Advanced International Journal for Research (AIJFR)
E-ISSN: 3048-7641 ● Website: www.aijfr.com ● Email: editor@aijfr.com

Reproducibility Audit of a Five-Hidden-Layer Feedforward Network for Customer Churn Classification

Shabista Khan[1], Dipti Sharma[2]

[1]M. Tech Scholar, Department of Computer Science and Engineering, Patel College of Science and Technology, Indore, India
[2]Assistant Professor, Patel College of Science and Technology, Indore, India

Abstract.

This paper presents a reproducibility audit of a customer-churn experiment using a supplied 64,374-record CSV, a MATLAB source listing, and stored result figures. The active program excludes CustomerID, uses ten behavioral and service-related predictors, trains on records 1-10,000, and evaluates records 10,001-13,001 as a 3,001-record external holdout. It constructs a feedforward fitting network with five hidden layers of ten neurons each. Dense connectivity implies 561 trainable weights and biases. Stored MATLAB artefacts report Levenberg-Marquardt training, termination at epoch 24, best validation mean-squared error of 0.032864 at epoch 18, final training performance of 0.00902, and an all-development regression coefficient of 0.93989. A stored external figure reports 95.77% accuracy. The source variable calls the associated 4.23% quantity MAPE, although the implemented expression contains no percentage-error denominator; it is an absolute-label error and equals misclassification rate only when every rounded output is binary. No active particle-swarm-optimization routine exists in the executable code. The contribution is therefore an implementation-aligned audit rather than a new experiment: it corrects the model description and parameter count, documents the actual row partitions, separates internal validation from the external holdout, and marks unavailable evidence explicitly. Because the exact MAT-file, trained object, random seed, scores, and predictions are absent, the paper does not claim an independent rerun or invent confusion-matrix, AUC, calibration, or uncertainty results.

Keywords. Customer Churn; Reproducibility Audit; Feedforward Neural Network; Levenberg-Marquardt; Source-Code Verification; External Holdout; Research Integrity

1. Introduction

Customer churn classification estimates whether a customer will discontinue a service relationship within a defined horizon. A useful model must be evaluated on data that represent its intended decision setting, and its reported metrics must correspond to the operations actually executed by the code. Recent reviews and empirical studies cover ensembles, explainable tabular models, and composite neural architectures [2]-[9], but reported scores remain difficult to compare because label definitions, prevalence, preprocessing, and split protocols differ.

This paper addresses a narrower question: does the reported method match its executable evidence? The authors' prior paper described the experiment as a hybrid swarm-intelligence and deep-neural-network model [1]. The supplied MATLAB listing instead creates and trains a feedforward fitting network. The only block containing the label PSO is commented out, incomplete, and disconnected from the trained object. This manuscript corrects that characterization, retains stored numerical values with explicit provenance, and does not present textual revision as new experimental evidence.

The contributions are fourfold: (1) reconciliation of the CSV, MATLAB listing, stored figures, and prior report; (2) correction of the implemented architecture, trainable-parameter count, and error terminology; (3) audit of the exact row ranges and class distributions used by the source; and (4) a traceability boundary separating source-reported results from quantities that require an unavailable prediction vector or independent rerun.

2. Related Work and Audit Motivation

Systematic and practitioner-oriented reviews identify data definition, leakage control, class imbalance, model selection, and evaluation as recurring weaknesses in churn studies [2], [3]. Wu et al. combined churn prediction with customer segmentation [4], while recent tabular studies emphasize interpretable models and factor analysis [5]-[7]. Composite and hybrid neural approaches have also reported strong performance [8], [9]. These contributions show that high accuracy can arise from many model families; they do not make scores portable across different datasets and protocols.

Reproducibility requires more than citing an algorithm name. Dataset documentation should identify provenance, variables, exclusions, and intended use [10]. For executable studies, the model-building statement, preprocessing, data partitions, seed, stopping rule, saved predictions, and environment version are equally important. When a manuscript claims an optimizer that is absent from active code, the methodological label becomes more consequential than a stylistic error because it changes the claimed contribution.

