Deep Neural Network-Based Financial Forecasting Through Automated Feature Extraction and Adaptive Model Optimization

Cosima Salz
School of Computing, Clemson University, Clemson, SC, USA.
cosima.salz@clemson.edu

Giampaolo Inzaghi
Department of Computer Science, Colorado State University, Fort Collins, CO, USA.
inzaghigiampaolo@colostate.edu

Abstract

Financial forecasting with deep neural networks is constrained by two coupled design problems: informative features must be constructed from heterogeneous, noisy, and asynchronously released signals, and model parameters must remain effective as market conditions change. Manual feature engineering is difficult to maintain across assets and horizons, while static hyperparameter choices often become poorly matched to evolving regimes. This manuscript develops an integrated methodological framework for financial forecasting that couples automated feature extraction with adaptive model optimization. The representation component combines variable-specific encoders, temporal recurrence or convolution, and attention-based aggregation with group-sparse regularization, so that predictive features are learned from raw and moderately processed inputs while irrelevant variable blocks can be suppressed. The optimization component separates inner parameter estimation from outer hyperparameter and architecture adaptation through a bilevel objective, and it supports regularized online updates for controlled adaptation after initial training. Predictive uncertainty is treated as part of the forecasting object rather than an afterthought, using ensemble dispersion and residual variance estimates to inform calibration and risk-aware evaluation. The framework specifies leakage-resistant rolling-origin validation, purged temporal splits, and statistical comparison against credible baselines. Analytical assessment identifies the main trade-offs among representation flexibility, overfitting risk, adaptation speed, stability, interpretability, and computational cost. The contribution is a coherent specification of how automated representation learning and adaptive optimization can be combined and evaluated for financial forecasting without assuming that additional model complexity automatically improves predictive accuracy.

Keywords

financial forecasting; deep neural networks; automated feature extraction; adaptive optimization; nonstationarity; uncertainty calibration

1. Introduction

Deep neural networks provide flexible function approximators for high-dimensional nonlinear relationships [1], [2]. Financial forecasting has adopted such models to capture nonlinear dependence, interaction effects, and state-dependent predictability in market and firm-level data [3], [4]. The appeal of deep architectures is straightforward: financial variables

are numerous, correlated, noisy, and often observed at different frequencies, so a model that learns representations directly from data can reduce reliance on ad hoc indicator construction. Automated construction of representations has also been explored outside finance, including computational workflows that automate materials-oriented tasks [5], which illustrates a broader methodological interest in separating representation construction from downstream decision rules. In financial applications, systematic reviews nevertheless report substantial variation in architecture choice, validation practice, and reproducibility [6]. The central problem is not whether deep networks can fit financial data, but how to construct and adapt them in a way that preserves temporal integrity and yields forecasts whose uncertainty is explicit.

The representation problem is especially salient in forecasting. Recurrent networks such as long short-term memory models can represent sequential dependence and gating behavior [7], while attention mechanisms allow a model to weight available information dynamically rather than relying solely on a fixed lag structure [8]. These tools are useful because financial predictors rarely enter the forecast function independently. A volatility indicator may matter only under certain liquidity conditions; a valuation ratio may have different predictive content across monetary regimes; a text-derived signal may be informative only when its coverage is adequate. Automated feature extraction can learn such interactions, but it also increases the number of effective modeling choices. Without regularization and temporal controls, representation learning can memorize period-specific noise and produce forecasts that deteriorate outside the estimation window.

The optimization problem is equally important. Stochastic gradient methods provide the computational basis for training large networks [9], but financial forecasting requires decisions above the level of a single parameter update: learning rates, regularization strengths, architecture width, lookback length, and feature-group inclusion must be selected. Meta-learning formalizes the adaptation of such choices across related tasks [10], and survey work clarifies both the promise and the computational burden of this approach [11]. Financial time series also exhibit concept drift, in which the relationship between predictors and targets changes over time [12]. A forecasting system therefore needs an adaptation mechanism that responds to change without treating every recent fluctuation as a persistent regime shift. The framework developed here treats automated feature extraction and adaptive model optimization as linked components: representation choices determine what can adapt, and adaptation choices determine whether learned representations remain useful.

