 (NeurIPS 2022), Datasets and Benchmarks Track (2022)
12. McElfresh, D., Khandagale, S., Valverde, J., Prasad C, V., Ramakrishnan, G., Goldblum, M., White, C.: When do neural nets outperform boosted trees on tabular data? In: Advances in Neural Information Processing Systems 36 (NeurIPS 2023), Datasets and Benchmarks Track (2023)
13. Borisov, V., Leemann, T., Seßler, K., Haug, J., Pawelczyk, M., Kasneci, G.: Deep neural networks and tabular data: a survey. IEEE Trans. Neural Netw. Learn. Syst. 35(6), 7499–7519 (2024). https://doi.org/10.1109/TNNLS.2022.3229161
14. Jain, S., Wallace, B.C.: Attention is not explanation. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, vol. 1, pp. 3543–3556. ACL, Minneapolis (2019). https://doi.org/10.18653/v1/N19-1357
15. Kapoor, S., Narayanan, A.: Leakage and the reproducibility crisis in machine-learning-based science. Patterns 4(9), 100804 (2023). https://doi.org/10.1016/j.patter.2023.100804
16. Bauder, R., da Rosa, R., Khoshgoftaar, T.: Identifying Medicare provider fraud with unsupervised machine learning. In: 2018 IEEE International Conference on Information Reuse and Integration (IRI),

pp. 285–292. IEEE, Salt Lake City (2018). https://doi.org/10.1109/IRI.2018.00051

17. Bauder, R.A., Khoshgoftaar, T.M.: The effects of varying class distribution on learner behavior for Medicare fraud detection with imbalanced big data. Health Inf. Sci. Syst. 6, 9 (2018). https://doi.org/10.1007/s13755-018-0051-3
18. Hancock, J.T., Bauder, R.A., Wang, H., Khoshgoftaar, T.M.: Explainable machine learning models for Medicare fraud detection. J. Big Data 10, 154 (2023). https://doi.org/10.1186/s40537-023-00821-5
19. Hancock, J.T., Wang, H., Khoshgoftaar, T.M., Liang, Q.: Data reduction techniques for highly imbalanced Medicare Big Data. J. Big Data 11, 8 (2024). https://doi.org/10.1186/s40537-023-00869-3
20. Nabrawi, E., Alanazi, A.: Fraud detection in healthcare insurance claims using machine learning. Risks 11(9), 160 (2023). https://doi.org/10.3390/risks11090160
21. Wang, Z., Chen, X., Wu, Y., Jiang, L., Lin, S., Qiu, G.: A robust and interpretable ensemble machine learning model for predicting healthcare insurance fraud. Sci. Rep. 15, 218 (2025). https://doi.org/10.1038/s41598-024-82062-x
22. Manyike, G., Paepae, T.: Enhancing healthcare insurance fraud detection using explainable ensemble machine learning on imbalanced datasets. Sci. Rep. (2026). https://doi.org/10.1038/s41598-026-66566-2
23. Bounab, R., Zarour, K., Guelib, B., Khlifa, N.: Enhancing Medicare fraud detection through machine learning: addressing class imbalance with SMOTE-ENN. IEEE Access 12, 54399–54413 (2024). https://doi.org/10.1109/ACCESS.2024.3385781
24. Farahmandazad, D., Danesh, K., Abadi, H.F.N.: Application of standard machine learning models for Medicare fraud detection with imbalanced data. Risks 13(10), 198 (2025). https://doi.org/10.3390/risks13100198
25. Xiao, F., Li, H., Wang, X., Wang, J., Chen, S.: Predictive analysis for healthcare fraud detection: integration of probabilistic model and interpretable machine learning. Inf. Sci. 719, 122499 (2025). https://doi.org/10.1016/j.ins.2025.122499
26. Fourkiotis, K.P., Tsadiras, A.: Future Internet applications in healthcare: big data-driven fraud detection with machine learning. Future Internet 17(10), 460 (2025). https://doi.org/10.3390/fi17100460
27. Shekhar, S., Leder-Luis, J., Akoglu, L.: Can machine learning target health care fraud? Evidence from Medicare hospitalizations. J. Policy Anal. Manag. 45(1), e70078 (2026). https://doi.org/10.1002/pam.70078
28. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proc. IEEE 86(11), 2278–2324 (1998). https://doi.org/10.1109/5.726791
29. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. 9(8), 1735–1780 (1997). https://doi.org/10.1162/neco.1997.9.8.1735
30. Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y.: Learning phrase representations using RNN encoder–decoder for statistical machine translation. In:

Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1724–1734. ACL, Doha (2014). https://doi.org/10.3115/v1/D14-1179

31. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. In: 3rd International Conference on Learning Representations (ICLR) (2015)
32. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: Advances in Neural Information Processing Systems 30, pp. 5998–6008 (2017)
33. Gorishniy, Y., Rubachev, I., Khrulkov, V., Babenko, A.: Revisiting deep learning models for tabular data. In: Advances in Neural Information Processing Systems 34, pp. 18932–18943 (2021)
34. Hollmann, N., Müller, S., Purucker, L., Krishnakumar, A., Körfer, M., Hoo, S.B., Schirrmeister, R.T., Hutter, F.: Accurate predictions on small data with a tabular foundation model. Nature 637(8045), 319–326 (2025). https://doi.org/10.1038/s41586-024-08328-6
35. Chen, T., Guestrin, C.: XGBoost: a scalable tree boosting system. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 785–794. ACM, New York (2016). https://doi.org/10.1145/2939672.2939785
36. Bouthillier, X., Delaunay, P., Bronzi, M., et al.: Accounting for variance in machine learning benchmarks. In: Proceedings of Machine Learning and Systems 3 (MLSys 2021), pp. 747–769 (2021)
37. Wiegreffe, S., Pinter, Y.: Attention is not not explanation. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pp. 11–20. ACL, Hong Kong (2019). https://doi.org/10.18653/v1/D19-1002
38. Bibal, A., Cardon, R., Alfter, D., Wilkens, R., Wang, X., François, T., Watrin, P.: Is attention explanation? An introduction to the debate. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 3889–3900. ACL, Dublin (2022). https://doi.org/10.18653/v1/2022.acl-long.269
39. Tamilselvan, K.: Healthcare providers data for anomaly detection. Kaggle. https://www.kaggle.com/datasets/tamilsel/healthcare-providers-data. Accessed 15 March 2024
40. Gupta, R.A.: Healthcare provider fraud detection analysis. Kaggle. https://www.kaggle.com/datasets/rohitrox/healthcare-provider-fraud-detection-analysis. Accessed 10 October 2026
41. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: SMOTE: synthetic minority over-sampling technique. J. Artif. Intell. Res. 16, 321–357 (2002). https://doi.org/10.1613/jair.953
42. Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., Liu, T.Y.: LightGBM: a highly efficient gradient boosting decision tree. In: Advances in Neural Information Processing Systems 30, pp. 3146–3154 (2017)
43. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: 2017 IEEE International Conference on Computer Vision (ICCV), pp. 2999–3007 (2017).