The present study is therefore positioned as a technical audit. Its novelty lies in source-to-claim reconciliation and a corrected evidence boundary, not in claiming a second experiment on the same stored output. This distinction is central to interpreting the retained 95.77% value and to avoiding duplicate-publication or unsupported-novelty claims.

3. Source Package and Reproducibility Status

The source package contains churn_data.csv, a 132-line MATLAB text file, and screenshots of the network, training, regression, performance, state, and external prediction plots. The exact churndata.mat loaded by the source is absent. The trained network object, TrainingRecord object, internal split indices, random seed, and holdout scores are also absent. Consequently, descriptive statistics are recomputed from the CSV, whereas neural-network outcomes are transcribed from the stored figures.

The local CSV structure is consistent with the publicly listed Customer Churn Dataset [11], but the package does not include a download record or checksum from the original source. The audited local file has SHA-256 digest 29cdc0c919dd0d5c2b5e3010db8a6eb1aea77218fc4d46ae7b568474112ddca4. This digest identifies the analysed file; it does not establish collection provenance or prove equivalence with the missing MAT-file.

The relationship to the 2025 paper is disclosed in the title-page note, Introduction, and reference list [1]. The result figures are treated as evidence from that earlier experiment. To avoid implying newly generated output and to improve print quality, this manuscript presents newly drawn summaries of the transcribed values rather than reusing the low-resolution screenshots as primary illustrations.

4. Dataset Audit

4.1 Schema and Data Quality

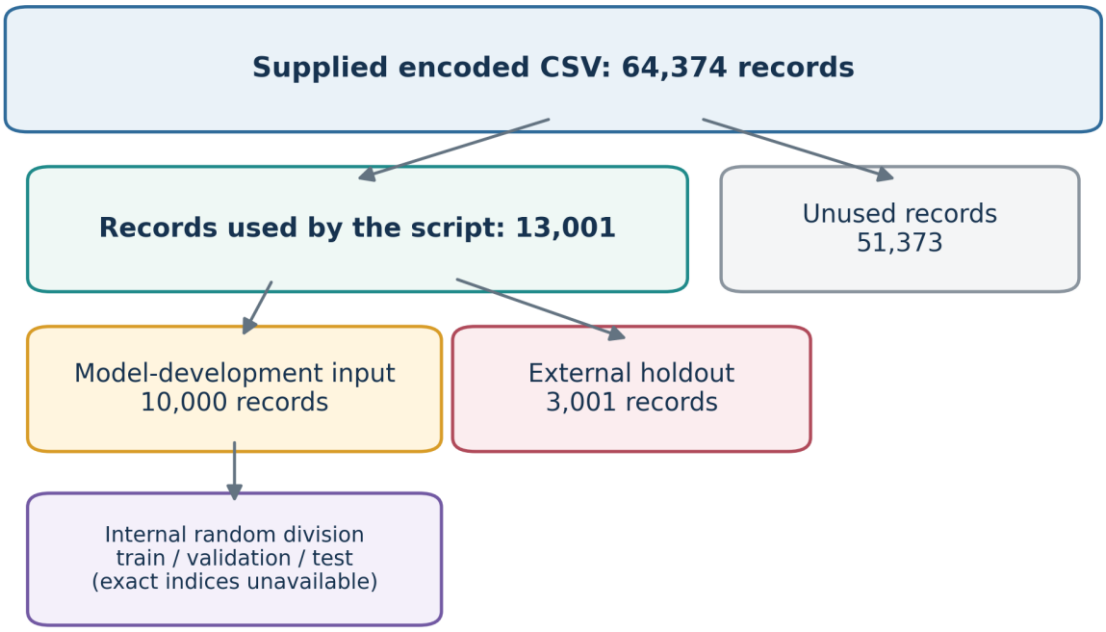
The CSV contains 64,374 rows and 12 integer-valued columns. CustomerID is unique, no exact duplicate rows are present, and no value is missing. Churn is binary. The ten predictors are Age, Gender, Tenure, Usage Frequency, Support Calls, Payment Delay, Subscription Type, Contract Length, Total Spend, and Last Interaction. Gender has observed codes {0,1}; Subscription Type and Contract Length have codes {1,2,3}. Because semantic category labels are not supplied, the codes are not assigned names in this paper.