The manuscript makes one coherent contribution. It specifies a deep forecasting framework in which automated feature extraction is regularized at the level of variable groups, adaptive optimization is expressed as a bilevel procedure with an explicit inner estimation problem and an outer validation objective, and online adaptation is constrained by shrinkage toward a reference model. It then analyzes the conditions under which this design is defensible, with particular attention to look-ahead leakage, temporal validation, uncertainty calibration, and the economic interpretation of forecast errors. The treatment is methodological and analytical rather than an account of a completed empirical study. Its purpose is to make the design logic explicit enough that subsequent estimation and evaluation can be conducted without conflating representation learning, hyperparameter tuning, and regime adaptation.

2. Problem Formulation and Forecasting Setting

Consider a target variable y_{t+h} observed at horizon h and an information set X_t available at time t. The information set may include market returns, volatility measures, accounting variables, liquidity indicators, and text-derived signals, possibly with missing values and publication lags. The forecasting object is the conditional distribution of y_{t+h} given X_t, but a practical starting point is a point forecast map with an additive error term. Let E_φ denote a representation encoder parameterized by φ, g_θ a forecast head parameterized by θ, and ε_{t+h} an error term. The basic forecast is written as

$$y_{t+h} = g_\theta\big(E_\varphi(X_t)\big) + \varepsilon_{t+h}$$

where E_φ maps the raw or lightly processed information set into a latent representation, g_θ maps that representation to the target horizon, and ε_{t+h} collects unpredictable variation and model misspecification. The formulation is deliberately general: E_φ may contain recurrent layers, temporal convolutions, variable-specific embeddings, or attention blocks, and g_θ may be a linear readout or a shallow nonlinear head. The important constraint is temporal: every element of X_t must be observable before t, and any transformation that uses future information, including normalization estimated over the full sample or feature selection conducted before splitting, violates the forecasting problem.

Financial data add three complications that the framework must respect. First, observations are not exchangeable over time; distributional stability is a local rather than global assumption. Second, labels may mature with delay, so an update at time t cannot use y_{t+h} until the horizon has elapsed. Third, forecast losses are often asymmetric in economic terms: a small numerical error may imply a large portfolio decision error, while a larger error in a low-stakes period may be economically irrelevant. These complications do not change the basic forecast equation, but they determine how the encoder, optimizer, and evaluation procedure should be specified.

3. Methodological Framework

3.1 Automated Feature Extraction

The representation encoder is organized around variable groups rather than individual raw columns. Each group corresponds to an economically interpretable block, such as a return-based block, a volatility block, a fundamental accounting block, or a text-derived sentiment block. Group-level structure serves two purposes. It allows the network to learn nonlinear transformations within a block while retaining a meaningful unit for regularization, and it makes feature suppression auditable. A variable-specific encoder first maps each block into a sequence of hidden states. Temporal dependence can be modeled with recurrent units [7] or with convolutional and attention mechanisms [8]. The resulting hidden states are then aggregated by a learned attention score so that the forecast head receives a representation weighted toward currently relevant information.

Let h_{t,j} denote the hidden representation of feature group j at time t, let s_{t,j} be a learned relevance score, and let m be the number of groups. The aggregated representation is

$$z_t = \sum_{j=1}^{m} \alpha_{t,j} h_{t,j}, \alpha_{t,j} = \exp\frac{\big(s_{t,j}\big)}{\sum_{k=1}^{m} \exp\big(s_{t,k}\big)}$$

where α_{t,j} is the normalized attention weight assigned to group j, h_{t,j} is its encoded state, and z_t is the representation passed to the forecast head. The attention weights are

diagnostic rather than causal: they indicate which encoded groups the model uses conditional on its parameters, not which variables causally drive the target. This distinction matters because financial attention maps can be unstable across adjacent windows even when predictive performance is similar. The framework therefore treats attention as an internal weighting mechanism and requires separate stability checks before any economic interpretation is attached to it.

