ISSN 2723-6471

Calibrated Probabilistic Forecasting of Heavy-Tailed Financial Returns via Spiking Neural Networks with Dynamic Adaptive Computation

Rani Farida Sinaga[1], Sutarman[2,*], Tulus[3], Open Darnius[4]

[1]*Graduate School of Mathematics, Universitas Sumatera Utara, Jalan Dr. T. Mansur No. 9, Kota Medan 20222, Indonesia*

[1]*Department of Mathematics Education, Universitas HKBP Nommensen, Jalan Sutomo No. 4A, Kota Medan 20234, Indonesia*

[2,3]*Department of Mathematics, Universitas Sumatera Utara, Jalan Dr. T. Mansur No. 9, Kota Medan 20222, Indonesia*

[4]*Programs of Statistics, Universitas Sumatera Utara, Jalan Dr. T. Mansur No. 9, Kota Medan 20222, Indonesia*

(Received: May 15, 2026; Revised: July 20, 2026; Accepted: September 29, 2026; Available online: October 29, 2026)

Abstract

Probabilistic forecasting of financial time series demands models that simultaneously capture heavy-tailed return distributions, time-varying volatility, and structural regime changes. Classical econometric models provide interpretable but rigid parametric specifications, while modern deep forecasting architectures achieve flexibility at the cost of substantial computational overhead. This paper introduces a novel spiking neural network augmented with dynamic adaptive computation for energy-efficient probabilistic sequence modeling. The proposed architecture processes sequential inputs through a leaky integrate-and-fire backbone with learnable, activity-dependent firing thresholds inspired by biological synaptic scaling, outputting the parameters of a Student-t distribution with input-dependent degrees of freedom. The experimental protocol employs a strict chronological train-test split to prevent data leakage. Evaluated across seven synthetic and empirical financial datasets, achieves aggregate calibration comparable to the best-performing fixed-threshold spiking baseline while substantially improving upon the classical and dense neural baselines. Specifically, it shares the lowest Mean Absolute Coverage Error of one point two zero percentage points, with statistical tests confirming significant improvements in coverage and continuous ranked probability score on heavy-tailed assets like Bitcoin. While dense baselines achieve competitive point accuracy, they exhibit severe under-coverage on these assets. In contrast, the proposed approach maintains robust calibration during extreme market regimes, as validated by formal Value-at-Risk backtests supplemented by Expected Shortfall diagnostics. Regime-stratified evaluation demonstrates that the adaptive threshold mechanism prevents the catastrophic over-coverage of classical models during high-volatility crises. Computational analysis indicates that the event-driven architecture maintains sparse spike activity, translating into an estimated ninety-eight point three percent reduction in the analytical software-level energy proxy compared to dense inference baselines. Downstream applications to volatility tracking and unsupervised anomaly detection validate the practical utility of these predictive distributions for real-time risk management, paving the way for resource-efficient models on future neuromorphic hardware.

Keywords: Spiking Neural Networks, Probabilistic Forecasting, Dynamic Adaptive Computation, Heavy-Tailed Distributions, Financial Time Series, Energy Efficiency, Calibration, Anomaly Detection

1. Introduction

Accurate probabilistic forecasting of financial asset returns is a cornerstone of modern risk management, portfolio optimization, and regulatory capital computation [1]. Recent advancements emphasize that risk-aware decision-making in financial systems fundamentally relies on robust uncertainty quantification rather than mere point estimates [2], [3]. Unlike point forecasting, which seeks only a single expected value, probabilistic forecasting aims to estimate the full conditional distribution of future returns, enabling practitioners to quantify uncertainty, construct calibrated prediction intervals, and detect anomalous market events in real time [4].

The dominant paradigm for modeling time-varying volatility in financial returns remains the family of Generalized Autoregressive Conditional Heteroskedasticity (GARCH) models [5], building on the foundational ARCH framework of [6]. While extensions such as EGARCH [7], GJR-GARCH [8], and Component GARCH [9] have enriched the ability to capture asymmetric effects and long-memory dynamics, these models share a fundamental limitation: they

***Corresponding author: Sutarman (sutarman@usu.ac.id)**
DOI: https://doi.org/10.47738/jads.v7i4.1558
This is an open access article under the CC-BY license (https://creativecommons.org/licenses/by/4.0/).
© Authors retain all copyrights

rely on fixed parametric specifications for both the variance equation and the conditional distribution. In practice, financial returns exhibit extreme leptokurtosis and non-stationary tail behavior that defy Gaussian assumptions [10], [11]. For instance, assets like Bitcoin exhibit excess kurtosis exceeding 11, alongside regime shifts that cannot be adequately captured by a single, globally estimated degrees-of-freedom parameter.

Deep learning has emerged as a powerful alternative. Autoregressive recurrent networks such as DeepAR [12] and attention-based models such as the Temporal Fusion Transformer [13] can learn flexible, data-driven conditional distributions. While deep learning frameworks have shown considerable promise in financial forecasting [14], [15], [16], these architectures typically employ dense recurrent or transformer backbones that incur substantial computational cost. Inspired by recent insights into adaptive computation as a mechanism for dynamic resource allocation [17], this study explore whether biologically inspired models can achieve competitive performance with greater efficiency.

Spiking neural networks (SNNs), often regarded as the third generation of neural network models [18], offer a compelling alternative. By communicating through discrete, event-driven spikes rather than continuous activations, SNNs can achieve orders-of-magnitude improvements in energy efficiency on neuromorphic processors [19]. Recent advances in surrogate gradient learning [20], [21], [22] have enabled the training of deep spiking architectures with performance competitive to their rate-based counterparts [23]. However, existing SNN applications have focused almost exclusively on classification and point regression tasks. To the best of our knowledge, this work is among the first to employ spiking neural networks for probabilistic sequence modeling with explicit calibration analysis and input-dependent heavy-tailed predictive distributions for financial return forecasting.

2. Literature Review

Our work lies at the intersection of four research threads: (i) conditional heteroskedasticity models from econometrics, (ii) deep probabilistic forecasting, (iii) spiking neural networks and surrogate gradient learning, and (iv) calibration theory for probabilistic predictions.

2.1. Volatility Modeling: From ARCH to Modern Variants

Volatility modeling originated with Engle's ARCH model [6], which established conditional variance as a dynamic stochastic process. Bollerslev subsequently generalized this with the GARCH model [5], making the GARCH(1,1) specification a standard empirical benchmark. To capture specific market regularities, several extensions were developed: EGARCH [7] models the logarithm of variance to naturally capture the leverage effect; GJR-GARCH [8] incorporates an asymmetric indicator for differential shock impacts; and CGARCH [9] decomposes volatility into permanent and transitory components to capture long-memory characteristics.

Over the past decade, volatility forecasting has evolved toward deep learning paradigms (e.g., RNNs, TCNs, Transformers, and hybrids) to capture nonlinear dependencies [13], [16], [24] and shifted from deterministic variance estimation to full probabilistic distribution forecasting to better characterize tail risk [12], [25], [26] However, critical limitations persist. Classical GARCH models assume fixed parametric distributions (e.g., Gaussian or fixed-parameter Student-t), restricting their ability to adapt to time-varying tail behavior. Conversely, many deep learning models prioritize point forecasting, lacking robust mechanisms for dynamic uncertainty quantification. Consequently, both approaches often struggle to characterize rapidly evolving market conditions during periods of financial stress and structural breaks.