Table 1. Class distribution across the implemented row partitions.

Partition	Rows	n	Non-churn	Churn	Churn rate
Development	1-10,000	10,000	7,373	2,627	26.27%
External holdout	10,001-13,001	3,001	2,225	776	25.86%
Unused	13,002-64,374	51,373	24,283	27,090	52.73%
Complete CSV	1-64,374	64,374	33,881	30,493	47.37%

The target distribution changes across row ranges. The full CSV is nearly balanced, but both partitions used by the script contain approximately 26% churn, while the unused remainder contains 52.73% churn. This pattern makes the row-selection rule scientifically relevant and precludes describing the active experiment as a random split of the whole file.

Figure 1. Index-defined development, external holdout, and unused partitions.

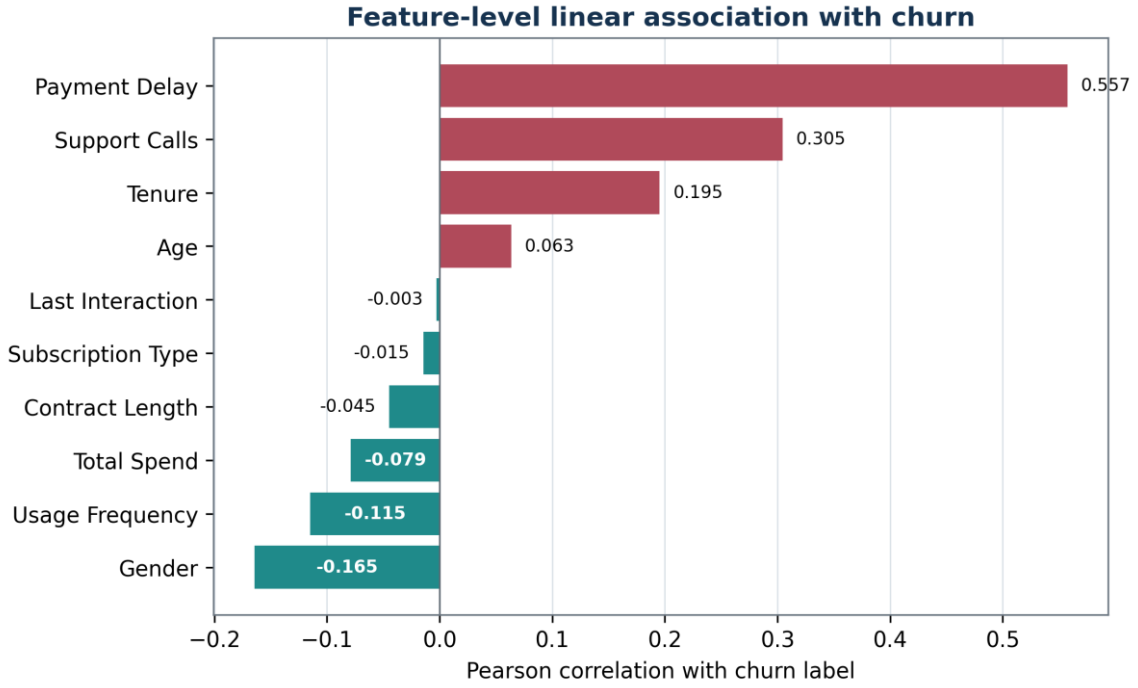


4.2 Descriptive Associations

On the complete CSV, the strongest Pearson correlations between eligible predictors and Churn are Payment Delay (0.557), Support Calls (0.305), and Tenure (0.195). Usage Frequency (-0.115), Total Spend (-0.079), and Gender code (-0.165) are negatively correlated. These are univariate descriptive associations, not neural-network feature attributions and not causal effects.