Regularization is applied at the group level to prevent the encoder from distributing small weights across many weak signals. Let ρ be a forecast loss such as a Huber or quantile loss, let W_j denote the parameters associated with group j, and let λ_1 and λ_2 be nonnegative penalty parameters. A training objective for fixed hyperparameters is

$$L_tr \;=\; (1/N)\sum_\{i=1\}\^N \rho(y_i, g_\theta(E_\varphi(X_i))) \;+\; \lambda_1||\theta||_2\^2 \;+\; \lambda_2\sum_\{j=1\}\^m ||W_j||_2$$

where N is the number of training observations, ρ measures forecast loss, the first penalty controls overall model complexity, and the second imposes group sparsity on representation parameters. When λ_2 is large, entire variable groups can be driven toward zero, reducing variance at the cost of possibly discarding weak but genuine signals. When λ_2 is small, the encoder can exploit subtle interactions but becomes more vulnerable to period-specific noise. The choice is not a single global constant; it defines a path of models whose validation behavior must be examined across rolling windows.

3.2 Adaptive Model Optimization

Adaptive optimization is represented as a bilevel problem. The inner problem estimates forecast parameters for a given configuration, while the outer problem selects configuration variables using validation performance. Configuration variables may include learning rate, weight decay, group-sparsity strength, attention temperature, lookback length, and the number of hidden units. Let τ index a forecasting task, such as an asset class, horizon, or regime window, and let P(τ) denote the distribution of tasks used for adaptation. Let η collect the outer configuration variables and let θ(η,φ) be the inner solution. The outer objective is

$$min_\eta\, E_\{\tau \sim P(\tau)\}[L_val(\theta*(\eta,\varphi), D_val\^\tau)]\; s.t.\; \theta* = argmin_\theta\, L_tr(\theta,\varphi,\eta)$$

where L_val is the validation loss on held-out temporal data, D_val^τ is the validation set for task τ, and θ is the parameter vector obtained by minimizing the inner training objective. The formulation is close in spirit to meta-learning over related tasks [10], [11], but the financial setting imposes a stricter validation discipline: the outer loss must be computed on data that occur after the inner training window and before any test window. If the inner and outer samples overlap, the procedure selects configurations that appear effective only because they have seen information that would not have been available at forecast time.

Initial training with a selected configuration does not exhaust the adaptation problem. Financial relationships can change gradually or abruptly, and a model that is never updated may become stale. Let θ_t be the parameter vector before forecasting period t, let L_t be the loss on matured observations available at that time, let η_t be a learning rate, let θ_ref be a reference model estimated on an earlier stable window, and let β be a shrinkage parameter. A regularized online update is

$$\theta_\{t+1\} \;=\; \theta_t \;-\; \eta_t\, \nabla_\theta\, [L_t(\theta_t) \;+\; \beta\, ||\theta_t \;-\; \theta_ref||_2\^2]$$

where ∇_θ denotes the gradient with respect to parameters, L_t is computed only on labels whose horizon has matured, and the penalty discourages drift away from the reference

model. When β is zero, the update reduces to recursive gradient adaptation, which can respond quickly but may chase transient noise. As β increases, adaptation slows and the model remains closer to the reference solution. The appropriate balance is a stability issue, not merely a tuning issue: a system that updates too aggressively accumulates estimation variance, while a system that updates too slowly fails to incorporate persistent change. The use of matured labels and a reference model are therefore part of the method, not implementation details.