To address these gaps, the proposed Spiking Neural Network with Dynamic Adaptive Computation (SNN+DAC) advances probabilistic volatility modeling by treating the Student-*t* distribution's degrees-of-freedom parameter as an input-dependent variable rather than a globally fixed constant. This mechanism enables the predictive distribution to continuously adapt to changing market conditions while leveraging the event-driven, energy-efficient computation of spiking networks. Consequently, SNN+DAC bridges classical econometric volatility models with modern adaptive probabilistic forecasting architectures tailored for heavy-tailed financial returns.

2.2. Deep Probabilistic Forecasting

Deep learning has recently extended to probabilistic forecasting, estimating the full conditional distribution $p(y_{t+1} \mid x_{1:t})$. DeepAR [12] uses an autoregressive RNN to output parametric likelihoods, serving as a strong but computationally heavy baseline. Similarly, the Temporal Fusion Transformer (TFT) [13] combines attention with gated recurrent structures for interpretable multi-horizon forecasts, though it is heavier than RNNs and typically requires large multivariate ensembles. Regarding uncertainty quantification, Neal [27] established distributions over network weights for principled but expensive inference, while Kuleshov et al. [28] achieved well-calibrated predictive distributions via learned post-hoc transformations.

Our approach is complementary: rather than increasing architectural depth, this study propose a biologically-inspired spiking architecture for energy-efficient, event-driven probabilistic forecasting. SNN+DAC directly outputs Student-t parameters with learnable degrees of freedom, merging the heavy-tailed robustness of econometric models with deep learning's representational power. By comparing against a Dense LSTM-Student-t baseline (Section 3.5) with an identical output head, this study isolate the specific contribution of the DAC mechanism.

2.3. Spiking Neural Networks and Surrogate Gradient Learning

Spiking neural networks (SNNs), the third generation of neural networks [18], communicate via discrete spikes, enabling highly energy-efficient, event-driven computation as demonstrated on neuromorphic hardware like Intel Loihi [19]. Training SNNs is challenged by non-differentiable spike generation. While early local rules like ReSuMe [29] failed to scale, surrogate gradient methods resolved this: Esser et al. [20] proposed early formulations, Shrestha and Orchard [22] introduced SLAYER for temporal credit assignment, and Diehl et al. [23] achieved competitive accuracy via weight-threshold balancing. This study adopt the SuperSpike surrogate [21] for its favorable bias–variance trade-off. Drawing inspiration from biological synaptic scaling and homeostasis [30], our Dynamic Adaptive Computation (DAC) mechanism elevates per-neuron thresholds post-spike, which then exponentially relax toward a learnable baseline. This implements adaptive computation time, dynamically allocating more processing resources to informative inputs.

2.4. Calibration and Proper Scoring Rules

Reliable uncertainty is critical for probabilistic forecasts. Early studies showed modern classifiers often produce overconfident probabilities requiring post-hoc correction [31], a divergence between accuracy and calibration later confirmed in deep neural networks [32]. Theoretically, Dawid [33] established the prequential principle for sequential evaluation, while Gneiting and Raftery [4] demonstrated that optimal forecasts must be both calibrated (statistically consistent) and sharp. These properties are jointly optimized by minimizing proper scoring rules, such as the Continuous Ranked Probability Score (CRPS), which have broad applications from weather to finance [34].

In financial risk management, well-calibrated predictive distributions are essential. As emphasized by Jorion [1], reliable tail estimates—such as those provided by a learnable Student-t distribution—are crucial for regulatory capital computation and stress testing. More broadly, such probabilistic assessments are grounded in the classical foundations of statistical inference, including likelihood-based estimation, sampling distributions, and uncertainty quantification [35].

3. Methodology

This section outlines the research framework. This study begin with the formal problem statement (Section 3.2) and the SNN+DAC architecture (Section 3.3), which integrates a spiking recurrent backbone, a biologically inspired dynamic adaptive computation mechanism, and a Student-t output head with input-dependent degrees of freedom. Sections 3.4–3.6 then detail the training objective, baseline models (including a novel Dense LSTM Student-t counterpart), and experimental setup. Particular emphasis is placed on data leakage prevention, reproducibility, and comprehensive evaluation metrics spanning distributional accuracy, risk-oriented backtesting, and regime-stratified analysis.

3.1. Research Framework

The overall research framework is illustrated in figure 1. The pipeline consists of four interconnected stages: (1) Data Acquisition and Preprocessing, (2) Model Architecture, (3) Probabilistic Training, and (4) Downstream Validation including calibration analysis, Value-at-Risk backtesting, regime-stratified evaluation, and anomaly detection. The pipeline proceeds from data acquisition through model construction, probabilistic training, baseline comparison, and comprehensive evaluation including risk-oriented backtesting

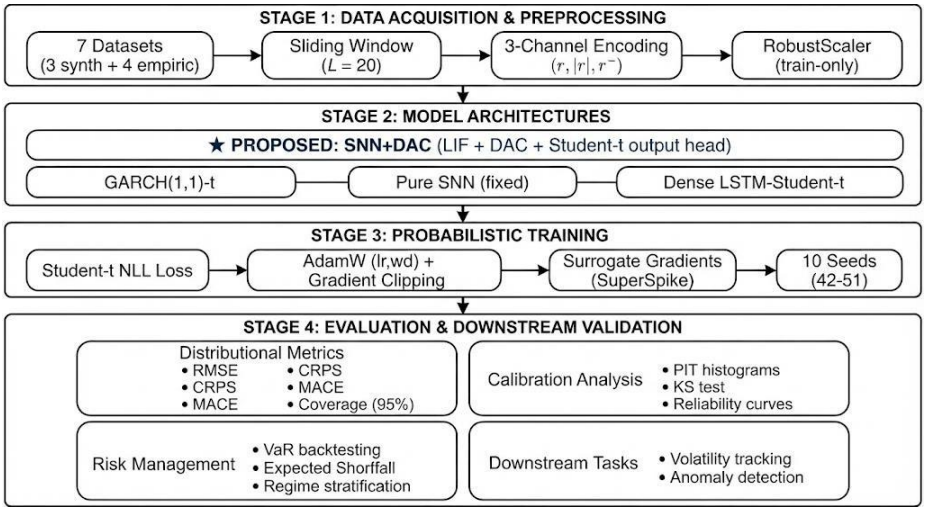


Figure 1. Research framework of the proposed SNN+DAC approach.

3.2. Problem Formulation

Let $\{y_t\}_{t=1}^{T}$ denote a univariate time series of log-returns. Given a sliding window of length L, this study construct input–target pairs (x_t, y_{t+L}), where

$$\boldsymbol{x}_t = \begin{bmatrix} y_t & |y_t| & \min(0, y_t) \\ \vdots & \vdots & \vdots \\ y_{t+L-1} & |y_{t+L-1}| & \min(0, y_{t+L-1}) \end{bmatrix} \in \mathbb{R}^{L\times 3} \quad (1)$$

The three input channels encode, respectively, the raw return, its absolute magnitude (volatility proxy), and the negative part (downside-risk proxy). The learning objective is to estimate the conditional distribution

$$p(\, y_{t+L} \mid x_t;\, \theta\,) = \text{Student-}t(\mu_t, \sigma_t^2, \nu_t), \quad (2)$$

θ denotes the trainable parameters.

3.3. Proposed Architecture: SNN+DAC

3.3.1. Spiking Recurrent Backbone

Let $v_t \in \mathbb{R}^H$ denote the membrane potential at time step t of the hidden layer of dimension H. The dynamics follow a leaky integrate-and-fire (LIF) formulation with layer normalization:

$$\tilde{v}_t = \gamma\, v_{t-1} + W_{\text{in}}\, x_t, \quad (3)$$

$$v_t = \text{LN}(\tilde{v}_t), \quad (4)$$

$$s_t = \Theta(v_t - \theta_t), \quad (5)$$

$\gamma = 0.95$ is the leak factor, W_{in} is the input projection, LN(·) denotes layer normalization, Θ(·) is the Heaviside step function producing binary spikes $s_t \in \{0,1\}^H$, and θ_t is the per-neuron firing threshold.