CustomerID has a correlation of approximately 0.530 with Churn, despite being an identifier. Its exclusion by the source code is therefore important; otherwise a model could exploit row-order or construction artefacts. The finding also motivates testing models across later cohorts instead of assuming independent and identically distributed rows.

Figure 2. Pearson correlations of the ten model predictors with Churn in the complete CSV.



5. Implemented Neural-Network Method

5.1 Outer Split and Internal Division

The script constructs a 10-column input matrix and a binary target vector, assigns rows 1-10,000 to trinp/trout, and assigns rows 10,001-13,001 to testinp/testout. The network routine internally divides the development block using dividerand, which performs a random train/validation/test division [12]. The stored interface shows those three internal subsets, but the exact indices and realized proportions are unavailable. The 3,001-row block is therefore an outer holdout, not the internal test subset shown in the regression panel.

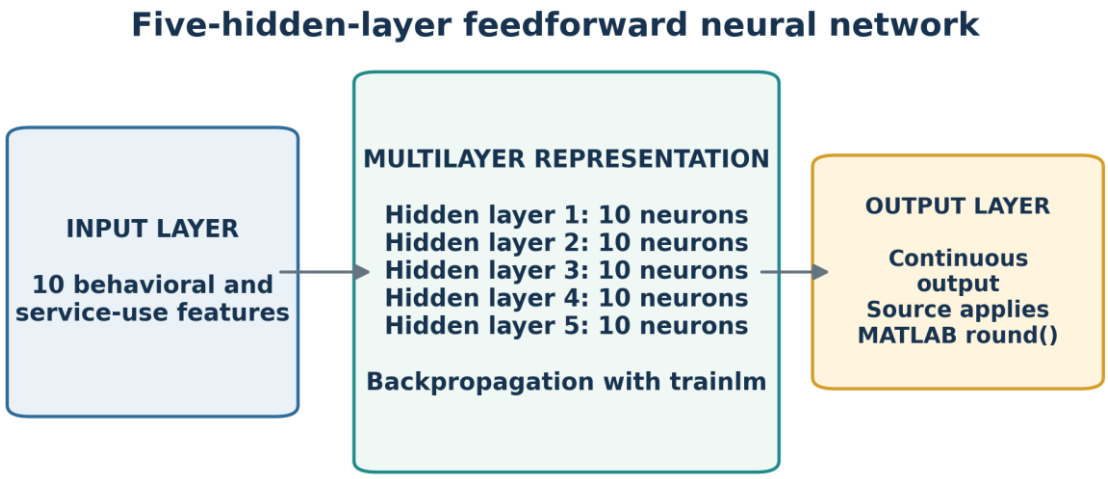
5.2 Architecture and Parameter Count

The legacy statement newfit(trinp', trout', [10 10 10 10 10]) creates a fitting network with five hidden layers of ten neurons each and one scalar output; fitnet is the documented legacy counterpart [13]. With dense connectivity, the model has 510 connection weights and 51 biases, giving 561 trainable parameters. The earlier count of 551 omitted ten bias terms and is corrected here.

$$P = (10 \times 10) + 4(10 \times 10) + (1 \times 10) + (5 \times 10) + 1 = 561 \quad (1)$$

The listing does not explicitly set transfer functions, automatic input-processing functions, division ratios, or the MATLAB release. Those properties are version- and object-dependent and cannot be verified without the saved network. This audit therefore does not assert undocumented activation or preprocessing defaults.

Figure 3. Implemented five-hidden-layer feedforward architecture.