3.3 Uncertainty and Evaluation Logic

A point forecast is insufficient for financial decision-making because forecast error interacts with position sizing, risk limits, and model selection. The framework treats predictive uncertainty as a modeled quantity. One transparent approach is to train an ensemble of M forecasters with different initializations, subsamples, or architecture seeds, and to combine ensemble dispersion with an estimate of residual variance. Let f_m(X_t) be the m-th ensemble forecast, let f̄(X_t) be the ensemble mean, and let σ̂^2 be an estimated irreducible or residual variance term. A predictive variance summary is

$$Var(y_\{t+h\}|X_t) \;=\; (1/M)\sum_\{m=1\}\text{^}M\,(f_m(X_t) - f(X_t))\text{^}2 \;+\; \hat{\sigma}\text{^}2$$

where M is the number of ensemble members and the first term captures model disagreement while the second captures residual noise. Deep ensembles provide a practical uncertainty estimate without requiring a fully Bayesian treatment [15], and approximate Bayesian alternatives based on dropout offer another route [16]. The choice between them should be evaluated by calibration and sharpness rather than by whether the resulting intervals appear visually plausible. In particular, ensemble disagreement can be small when all members share the same misspecification, so calibration must be checked against realized forecast errors.

Evaluation must preserve the temporal ordering that defines the forecasting problem. Rolling-origin evaluation is appropriate because it mimics the sequence of information sets a forecaster would have faced. Purging and embargo rules are needed when features or labels overlap across adjacent periods [13]. Cross-validation can be used for model selection only when the splitting scheme respects temporal dependence; otherwise it can produce optimistic estimates [14]. The evaluation should report point accuracy, such as mean absolute error or root mean squared error, alongside directional accuracy and an economic utility measure that accounts for transaction costs and turnover. Differences among forecasters should be tested rather than inferred from raw averages; the Diebold-Mariano procedure provides a standard test for equal predictive accuracy [17], and the model confidence set provides a way to retain a set of statistically indistinguishable models rather than selecting a single winner [18]. Large-scale forecasting evidence also cautions against assuming that a more complex model dominates simpler benchmarks across all series and horizons [19]. For multi-horizon settings, models that expose temporal fusion and interpretable attention can be useful, but their added capacity must be justified by out-of-sample evaluation [20].

4. Analytical Assessment

The framework's central analytical trade-off is between representation flexibility and estimation stability. Automated feature extraction reduces dependence on handcrafted indicators and can represent interactions that a fixed linear specification would miss. However, every additional learned transformation increases the effective number of models that could have been selected by the training procedure. Group-level sparsity is a partial response

because it forces competition among variable blocks rather than allowing the encoder to spread weight thinly across many predictors. The penalty parameter λ_2 therefore controls a bias-variance trade-off at the block level: stronger penalties yield more stable but less expressive representations, while weaker penalties allow richer interactions but increase the risk that the encoder captures period-specific artifacts.

The bilevel formulation introduces a second trade-off between selection quality and computational cost. Exact bilevel optimization is expensive because the inner problem must be solved repeatedly for candidate configurations. Approximate schemes, such as truncated inner training or shared inner solutions across nearby candidates, reduce cost but can bias the outer selection toward configurations that are easy to optimize rather than configurations that generalize. In financial forecasting, this bias is dangerous because the outer validation sample is small relative to the number of candidate configurations. A defensible procedure therefore limits the configuration space, uses nested rolling validation, and reports the variability of selected configurations across windows. A configuration that wins in one window but is never selected again should not be treated as a stable modeling insight.

Online adaptation adds a third trade-off between responsiveness and drift. The regularized update in Equation 5 makes the adaptation rate explicit through η_t and the shrinkage strength explicit through β. If the data-generating process changes persistently, a positive β that is too large will slow recovery and leave the model anchored to an obsolete reference. If β is too small, the model may interpret sampling noise as regime change and increase turnover in downstream decisions. The practical implication is that adaptation should be evaluated at the level of forecast stability and economic turnover, not only at the level of average loss. A model that improves average accuracy while producing unstable position changes may be inferior for risk-controlled applications.