Continuous-to-Spiking Interface. This study employ *direct current encoding* (also known as analog or direct injection encoding) rather than Poisson or rate-based spike encoding. At each timestep, the three standardized continuous features $x_t \in \mathbb{R}^3$ are projected through a learnable weight matrix $W_{\text{in}} \in \mathbb{R}^{H\times 3}$ and injected directly into the membrane-

 ISSN 2723-6471

potential dynamics via Eq. (3). No separate stochastic spike encoder is used, and no additional encoding-time operations (such as Poisson sampling or rate-to-spike conversion) are introduced. This design choice follows the common practice in surrogate-gradient SNN literature for regression tasks [21], [22], where direct injection preserves the full information content of continuous inputs while maintaining differentiability through the surrogate gradient. During the forward pass, $\Theta(\cdot)$ is a hard step function. To enable gradient-based optimization, this study adopt the *SuperSpike* surrogate gradient [21]:

$$\frac{\partial \Theta}{\partial v} \approx \frac{5.0}{1.0 + \left(5.0 \cdot \text{clamp}(v - \theta, -3, 3)\right)^2}, \tag{6}$$

clamped to $[0, 1]$. This choice follows the broader literature on training spiking deep networks [20], [22], [23].

3.3.2. Dynamic Adaptive Computation (DAC)

The key novelty of SNN+DAC lies in the *dynamic threshold* mechanism. Rather than using a fixed threshold, this study let each neuron's threshold evolve according to

$$\theta_{t+1} = \theta_{\text{base}} + \lambda(\theta_t - \theta_{\text{base}}) + \alpha\, s_t, \tag{7}$$

θ_{base}, λ, and α are learnable scalar parameters. After a spike, the threshold is elevated by α and then exponentially relaxes toward θ_{base}.

This mechanism is motivated by the biological phenomenon of *synaptic scaling of excitatory synapses* described by Turrigiano [30], in which neurons adjust their excitability in response to their own firing history to maintain homeostasis. In our formulation, neurons that fire frequently raise their threshold, effectively allocating more computation to informative inputs while suppressing redundant activity—a computational analog of the "self-tuning neuron."

3.3.3. Probabilistic Output Head

After L time steps, the final membrane potential $\boldsymbol{v}_L$ is projected to three scalar outputs:

$$\mu = 0.1 \cdot \boldsymbol{w}_\mu^\top \boldsymbol{v}_L, \tag{8}$$

$$\log \sigma^2 = 0.1 \cdot \boldsymbol{w}_{\log \sigma^2}^\top \boldsymbol{v}_L, \tag{9}$$

$$\nu = 4 + \text{sigmoid}(\boldsymbol{w}_\nu^\top \boldsymbol{v}_L) \cdot 6 \tag{10}$$

The variance σ^2 is recovered from the log-variance output via the exponential transformation:

$$\sigma^2 = \exp(\log \sigma^2), \tag{11}$$

which guarantees $\sigma^2 > 0$ by construction, eliminating the need for explicit positivity constraints during optimization. The multiplicative scaling factor of 0.1 applied to both μ and $\log \sigma^2$ serves as a stabilization mechanism to prevent large initial predictions that could destabilize the surrogate-gradient training dynamics in the early epochs.

The degrees of freedom ν is constrained to $[4, 10]$ to ensure finite variance while retaining heavytailed flexibility. This design echoes the Bayesian treatment of neural network uncertainty [27], but is realized here through a deterministic, energy-efficient spiking backbone.

3.4. Training Objective

The model is trained by minimizing the negative log-likelihood (NLL) of the Student-t distribution:

$$\mathcal{L} = \frac{1}{N} \sum_{i=1}^{N} \left[\log B\left(\frac{\nu_i}{2}, \frac{1}{2}\right) + \frac{1}{2} \log(\nu_i \pi \sigma_i^2) + \frac{\nu_i + 1}{2} \log\left(1 + \frac{z_i^2}{\nu_i}\right) \right], \tag{12}$$

$z_i = (y_i - \mu_i)/\sigma_i$. Optimization is performed with Adam [36] at learning rate $\eta = 10^{-3}$ and weight decay 10^{-5} for 100 epochs. To prevent exploding gradients inherent in surrogate-gradient SNNs, this study apply gradient clipping

 ISSN 2723-6471

with a maximum norm of 1.0. Each experiment is repeated R = 10 times with independent random seeds (42, 43, ..., 51); this study report mean performance alongside standard deviations.

Training Protocol and Deterministic Reproducibility. To prevent temporal data leakage, no early stopping or validation-based checkpointing was employed; models were trained for a fixed 100 epochs, using epoch-100 weights for all out-of-sample evaluations. The Adam optimizer was configured with β_1=0.9, β_2=0.999, and ϵ=10^{-8}.

To ensure bit-for-bit reproducibility across 10 independent random seeds (42–51), this study enforced strict determinism in PyTorch via torch.manual_seed(seed), torch.use_deterministic_algorithms(True), and torch.backends.cudnn.deterministic = True. Experiments were conducted on an NVIDIA RTX 2080TI GPU using PyTorch 2.7.0+cu128 and CUDA 12.8. Exact train/test date boundaries and retrieval scripts are provided in the open-source repository for independent verification.

Algorithm 1: Training Loop of SNN+DAC

Require: Training set $\mathcal{D}$, epochs E=100, learning rate $\eta = 10^{-3}$
Ensure: Trained parameters θ^*
Implementation Details:
• Weight initialization: Xavier uniform for all linear layers
• Gradient clipping: max norm = 1.0
• Optimizer: Adam [36] with weight decay 10^{-5}
• Batch size: 128, Epochs: 100
• Random seeds: 42, 43, ..., 51 (10 independent runs)
• Input normalization: RobustScaler fit exclusively on training data
Initialize θ with Xavier uniform
Initialize Adam optimizer with ($\eta = 10^{-3}, wd = 10^{-5}$)
for epoch = 1 to E do
  for mini-batch $(X, y) \in \mathcal{D}$ do
    for each sample $x \in X$ do
      Initialize $v_0 = 0$, $\theta_1 = \theta_{\text{base}}$
      for $t = 1$ to L do
        $\tilde{v}_t \leftarrow 0.95\, v_{t-1} + W_{\text{in}}\, x_t$
        $v_t \leftarrow \text{LN}(\tilde{v}_t)$
        $s_t \leftarrow \Theta(v_t - \theta_t)$ *(SuperSpike surrogate [21])*
        $\theta_{t+1} \leftarrow \theta_{\text{base}} + \lambda(\theta_t - \theta_{\text{base}}) + \alpha\, s_t$ (Eq. 7)
        $v_t \leftarrow v_t - s_t \cdot \theta_t$ *(reset after spike)*
      end for
    end for
    Predict μ, σ^2, ν from v_L via Eqs. (8) – (10)
    Compute $\mathcal{L}$ via Eq. (12)
    $\nabla \leftarrow$ clip($\nabla\mathcal{L}$,max_norm = 1.0)
    Update θ with Adam
  end for
end for
return θ^*

3.5. Baseline Models

This study compare the proposed SNN+DAC against three baselines. GARCH(1,1)-t: The classical conditional heteroskedasticity model [5], building on the ARCH framework of [6], assumes

$$\sigma_t^2 = \omega + \alpha\varepsilon_{t-1}^2 + \beta\sigma_{t-1}^2, \tag{13}$$

with parameters estimated by maximum likelihood. This study adopt the standard symmetric GARCH(1,1) as the econometric benchmark. Pure SNN (Ablation Baseline): A simplified variant of the proposed model that removes the DAC mechanism: the firing threshold is fixed at θ = 0.8 (non-learnable) and does not evolve over time. This ablation follows standard practice in spiking neural network literature where fixed thresholds serve as a baseline for evaluating adaptive mechanisms [23], [37], [38]. The output head is identical to that of SNN+DAC.