5.3 Training Objective and Optimizer

The stored interface identifies mean squared error as the performance function and trainlm as the training function. MathWorks documents trainlm as Levenberg-Marquardt backpropagation [14]. For a residual vector e and Jacobian J, a standard update uses the damped normal-equation system shown in Eq. (2); the interface screenshot provides evidence of the selected routine, not enough information to reconstruct every internal numerical setting.

$$\text{theta}(k+1) = \text{theta}(k) - (J'J + \text{mu } I)^{(-1)}J'e \quad (2)$$

As of the access date, MathWorks marks dividerand, fitnet, and trainlm for future removal. They are cited here to document the legacy experiment, not as recommendations for a new implementation. A rerun should preserve the original MATLAB release in its environment record or port the workflow to supported classification APIs with equivalence checks.

5.4 Decision Rule and Metric Terminology

The network produces continuous holdout outputs. The source applies MATLAB round(), computes the mean absolute difference from the binary targets, multiplies by 100, and subtracts that value from 100. When every rounded output is either 0 or 1, this mean absolute difference equals the misclassification proportion. If any rounded value falls outside {0,1}, it remains an absolute-label error but is not a count of misclassified records.

$$\text{Reported error (\%)} = 100 \times (1/n) \sum |y(i) - \text{round}(f(x(i)))| \quad (3)$$

The variable name MAPE is incorrect because the expression contains no division by the actual value. The manuscript therefore uses source-derived error and source-reported accuracy, with the binary-output condition stated explicitly.

5.5 Absence of Active PSO

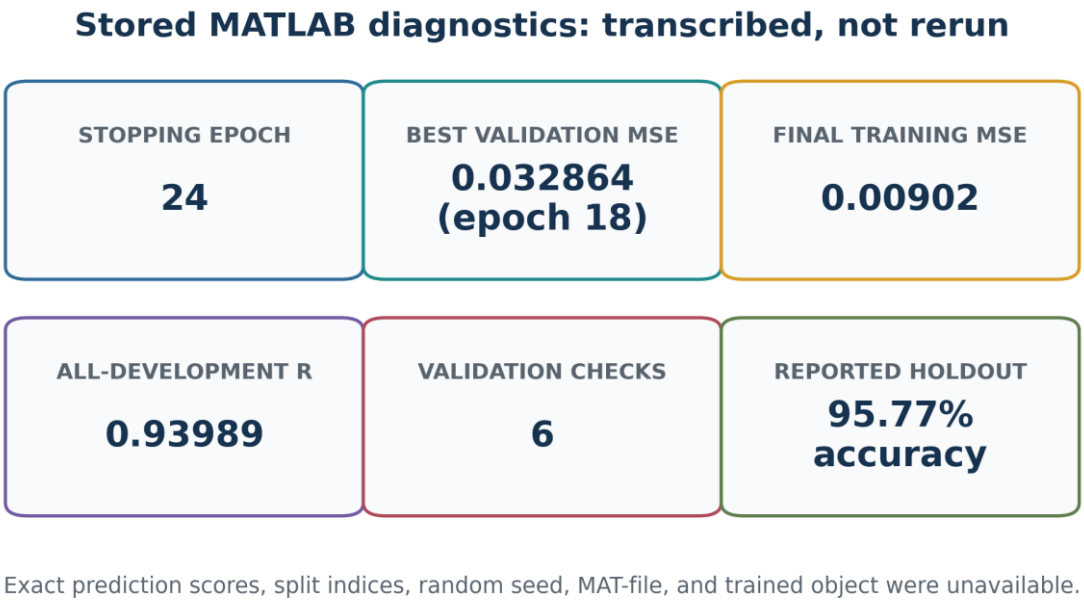
An implemented PSO routine would require particles, velocities, personal and global best states, a fitness function tied to network parameters, and iterative swarm updates. None appears in active code. A

multiline comment contains the label PSO but is ignored by MATLAB and does not invoke or modify NNmodel. The experimental result can only be attributed to the feedforward network and its configured training function.

6. Results

6.1 Stored Training and Internal Diagnostics

Figure 4. Values transcribed from the stored MATLAB diagnostics; no independent rerun is claimed.