https://doi.org/10.1109/ICCV.2017.324

Figures

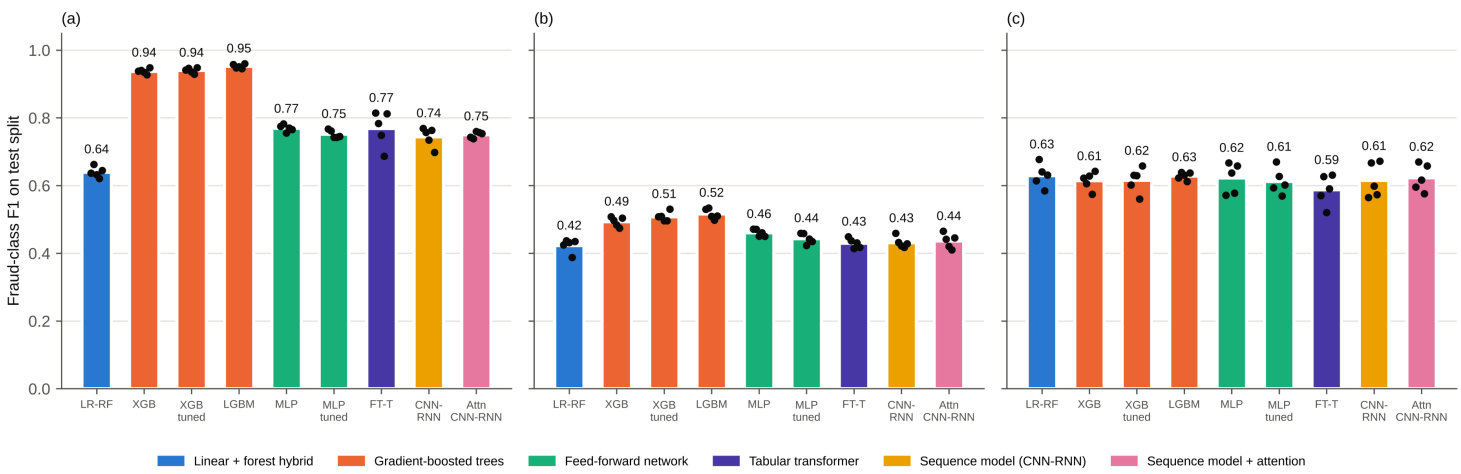


Figure 1

Fraud-class F1 on the test split for each model: (a) dataset A with all features, (b) dataset A with the submitted charge and the two charge-derived ratios removed, (c) dataset B. Bars show the mean over random splits and dots show individual splits; colours mark model families

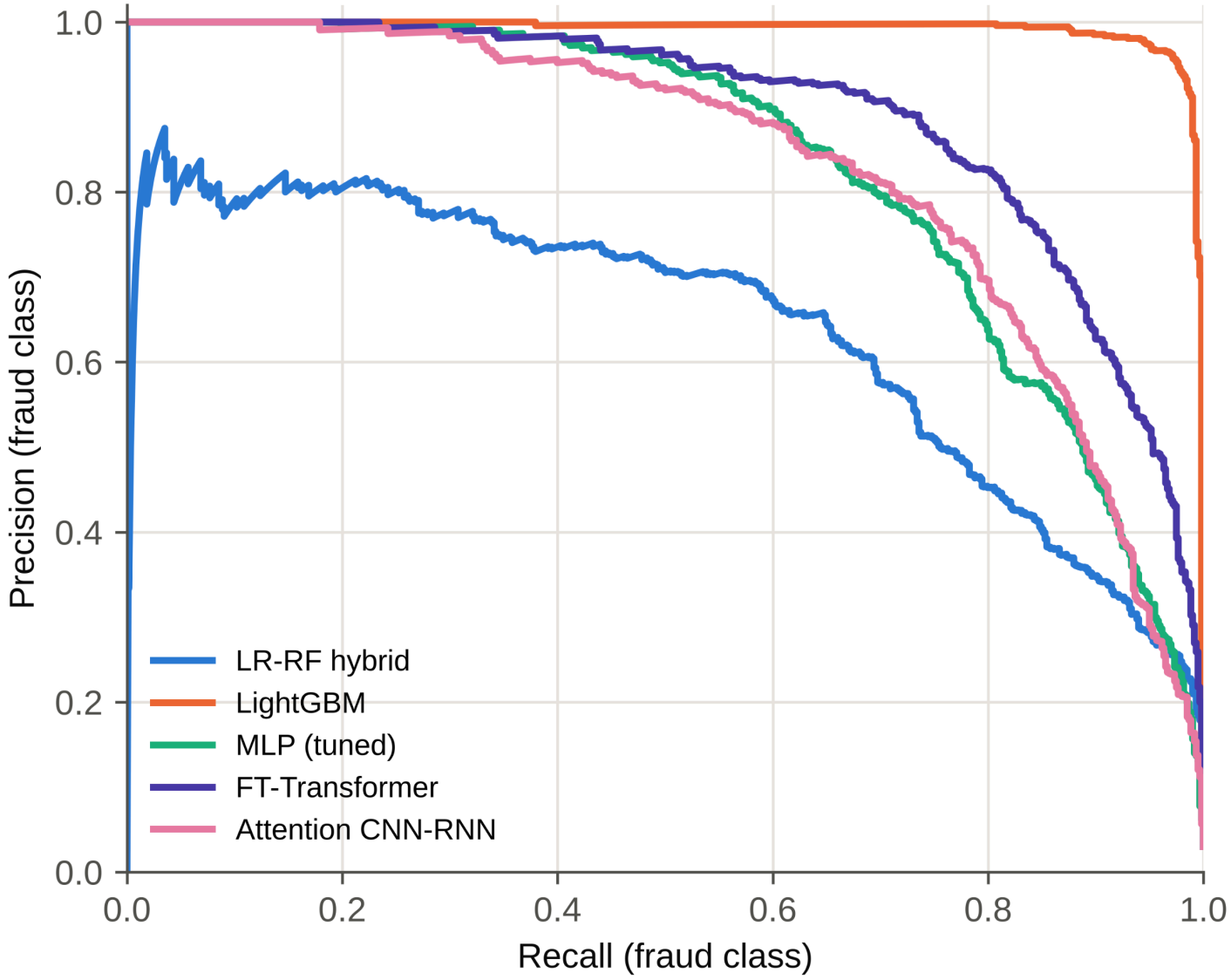


Figure 2

Precision-recall curves on the test set of one random split (seed 42) in the all-features setting, for one representative model of each family