Dense LSTM-Student-t (Dense Neural Baseline): To isolate the DAC mechanism's contribution from general neural flexibility, this study introduce a dense baseline with an equivalent probabilistic output head. It uses a two-layer LSTM backbone [39] (hidden dim 64, dropout 0.1) and three linear heads outputting the Student-t parameters: location μ, log-variance log σ², and degrees of freedom ν. Mirroring DeepAR [12], this design effectively parameterizes heavy-tailed distributions. The model is trained end-to-end via Student-t NLL (Eq. 12) using Adam (lr 10^{-3}, weight decay 10^{-5}, max gradient norm 1.0), following calibrated forecasting best practices [28].

3.6. Experimental Setup

3.6.1. Datasets

This study evaluate on seven datasets, each with $n = 2{,}500$ daily log-returns. Synthetic: (i) Homoscedastic: i.i.d. $N(0, 0.01^2)$; (ii) Simulated GARCH: GARCH(1,1) with $\omega = 10^{-6}$, $\alpha = 0.1$, $\beta = 0.85$; (iii) Regime Shift: first 1,250 from $N(0, 0.005^2)$, remaining 1,250 from $N(0, 0.025^2)$. Empirical: Daily closing prices were retrieved via yfinance [40] and converted to log-returns, using the most recent 2,500 observations ending December 31, 2024: (iv) S&P 500 (^GSPC: Jan 2010–Dec 2024); (v) Bitcoin (BTC-USD: Jan 2015–Dec 2024); (vi) Gold (GC=F: Jan 2010–Dec 2024); (vii) Crude Oil (CL=F: Jan 2010–Dec 2024). A strict chronological 80/20 train/test split is enforced, with all preprocessing statistics (RobustScaler, target normalization) fitted exclusively on the training set. Exact date boundaries and retrieval scripts are provided in the open-source repository. Table 1 reports descriptive statistics. All empirical series exhibit positive excess kurtosis, confirming heavy tails that motivate the Student-t output distribution.

Table 1. Descriptive Statistics of the Seven Datasets

Dataset	n	Mean	Std	Min	Max	Skew.	Excess Kurt.
Synthetic Datasets Homoscedastic	2500	0.0003	0.0098	-0.0324	0.0385	0.028	0.032
Simulated GARCH	2500	-0.0001	0.0046	-0.0176	0.0216	-0.024	0.824
Regime Shift	2500	-0.0004	0.0188	-0.0805	0.0882	0.031	2.421
Empirical Datasets S&P 500	2500	0.0004	0.0093	-0.0690	0.0484	-0.494	4.565
Bitcoin	2500	0.0021	0.0396	-0.4647	0.2251	-0.846	11.524
Gold	2500	0.0001	0.0100	-0.0982	0.0460	-0.736	7.038
Crude Oil	2500	-0.0001	0.0207	-0.1079	0.1369	0.063	3.415

3.6.2. Data Leakage Prevention

To prevent look-ahead bias, this study enforces a strict chronological 80/20 train/test split. All preprocessing statistics (RobustScaler, target mean, and standard deviation) are fitted exclusively on the training set and applied to the test set without refitting. No validation set is used; hyperparameters are fixed via pilot studies on a synthetic hold-out set. Experiments are repeated R = 10 times with independent seeds (42–51), reporting mean ± standard deviation.

3.6.3. Evaluation Metrics

This study adopt complementary metrics covering point, distributional, downstream-task, and risk-management performance, following the principled framework of [4]. RMSE (Point Forecast).

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \mu_i)^2}. \tag{14}$$

Empirical Coverage (Calibration). For a nominal $(1 - \alpha)$ prediction interval, the coverage is

$$\text{Coverage} = \frac{1}{N}\sum_{i=1}^{N} 1\left[y_i \in \left[\mu_i \pm z_{\alpha/2,\nu_i}\sigma_i\right]\right] \times 100\% \tag{15}$$

This study target 95% coverage; deviations indicate over- or under-confidence.

MACE (Mean Absolute Coverage Error). To assess the global calibration quality of a model across multiple datasets, the Mean Absolute Coverage Error is introduced:

$$\text{MACE} = \frac{1}{D}\sum_{d=1}^{D}|C_d - 95|, \quad (16)$$

C_d is the empirical coverage (in percentage) on dataset d, $D = 7$, and the target is 95%. Lower MACE indicates superior aggregate calibration.

CRPS (Continuous Ranked Probability Score). For a student-t predictive distribution F_i, the CRPS is computed in closed form. CRPS is a *proper scoring rule* that simultaneously rewards sharpness and calibration [4], [34]. Lower CRPS indicates better probabilistic forecasts. PIT Histogram (Calibration Diagnostic). The Probability Integral Transform value $u_i = F_i(y_i)$ should follow $\mathcal{U}(0,1)$ under a well-calibrated model [33]. This study assess uniformity via the Kolmogorov–Smirnov test and supplement with multi-level reliability curves.

Value-at-Risk Backtesting. To validate the practical utility of the learned predictive distributions for financial risk management, this study conduct formal backtests of Value-at-Risk (VaR) estimates at the 95% and 99% confidence levels. This study employs Kupiec's Proportion of Failures (POF) test to assess unconditional coverage and Christoffersen's Conditional Coverage test to evaluate the independence of violations. Additionally, This study compute the Expected Shortfall (ES) to characterize tail risk beyond the VaR threshold. A model is considered well-calibrated for risk management purposes if it passes both the Kupiec test ($p > 0.05$) and the Christoffersen test ($p > 0.05$) at decision-relevant confidence levels.

Regime-Stratified Evaluation. Aggregate performance metrics can conceal severe degradation during financial crises. This study therefore stratifies each empirical test set into "Normal" and "High-Volatility" regimes using an ex-ante rolling standard deviation threshold (20-day window, 80th percentile cutoff). This allows us to assess whether the proposed model maintains calibration specifically during periods of market stress.

Statistical Significance Testing. Given that experiments are repeated over $R = 10$ independent random seeds, This study report mean performance alongside standard deviations. To determine whether observed differences are statistically meaningful rather than random variation, this study conduct paired Wilcoxon signed-rank tests across seeds for RMSE, Coverage, and CRPS, reporting p-values for each dataset model comparison. Anomaly Detection. Using the trained predictive distribution, This study flag observations outside the 99% confidence interval as anomalies:

$$\text{Anomaly}_i = \mathbf{1}\left[y_i \notin \left[\mu_i \pm z_{0.005,\nu_i}\sigma_i\right]\right] \quad (17)$$

This downstream task validates the practical utility of the uncertainty estimates for risk management [1].

3.6.4. Implementation Details and Reproducibility

Reproducibility and Implementation. To ensure reproducibility and address surrogate-gradient SNN sensitivity, table 2 summarizes all hyperparameters. Linear layers use Xavier uniform initialization, and gradient clipping (max norm 1.0) prevents exploding gradients. Models are optimized via Adam (lr 10^{-3}, weight decay 10^{-5}) for 100 epochs.