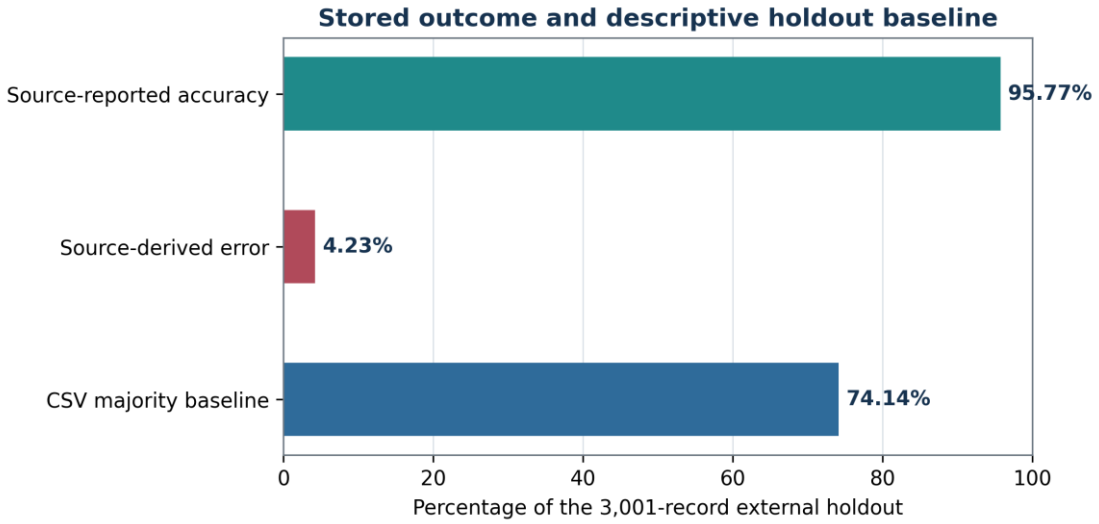


The stored training interface reports termination at epoch 24, performance 0.00902, gradient approximately 0.113, mu=0.000100, six validation checks, and elapsed time 00:00:03 in the recorded environment. The stored performance plot reports best validation MSE of 0.032864 at epoch 18. The elapsed time is environment-specific and is not generalized.

The stored regression panel reports R=0.95326 for internal training, 0.90798 for validation, 0.90745 for the internal test subset, and 0.93989 for all development samples. These coefficients measure linear association between continuous outputs and targets; they are not classification accuracies. Final training performance 0.00902 and best validation MSE 0.032864 apply to different partitions and should not be collapsed into one value.

6.2 External-Holdout Result

Figure 5. Source-reported holdout outcome and descriptive majority-class baseline from the audited CSV.



The external figure reports Prediction Accuracy=95.77 for 3,001 holdout records. Under the binary-rounded-output condition in Section 5.4, the value corresponds to a 4.23% misclassification rate and is consistent, after rounding to two decimals, with 2,874 correct labels and 127 errors. It does not reveal the true-positive, false-positive, true-negative, or false-negative counts. If rounded outputs outside {0,1} occurred, even the 127-error interpretation would not be valid.

Table 2. Traceability of the architecture and stored neural-network results.

Quantity	Reported value	Evidence boundary
Trainable parameters	561	Derived from active architecture
Epoch at stop	24	Stored training interface
Best validation MSE	0.032864 at epoch 18	Stored performance plot
Final training performance	0.00902	Stored training interface
All-development R	0.93989	Stored regression plot
External holdout n	3,001	Inclusive source indices
External accuracy	95.77%	Stored external plot
External error	4.23%	100 - 95.77; binary-output condition
Precision/recall/F1/AUC	Not available	Predictions/scores not supplied

7. Discussion

7.1 Interpretation

The stored experiment indicates that a compact 561-parameter feedforward network produced a strong source-reported result on the selected holdout. The corresponding slice of the audited CSV has a 74.14% majority-class baseline, so 95.77% is 21.63 percentage points higher if the missing MAT-file preserves identical rows and labels. This is a descriptive comparison, not an independently reproduced benchmark.