Uncertainty modeling completes the assessment. Equation 6 separates model disagreement from residual noise, but the decomposition is only as meaningful as the ensemble construction. Ensembles that differ only in random initialization may understate uncertainty because they share the same architecture and feature set. Ensembles that differ in feature groups, training windows, or model families may better represent structural uncertainty, at the cost of greater computational burden and more complex calibration. The framework therefore treats uncertainty estimation as a design axis rather than a fixed add-on. Calibration, sharpness, and the cost of forecast errors should be reported together, since a narrow interval that is poorly calibrated is less useful than a wider interval that supports reliable risk decisions.

5. Failure Mechanisms and Boundary Conditions

Several failure mechanisms are specific to deep financial forecasting. The first is temporal leakage through preprocessing. Normalization statistics, imputation models, and feature-selection decisions must be estimated within each training window and then applied to subsequent validation or test windows. If these operations are performed on the full sample, the model can appear to generalize while actually using information from the future. The second is leakage through overlapping labels. When the forecast horizon exceeds the sampling interval, adjacent labels share information; purging and embargo rules are required to prevent training and validation observations from overlapping in time [13], [14].

The third mechanism is over-adaptation. Financial time series contain many short-lived apparent patterns, and an adaptive optimizer can mistake noise for a persistent regime.

Regularization toward a reference model, conservative update rates, and drift detection based on out-of-sample deterioration can reduce this risk. Drift detection should not be based on training loss alone, because a model can fit recent data while losing predictive validity. Instead, adaptation should be triggered by sustained changes in validation performance, feature distributions, or calibration error. The concept-drift literature provides a useful vocabulary for distinguishing gradual change, abrupt change, and recurring regimes [12], but financial applications require an economic cost model to decide when adaptation is worthwhile.

The fourth mechanism is interpretability overreach. Attention weights and feature attributions can help diagnose model behavior, but they do not identify causal effects. A variable may receive high weight because it proxies for an omitted risk factor, because it is correlated with the target during the estimation window, or because it helps the model correct for missingness. The framework therefore recommends stability analysis across windows and perturbations, and it treats interpretability as evidence about model reliance rather than evidence about market mechanisms. The fifth mechanism is evaluation proliferation. Testing many architectures, feature sets, and adaptation schedules against the same test period inflates the chance of selecting a model that performs well by chance. The model confidence set is useful precisely because it retains multiple plausible models rather than forcing a single winner [18], and benchmark-based reporting helps situate complex models against simple alternatives [19].

Boundary conditions follow from these mechanisms. The framework is most defensible when the information set is rich enough to support representation learning, when labels mature within a predictable delay, and when there are sufficient rolling windows to assess stability. It is less defensible when the sample is short, when structural breaks are frequent and undocumented, or when the forecast target is so noisy that differences among models are dominated by irreducible variance. In those settings, a simpler regularized model with transparent features may be preferable, not because deep learning is intrinsically unsuitable, but because the evidence required to justify a complex adaptive system is unavailable. The framework's value lies in making these conditions explicit before complexity is added.

6. Conclusion

This manuscript has specified a deep neural network framework for financial forecasting that links automated feature extraction with adaptive model optimization. The representation component encodes variable groups, aggregates them with learned attention, and regularizes group-level parameters so that feature suppression remains an explicit modeling decision. The optimization component separates inner parameter estimation from outer configuration selection through a bilevel objective, and it adds regularized online updates for controlled adaptation after initial training. Uncertainty is modeled through ensemble dispersion and residual variance, and evaluation is organized around rolling-origin splits, purging, embargoes, statistical forecast comparison, and economic utility. Analytical assessment shows that the framework does not eliminate the core tensions of financial forecasting; it makes them operational. Representation flexibility increases expressive power but raises overfitting risk, adaptation improves responsiveness but can amplify noise, and uncertainty estimates are useful only when calibrated against realized errors.