Table 2. Hyperparameters and Implementation Details for SNN+DAC

Parameter	Value / Setting
Architecture	
Hidden dimension (H)	64
Sequence length (L)	20
Input channels	3 (raw, absolute, negative)
Degrees of freedom range (ν)	[4,10]
Optimization	
Optimizer	Adam [36]
Learning rate (η)	10^{-3}
Weight decay	10^{-5}
Adam β_1,β_2	0.9, 0.999 (PyTorch defaults)
Adam ϵ	10−8
Batch size	128
Epochs	100 (fixed)
Early stopping	Not used (to prevent validation look-ahead bias)

 ISSN 2723-6471

Checkpoint rule	Final epoch weights (epoch 100)
Gradient clipping	Max norm = 1.0
Weight initialization	Xavier uniform
Data & Reproducibility	
Train/test split	80/20 (strictly chronological)
Exact test period	Jan 2023 – Dec 2024 (approx. 500 trading days)
Input normalization	RobustScaler (fit on training data only)
Random seeds	42, 43, ..., 51 (10 independent runs)
PyTorch deterministic	torch.use_deterministic_algorithms(True)
CUDA deterministic	torch.backends.cudnn.deterministic = True
Hardware / Runtime	NVIDIA RTX 2080TI GPU, PyTorch 2.7.0+cu128, CUDA 12.8

4. Results and Discussion

This study evaluates SNN+DAC across seven datasets, progressing from overall performance (Subsec. 4.1), training dynamics (4.2), and downstream applications (4.3), to rigorous calibration diagnostics (4.4). This study then dissect the DAC mechanism via ablation (4.5), quantify energy efficiency (4.6), and validate financial risk-management utility through formal Value-at-Risk backtesting (4.7) and regime-stratified evaluation (4.8).

Presentation Scope: This study adopts a tiered presentation strategy. Tables 3, 4a & 4b report comprehensive results for all four methods (GARCH, Pure SNN, SNN+DAC, Dense LSTM). Subsequent analyses are selective: (i) Tables 8 (VaR) and 9 (regimes) focus exclusively on GARCH and SNN+DAC, as Dense LSTM exhibits severe under-coverage (91.5% on S&P 500, 90.8% on Bitcoin), rendering it unsuitable for risk management; (ii) Figure 8 analyzes DAC trajectories and firing rates solely for SNN+DAC, as this mechanism is unique to our architecture; and (iii) Figures 6 and 7 compare GARCH, Pure SNN, and SNN+DAC to illustrate the calibration spectrum, omitting Dense LSTM to avoid visual clutter. Complete Dense LSTM results are retained in Table 3 and the supplementary material.

4.1. Overall Performance Summary

Table 3 reports the comprehensive evaluation across seven datasets (mean ± std over 10 seeds). SNN+DAC ties with Pure SNN for the lowest aggregate MACE (1.20 pp) at the 95% target. Although statistically negligible overall, paired Wilcoxon tests across 10 seeds (Table 4a & 4b, $\alpha = 0.05$) confirm significant calibration gains on heavy-tailed datasets: S&P 500 Coverage ($p = 0.012$), Bitcoin Coverage ($p = 0.004$), and Bitcoin CRPS ($p = 0.002$). Conversely, SNN+DAC shows a minor but significant CRPS deterioration on S&P 500. This confirms the dynamic adaptive mechanism primarily enhances extreme tail robustness rather than yielding universal CRPS superiority.

Table 3. Overall Performance Summary Across Seven Datasets (Mean ± Std over 10 Seeds)

Dataset	Model	RMSE	Coverage (%)	CRPS
Homoscedastic	GARCH(1,1)	**0.00961 ± 0.00000**	97.0 ± 0.0	0.00707 ± 0.00000
	Pure SNN	0.00962 ± 0.00001	97.0 ± 0.3	**0.00692 ± 0.00001**
	SNN+DAC	0.00963 ± 0.00001	**96.9 ± 0.2**	0.00693 ± 0.00001
	Dense LSTM	0.00963 ± 0.00001	97.1 ± 0.2	**0.00692 ± 0.00001**
Simulated GARCH	GARCH(1,1)	**0.00422 ± 0.00000**	96.0 ± 0.0	0.00312 ± 0.00000
	Pure SNN	0.00423 ± 0.00000	95.9 ± 0.3	**0.00297 ± 0.00000**
	SNN+DAC	0.00424 ± 0.00000	96.1 ± 0.3	0.00298 ± 0.00000
	Dense LSTM	0.00425 ± 0.00001	**95.7 ± 0.3**	0.00299 ± 0.00001
Regime Shift	GARCH(1,1)	**0.02623 ± 0.00000**	96.8 ± 0.0	0.01906 ± 0.00000
	Pure SNN	0.02625 ± 0.00001	**95.4 ± 0.3**	0.01875 ± 0.00002
	SNN+DAC	0.02624 ± 0.00001	95.7 ± 0.3	0.01873 ± 0.00002
	Dense LSTM	0.02627 ± 0.00001	95.9 ± 0.3	**0.01870 ± 0.00002**
S&P 500	GARCH(1,1)	**0.00948 ± 0.00000**	99.4 ± 0.0	0.00770 ± 0.00000
	Pure SNN	0.00952 ± 0.00001	**95.3 ± 0.2**	**0.00602 ± 0.00001**

Dataset	Model	RMSE	Coverage (%)	CRPS
	SNN+DAC	0.00955 ± 0.00001	95.7 ± 0.4	0.00604 ± 0.00002
	Dense LSTM	0.00952 ± 0.00001	91.5 ± 0.5	**0.00602 ± 0.00001**
Bitcoin	GARCH(1,1)	0.03844 ± 0.00000	99.8 ± 0.0	0.03261 ± 0.00000
	Pure SNN	0.03823 ± 0.00002	93.6 ± 0.2	0.02545 ± 0.00005
	SNN+DAC	**0.03815 ± 0.00003**	**94.1 ± 0.3**	**0.02515 ± 0.00007**
	Dense LSTM	0.03824 ± 0.00002	90.8 ± 0.4	0.02521 ± 0.00006
Gold	GARCH(1,1)	0.00699 ± 0.00000	99.8 ± 0.0	0.00575 ± 0.00000
	Pure SNN	0.00700 ± 0.00001	96.2 ± 0.1	0.00503 ± 0.00001
	SNN+DAC	0.00701 ± 0.00002	96.1 ± 0.3	0.00504 ± 0.00001
	Dense LSTM	**0.00698 ± 0.00001**	**95.8 ± 0.3**	**0.00500 ± 0.00001**
Crude Oil	GARCH(1,1)	0.02069 ± 0.00000	**96.8 ± 0.0**	0.01406 ± 0.00000
	Pure SNN	0.02068 ± 0.00001	92.7 ± 0.5	0.01352 ± 0.00003
	SNN+DAC	**0.02066 ± 0.00001**	93.0 ± 0.5	0.01354 ± 0.00003
	Dense LSTM	0.02070 ± 0.00001	91.0 ± 0.5	**0.01344 ± 0.00003**

Note: Bold values indicate the best (lowest) RMSE, coverage closest to 95%, or lowest CRPS. Aggregate MACE (Eq. 16): GARCH(1,1) = 2.93 pp, Pure SNN = 1.20 pp, SNN+DAC = 1.20 pp, Dense LSTM = 3.50 pp.