The result does not establish that five hidden layers are necessary. No shallow-network ablation or equally tuned logistic, random-forest, boosting, or alternative neural baseline is available. Recent churn studies demonstrate that both tabular and composite models can perform strongly [5]-[9]. A new comparison should use identical partitions, preprocessing, tuning budgets, and multiple seeds.

7.2 Metric Adequacy

Accuracy alone does not reveal churn recall or false-positive workload. A complete rerun should report the confusion matrix, precision, recall, specificity, F1, Matthews correlation coefficient, ROC-AUC, and PR-AUC from saved scores. If outputs are interpreted as probabilities, Brier score and calibration curves are required. A campaign-oriented analysis should also report lift or expected value at realistic contact capacities.

7.3 Validity and Generalization

The outer split follows row order without a documented timestamp. The sharp prevalence change in the unused remainder indicates dataset shift across row ranges. Testing on later or independently collected cohorts would provide stronger evidence. Category semantics, feature timing, and the CSV-to-MAT conversion should be documented before any claim of operational generalization.

Model-selection bias is another risk. If the external result influenced architecture or reporting choices, the holdout is no longer a pristine final test. The source package contains no search trace, so the extent of prior experimentation cannot be determined. A locked evaluation set and preserved experiment log are needed for a defensible new result.

7.4 Explainability and Responsible Use

Descriptive correlations suggest that payment delay and support calls are informative, but they are not model-specific explanations. Recent explainable churn work illustrates the value of feature-attribution analysis [5]. Any future explanation should include sensitivity analysis for correlated predictors and must not be represented as a causal retention effect. Age and gender codes require governance review, lawful purpose, and segment-level error analysis before operational use.

8. Limitations

- The neural network was not independently rerun; its results are transcribed from stored artefacts.
- The exact MAT-file, conversion process, random seed, network object, partitions, and predictions are unavailable.
- Only 13,001 of 64,374 rows enter the reported experiment, and the remainder has a different target distribution.

- The category-code meanings and temporal definition of churn are undocumented.
- Accuracy is the only stored external classification metric; class-wise performance, calibration, and the binary range of rounded outputs cannot be recovered.
- No controlled baseline, ablation, uncertainty analysis, fairness assessment, or economic evaluation is available.

9. Reproducible Extension Protocol

A new experiment should import the CSV directly, record its checksum, define category semantics, and select a deployment-aligned split before training. All encoding, scaling, feature selection, and resampling must be fit within development folds. Seeds and partition indices must be saved. Regularized logistic regression, random forest, XGBoost, shallow neural networks, and the documented five-layer network should receive comparable tuning budgets.

The final package should archive continuous scores and labels for every evaluation row, confusion matrices, calibration data, model checkpoints, environment versions, and timing. Repeated seeds and confidence intervals should quantify uncertainty. An out-of-time cohort should test drift, and retention value should be evaluated at predeclared thresholds. If PSO is added, it must be implemented, parameterized, logged, and ablated against the same non-PSO network.

10. Data and Evidence Availability

The public dataset page is cited in [11], and the audited local CSV is identified by its SHA-256 digest in Section 3. The MATLAB listing and stored figures form part of the authors' source package. The exact churndata.mat, trained network, internal partition indices, and prediction vectors were not available for this audit. The absence of those items is why this paper reports a traceability audit rather than an independent replication.

11. Conclusion

This reproducibility audit shows that the active evidence supports a five-hidden-layer feedforward network trained with Levenberg-Marquardt backpropagation, not a particle-swarm-optimized model. The corrected architecture contains 561 trainable weights and biases. A stored result reports 95.77% accuracy on a sequential 3,001-record holdout, but its interpretation remains conditional because the exact MAT-file, random seed, trained object, and predictions are unavailable. The paper's defensible contribution is source-to-claim correction, transparent provenance, and a reproducible extension protocol. Any future claim of improved performance should come from a fresh, versioned, leakage-aware comparison with saved predictions and controlled baselines.