The main implication is that automated feature extraction and adaptive optimization should be evaluated jointly rather than as independent innovations. A representation that is effective under a fixed model may fail when the optimization procedure adapts aggressively,

and an adaptation rule that appears stable may depend on features that are themselves unstable. The framework provides a disciplined language for stating those dependencies: which information is available at forecast time, which parameters are estimated on which window, how configuration choices are validated, and how uncertainty enters the forecast. These specifications matter for scientific reproducibility and for practical risk control. Financial forecasting systems that omit them can appear sophisticated while remaining vulnerable to leakage, nonstationarity, and uncalibrated confidence. The contribution here is therefore a technically grounded design and assessment framework, not a claim of superior predictive accuracy, and its appropriate use is to guide transparent estimation and credible out-of-sample evaluation.

References

1. LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature, 521(7553), 436–444.
2. Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep learning. MIT Press.
3. Fischer, T., & Krauss, C. (2018). Deep learning with long short-term memory networks for financial market predictions. European Journal of Operational Research, 270(2), 654–669.
4. Gu, S., Kelly, B., & Xiu, D. (2020). Empirical asset pricing via machine learning. The Review of Financial Studies, 33(5), 2223–2273.
5. Luo, D., Chen, D., Wang, J., Zhu, G., & Xu, W. (2020). The smart robot crafting approach to computing materials. Construction Robotics, 4(3), 239-249.
6. Sezer, O. B., Gudelek, M. U., & Ozbayoglu, A. M. (2020). Financial time series forecasting with deep learning: A systematic literature review: 2005–2019. Applied Soft Computing, 90, 106181.
7. Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural Computation, 9(8), 1735–1780.
8. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. Advances in Neural Information Processing Systems, 30, 5998–6008.
9. Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. In Proceedings of the 3rd International Conference on Learning Representations.
10. Finn, C., Abbeel, P., & Levine, S. (2017). Model-agnostic meta-learning for fast adaptation of deep networks. In Proceedings of the 34th International Conference on Machine Learning (Vol. 70, pp. 1126–1135). PMLR.
11. Hospedales, T., Antoniou, A., Micaelli, P., & Storkey, A. (2022). Meta-learning in neural networks: A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence, 44(9), 5149–5169.
12. Gama, J., Žliobaitė, I., Bifet, A., Pechenizkiy, M., & Bouchachia, A. (2014). A survey on concept drift adaptation. ACM Computing Surveys, 46(4), 1–37.
13. López de Prado, M. (2018). Advances in financial machine learning. Wiley.
14. Bergmeir, C., & Benítez, J. M. (2012). On the use of cross-validation for time series predictor evaluation. Information Sciences, 191, 192–213.

15. Lakshminarayanan, B., Pritzel, A., & Blundell, C. (2017). Simple and scalable predictive uncertainty estimation using deep ensembles. Advances in Neural Information Processing Systems, 30, 6402–6413.

16. Gal, Y., & Ghahramani, Z. (2016). Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In Proceedings of the 33rd International Conference on Machine Learning (Vol. 48, pp. 1050–1059). PMLR.

17. Diebold, F. X., & Mariano, R. S. (1995). Comparing predictive accuracy. Journal of Business & Economic Statistics, 13(3), 253–263.

18. Hansen, P. R., Lunde, A., & Nason, J. M. (2011). The model confidence set. Econometrica, 79(2), 453–497.

19. Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2018). The M4 Competition: Results, findings, conclusion and way forward. International Journal of Forecasting, 34(4), 802–808.

20. Lim, B., Arik, S. O., Loeff, N., & Pfister, T. (2021). Temporal fusion transformers for interpretable multi-horizon time series forecasting. International Journal of Forecasting, 37(4), 1748–1764.