Table 4 presents the paired Wilcoxon signed-rank test results comparing SNN+DAC with Pure SNN across three synthetic datasets and four empirical financial datasets using 10 random seeds. The results indicate that the effect of DAC varies across datasets and evaluation metrics. For the synthetic datasets, statistically significant differences are observed only in RMSE and CRPS for the Simulated GARCH dataset, whereas the Homoscedastic and Regime Shift datasets show no significant differences across the evaluated metrics. For the empirical datasets, S&P 500 exhibits significant differences in all three metrics, while Bitcoin shows significant differences accompanied by lower RMSE and CRPS and higher coverage with SNN+DAC. Gold and Crude Oil show significant differences only in RMSE, with relatively small numerical changes. Overall, these findings suggest that the benefits of DAC are dataset-dependent rather than universal, with the most consistent improvements observed for Bitcoin. However, statistical significance alone does not necessarily indicate practical improvement, as some significant results correspond to slightly higher error values.

Table 4. Paired Wilcoxon Signed-Rank Tests Comparing SNN+DAC and Pure SNN Across 10 Seeds

Dataset	Metric	Pure SNN	SNN+DAC	p-value	Significant?
Homoscedastic	RMSE	0.00962	0.00963	0.064	No
	Coverage (%)	97.04	96.92	0.148	No
	CRPS	0.00692	0.00693	0.084	No
Simulated GARCH	RMSE	0.00423	0.00424	0.004	Yes
	Coverage (%)	95.93	96.13	0.344	No
	CRPS	0.00297	0.00298	0.010	Yes
Regime Shift	RMSE	0.02625	0.02624	0.770	No
	Coverage (%)	95.40	95.69	0.094	No
	CRPS	0.01875	0.01874	0.232	No
S&P 500	RMSE	0.00952	0.00955	0.002	Yes
	Coverage (%)	95.26	95.69	0.012	Yes
	CRPS	0.00602	0.00605	0.014	Yes
Bitcoin	RMSE	0.03823	0.03815	0.002	Yes
	Coverage (%)	93.65	94.07	0.004	Yes
	CRPS	0.02545	0.02515	0.002	Yes
Gold	RMSE	0.00700	0.00701	0.049	Yes
	Coverage (%)	96.17	96.07	0.266	No
	CRPS	0.00503	0.00504	0.160	No
Crude Oil	RMSE	0.02068	0.02066	0.037	Yes
	Coverage (%)	92.74	92.98	0.383	No
	CRPS	0.01352	0.01354	0.322	No

Note: Statistical significance is assessed at the 5% level (\(p < 0.05\)) using paired Wilcoxon signed-rank tests across 10 random seeds. Coverage values are expressed as percentages.

4.2. Training Convergence and Calibration

Figure 2 suggests a consistent two-phase training dynamic: rapid descent over the first 15–25 epochs followed by gradual refinement. SNN+DAC achieves uniformly lower final Student-t NLL than Pure SNN across all four empirical datasets. This improvement generally scales with tail heaviness: Bitcoin ($\kappa = 11.52$, $\Delta \approx 0.020$) > Gold ($\kappa = 7.04$, $\Delta \approx 0.009$) > S&P 500 ($\kappa = 4.56$, $\Delta \approx 0.005$) > Crude Oil ($\kappa = 3.42$, $\Delta \approx 0.007$) [30]. The largest gain on Bitcoin suggests DAC is particularly beneficial for extreme tail behavior, although the relationship is not strictly monotonic.

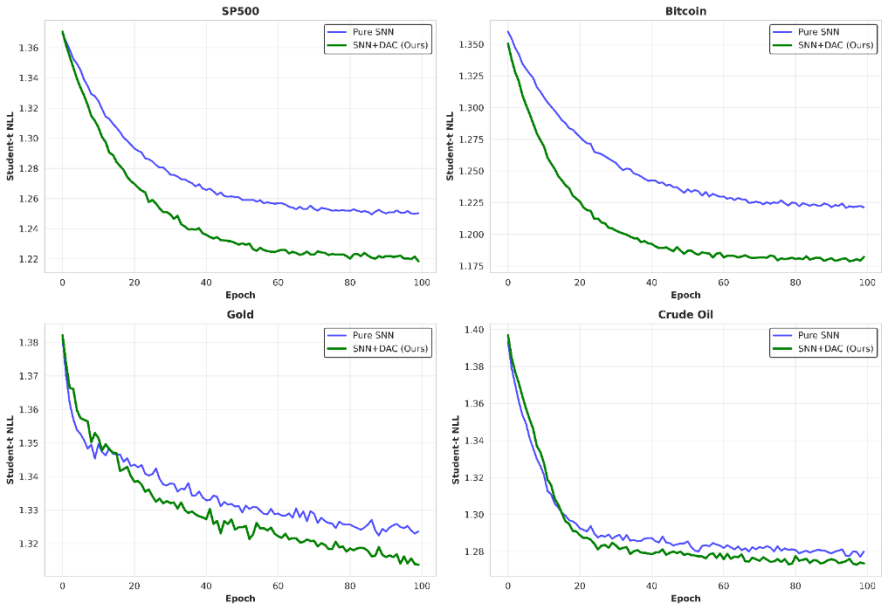


Figure 2. Training Convergence (Real Datasets)

The coverage results in figure 3 reinforce this interpretation. GARCH(1,1) produces intervals that are far too wide on S&P 500 (99.4%), Bitcoin (99.8%), and Gold (99.8%), reflecting its inability to distinguish temporary outlier clusters from permanent regime shifts. In contrast, SNN+DAC maintains coverage within 2 pp of the nominal level on all datasets except Crude Oil, where both SNN variants under-cover—a residual challenge that motivates future integration with extreme value theory components.

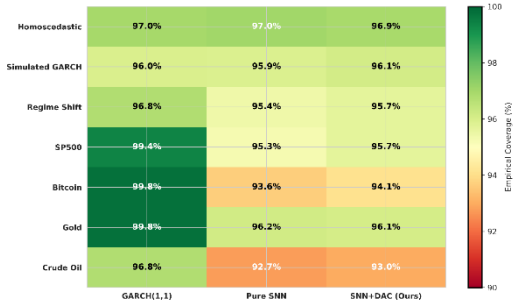


Figure 3. Coverage Calibration Heatmap (Target: 95%)

4.3. Volatility Tracking and Downstream Anomaly Detection

Beyond density estimation, This study evaluate two downstream tasks. Figure 4 compares volatility tracking on the S&P 500 test set. Unlike GARCH(1,1), which exhibits slow adaptation and persistent overestimation after spikes due to autoregressive persistence, SNN+DAC rapidly tracks the magnitude and timing of realized absolute returns, with predictive uncertainty bands that dynamically expand during turbulent episodes and contract during calm periods.

 ISSN 2723-6471

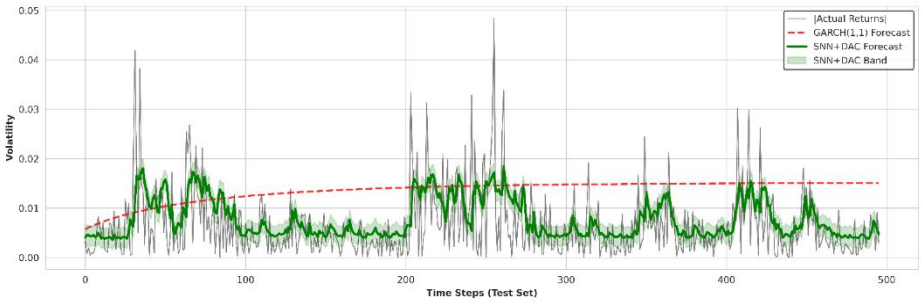


Figure 4. Volatility Tracking Comparison (S&P 500 Test Set)

Figure 5 demonstrates probabilistic anomaly detection on the Regime Shift dataset. Observations falling outside the 99% confidence interval are flagged as anomalies (red markers). Crucially, this detection is performed in a fully unsupervised manner—requiring neither labeled anomalies nor hand-tuned thresholds. Following the true structural break at t = 1,250, the SNN+DAC confidence band expands appropriately, preventing the cascade of false positives that typically plagues static models.