References

1. P. Gehlot, C. K. Tiwari, and D. Nagar, “A hybrid swarm intelligence-deep neural network model for churn rate prediction,” International Journal of Scientific Research in Engineering and Management, vol. 9, no. 9, 2025. doi: 10.55041/IJSREM52624. https://doi.org/10.55041/IJSREM52624
2. M. Imani, M. Joudaki, A. Beikmohammadi, and H. R. Arabnia, “Customer churn prediction: A systematic review of recent advances, trends, and challenges in machine learning and deep

learning," Machine Learning and Knowledge Extraction, vol. 7, no. 3, art. 105, 2025. doi: 10.3390/make7030105. https://doi.org/10.3390/make7030105
3. M. A. Manzoor et al., "A review on machine learning methods for customer churn prediction and recommendations for business practitioners," IEEE Access, vol. 12, pp. 70434–70463, 2024. doi: 10.1109/ACCESS.2024.3402092. https://doi.org/10.1109/ACCESS.2024.3402092
4. S. Wu, W.-C. Yau, T.-S. Ong, and S.-C. Chong, "Integrated churn prediction and customer segmentation framework for telco business," IEEE Access, vol. 9, pp. 62118–62136, 2021. doi: 10.1109/ACCESS.2021.3073776. https://doi.org/10.1109/ACCESS.2021.3073776
5. S. S. Poudel, S. Pokharel, and M. Timilsina, "Explaining customer churn prediction in telecom industry using tabular machine learning models," Machine Learning with Applications, vol. 17, art. 100567, 2024. doi: 10.1016/j.mlwa.2024.100567. https://doi.org/10.1016/j.mlwa.2024.100567
6. V. Chang et al., "Prediction of customer churn behavior in the telecommunication industry using machine learning models," Algorithms, vol. 17, no. 6, art. 231, 2024. doi: 10.3390/a17060231. https://doi.org/10.3390/a17060231
7. S. K. Wagh, A. A. Andhale, K. S. Wagh, J. R. Pansare, S. P. Ambadekar, and S. H. Gawande, "Customer churn prediction in telecom sector using machine learning techniques," Results in Control and Optimization, vol. 14, art. 100342, 2024. doi: 10.1016/j.rico.2023.100342. https://doi.org/10.1016/j.rico.2023.100342
8. A. Khattak et al., "Customer churn prediction using composite deep learning technique," Scientific Reports, vol. 13, art. 17294, 2023. doi: 10.1038/s41598-023-44396-w. https://doi.org/10.1038/s41598-023-44396-w
9. X. Liu, G. Xia, X. Zhang, W. Ma, and C. Yu, "Customer churn prediction model based on hybrid neural networks," Scientific Reports, vol. 14, art. 30707, 2024. doi: 10.1038/s41598-024-79603-9. https://doi.org/10.1038/s41598-024-79603-9
10. T. Gebru et al., "Datasheets for datasets," Communications of the ACM, vol. 64, no. 12, pp. 86–92, 2021. doi: 10.1145/3458723. https://doi.org/10.1145/3458723
11. M. S. Azeem, "Customer churn dataset," Kaggle, dataset page, accessed Aug. 21, 2026. https://www.kaggle.com/datasets/muhammadshahidazeem/customer-churn-dataset
12. MathWorks, "dividerand: (To be removed) Divide targets into three sets using random indices," Deep Learning Toolbox documentation, accessed Aug. 21, 2026. https://www.mathworks.com/help/deeplearning/ref/dividerand.html
13. MathWorks, "fitnet: (To be removed) Function fitting neural network," Deep Learning Toolbox documentation, accessed Aug. 21, 2026. https://www.mathworks.com/help/deeplearning/ref/fitnet.html
14. MathWorks, "trainlm: (To be removed) Levenberg–Marquardt backpropagation," Deep Learning Toolbox documentation, accessed Aug. 21, 2026. https://www.mathworks.com/help/deeplearning/ref/trainlm.html