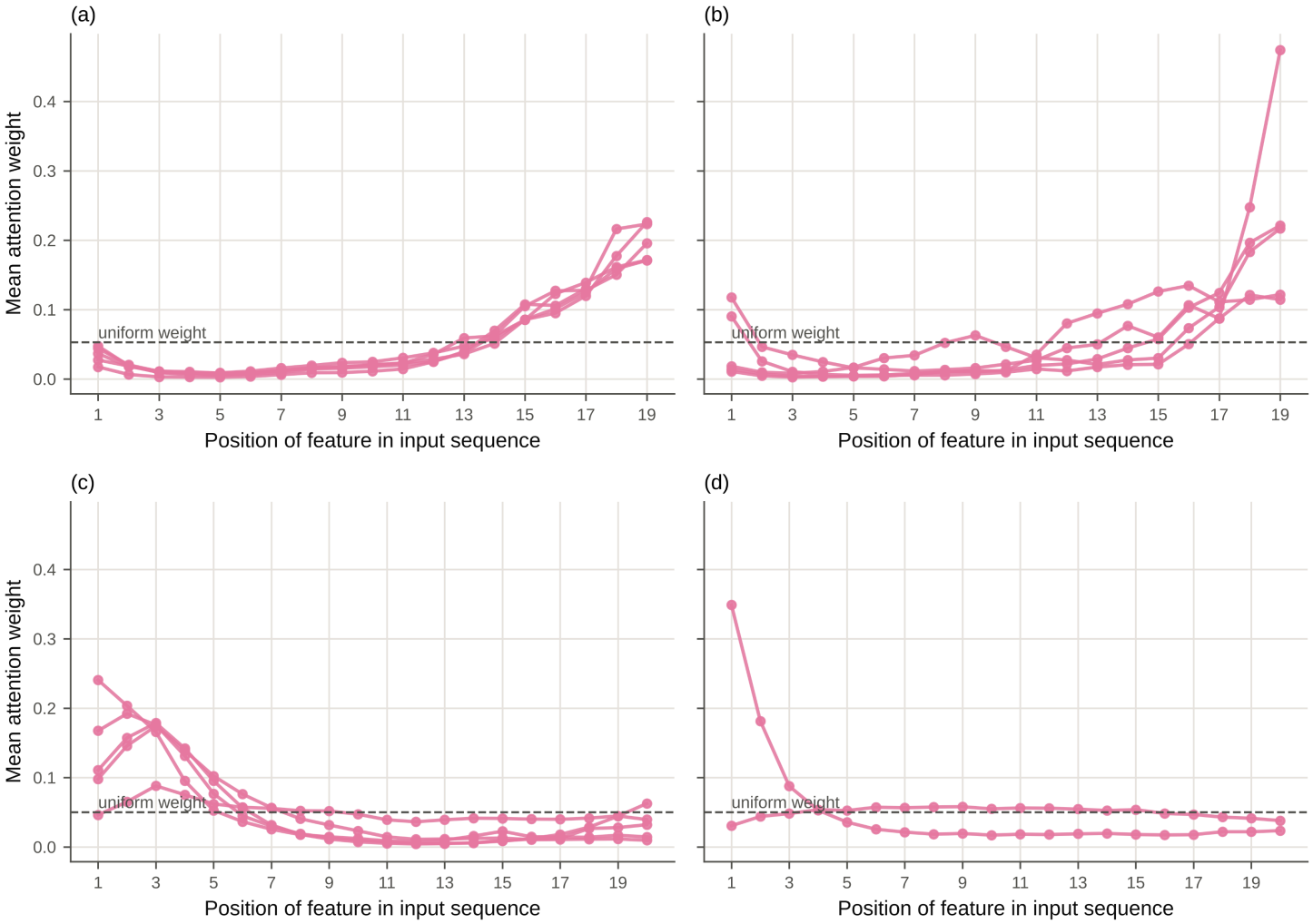


Figure 3

Mean attention weight of the attention CNN-RNN at each input position on the test set, for dataset A (a, b) and dataset B (c, d), with features in their original order (a, c) and in a random fixed order (b, d). Each line is one random split. The dashed line shows uniform weight