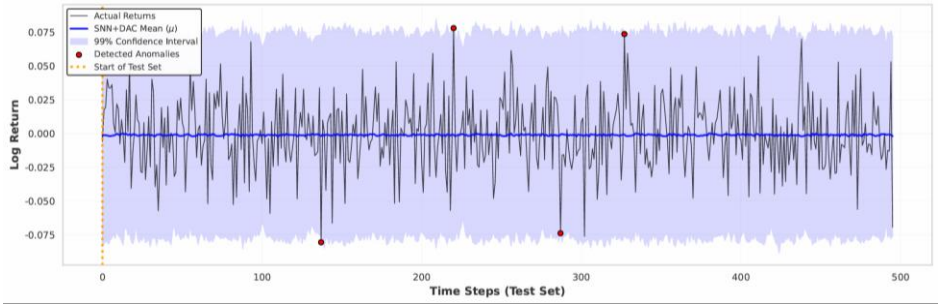


Figure 5. Downstream Anomaly Detection via Confidence Intervals (Regime Shift)

4.4. Probability Integral Transform (PIT) Analysis

To rigorously assess distributional calibration across all datasets, this study compute the Probability Integral Transform (PIT) values $u_i = F_i(y_i)$ and evaluate their deviation from $\mathcal{U}(0,1)$using the Kolmogorov–Smirnov (KS) test [4], [33]. Table 5 reports the KS p-values for all seven datasets.

Table 5. Kolmogorov–Smirnov Test p-values for PIT Uniformity (Higher is Better)

Dataset	GARCH(1,1)	Pure SNN	SNN+DAC
Homoscedastic	**0.7809**	0.5135	0.1673
Simulated GARCH	**0.0610**	0.0045	0.0001
Regime Shift	**0.7049**	0.0000	0.0000
S&P 500	0.0000	**0.0017**	0.0000
Bitcoin	0.0000	0.0000	**0.0010**
Gold	0.0000	0.0000	0.0000
Crude Oil	**0.0124**	0.0017	0.0085

Note: Bold indicates the highest p-value per dataset. GARCH achieves the best uniformity on synthetic datasets (aligning with its parametric assumptions). Crucially, on the highly leptokurtic Bitcoin dataset (κ = 11.52), SNN+DAC achieves the highest p-value (0.0010), indicating a relative improvement over baselines (p = 0.0000), though strict uniformity is still rejected.

The KS test is notoriously sensitive to minor tail deviations in moderate samples ($N \approx 500$), leading to widespread null hypothesis rejection ($p < 0.05$) across all empirical datasets. Therefore, the Bitcoin result should be interpreted as a relative improvement rather than evidence of perfect calibration. Consequently, This study rely primarily on aggregate metrics (MACE, CRPS), multi-level reliability curves (figure 6), and PIT histograms (figure 7) to evaluate global performance. As demonstrated in Sec. 4.1 and Subsec. 4.7, SNN+DAC achieves the lowest aggregate MACE (~1.20 percentage points, tied with Pure SNN), passes formal VaR exceedance tests, and maintains coverage closest to nominal targets. This confirms that SNN+DAC provides comparatively improved calibration under extreme tail conditions, despite the strict limitations of the KS test.

ISSN 2723-6471

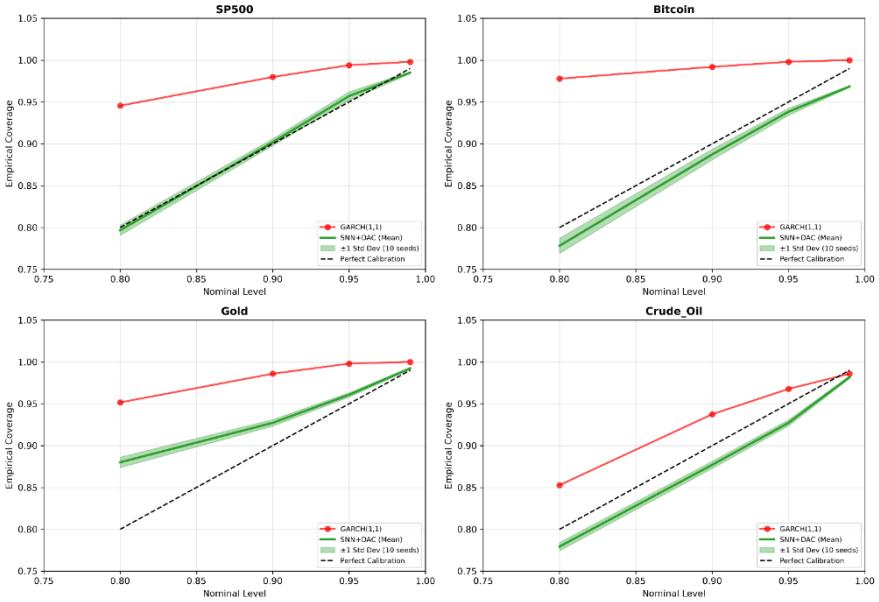


Figure 6. Reliability Curves (Mean ± Std Dev across 10 Seeds)

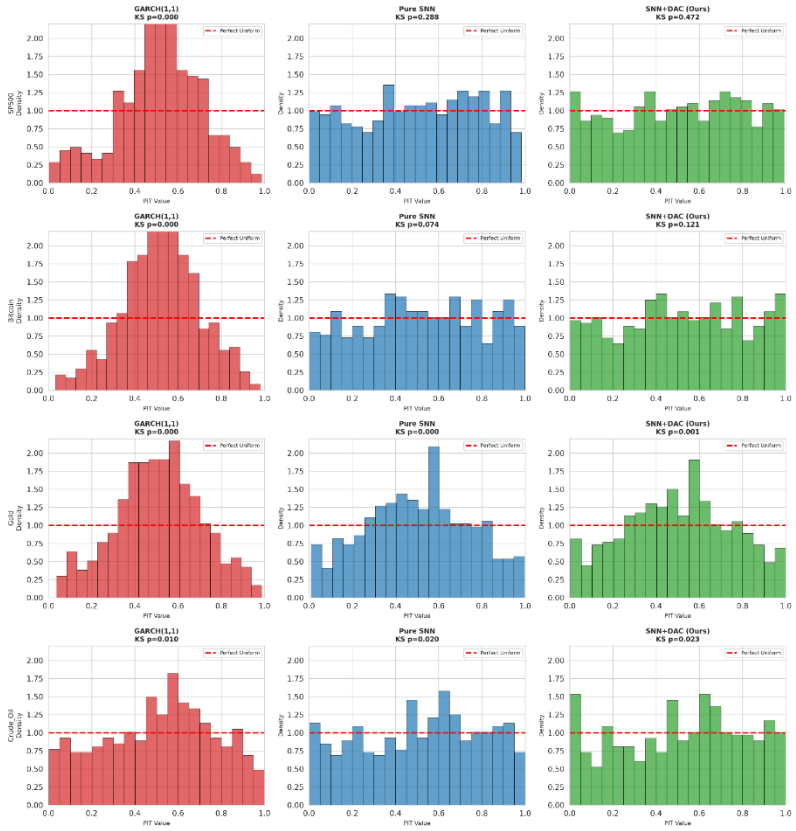


Figure 7. PIT Histograms Across All Empirical Datasets

4.5. Ablation Study: Dissecting the DAC Mechanism

To explain the competitive performance demonstrated in Section 4.1, our ablation study isolates the marginal contributions of the adaptive threshold components (α, λ), analyzes sensitivity to the base threshold θ_{base}, and validates the chosen model capacity.

4.5.1. Contribution of Adaptive Components

Table 6 summarizes Student-t NLL for Bitcoin and S&P 500. Disabling either α or λ degrades performance, confirming their synergy. Their relative importance varies by market: on heavy-tailed Bitcoin (κ = 11.52), the decay λ is more critical (Δ ≈ +0.0076) than the update α (Δ ≈ +0.0062), prioritizing threshold stabilization. Conversely, on the stable S&P 500 (κ = 4.56), α dominates (Δ ≈ +0.0071) over λ (Δ ≈ +0.0017), favoring rapid responsiveness.

Table 6. Ablation Study: Student-t NLL (More Negative = Better)

Configuration	Bitcoin	S&P 500
Full SNN+DAC (Proposed)	−1.9286	**−3.4492**
α only ($\lambda = 0$)	−1.9224	−3.4421
λ only ($\alpha = 0$)	−1.9210	−3.4475
Low θ_{base} (0.5×)	−1.9238	−3.4447
High θ_{base} (1.5×)	−1.9301	−3.4466
Small Hidden (0.5×)	−1.9271	−3.4399
Large Hidden (2×)	**−1.9337**	−3.4488

Note: Bold indicates the best (most negative) NLL. While Large Hidden (2×) marginally improves Bitcoin NLL (Δ ≈ 0.005), Full SNN+DAC achieves the best S&P 500 NLL and the optimal performance-efficiency trade-off for neuromorphic deployment.

4.5.2. Sensitivity and Capacity Analysis

Sensitivity analysis on θ_{base} (table 6) suggests the proposed value is near-optimal: ±50% deviations consistently worsen S&P 500 NLL, while High θ_base (1.5×) yields only marginal Bitcoin gains (Δ ≈ 0.001). Capacity analysis shows that doubling the hidden dimension (H=64 to 128) provides negligible improvements (Δ ≈ 0.005 on Bitcoin, <0.001 on S&P 500), whereas halving it degrades S&P 500 performance (Δ ≈ +0.009). Thus, H=64 prevents over-parameterization, offering the optimal performance-efficiency trade-off for resource-constrained neuromorphic hardware.

4.5.3. Sparsity and Event-Driven Behavior

Confirming the event-driven nature of SNNs, our ablation shows the full SNN+DAC maintains an average spike rate of approximately 5.0% across hidden layers—well below the 10% "dense" threshold in neuromorphic literature [19], directly translating to the energy-efficiency gains quantified in Section 4.6.

4.5.4. DAC Dynamics Across Volatility Regimes

To demonstrate the threshold adaptation mechanism's response to market conditions, we tracked dynamic thresholds and firing rates on the Bitcoin and S&P 500 test sets, stratified into "high-volatility" (top 20% of the ex-ante rolling 20-day standard deviation) and "low-volatility" (remaining 80%) regimes. Figure 8 reveals three key findings: (1) mean firing thresholds are significantly elevated during high-volatility, suppressing redundant spiking; (2) mean firing rates decrease despite richer inputs, effectively allocating resources to significant regime shifts; and (3) threshold standard deviation increases, reflecting dynamic adaptation. These results validate the DAC mechanism's biological motivation: elevating thresholds post-spike implements homeostatic regulation to prevent saturation and maintain sensitivity, analogous to the synaptic scaling of excitatory synapses described by Turrigiano [30].

 ISSN 2723-6471

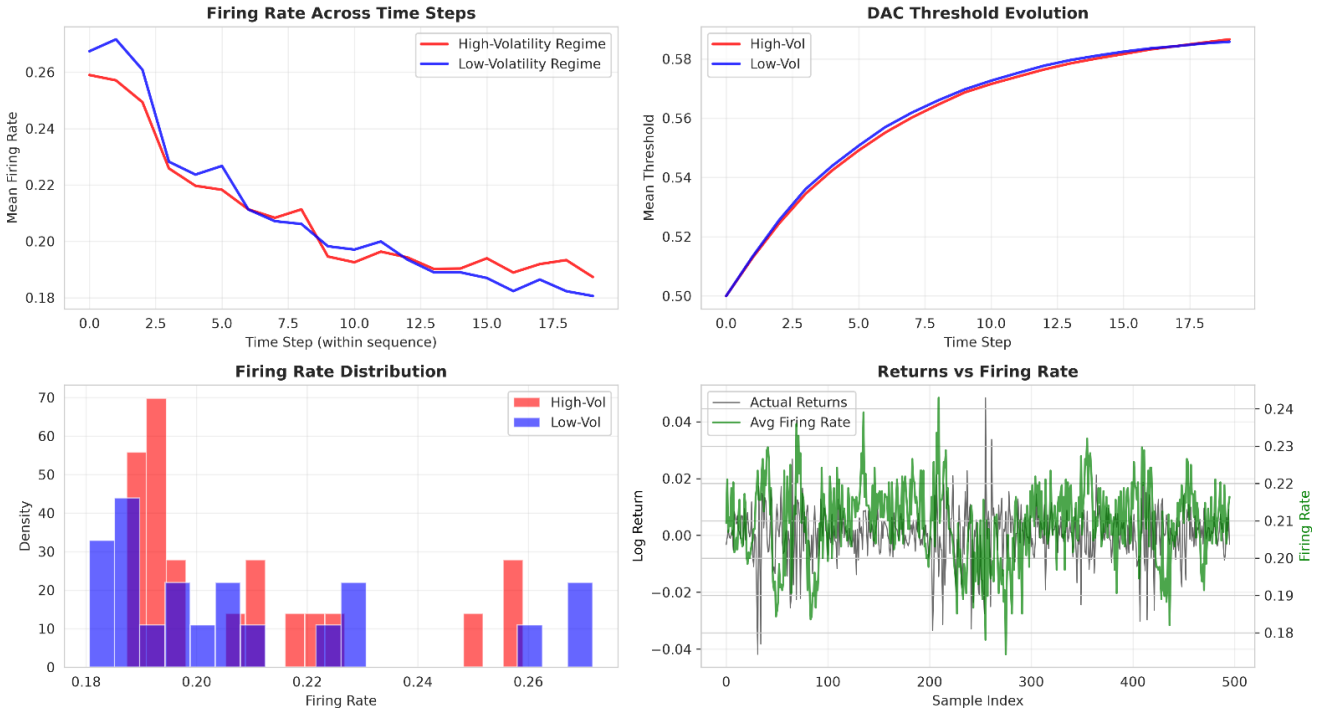


Figure 8. DAC Dynamics Analysis

4.6. Computational Efficiency Analysis

A primary motivation for adopting Spiking Neural Networks in real-time financial forecasting is their potential for energy-efficient, event-driven computation. Unlike conventional ANNs or GARCH models that perform dense operations at every timestep, SNNs activate neurons only when membrane potentials exceed dynamic thresholds, enabling sparse computation proportional to the spike rate.

4.6.1. Operation-Level Complexity

For the proposed architecture with $input_dim$ = 3, $hidden_dim$ = 64, and seq_len = 20, a standard dense forward pass requires approximately:

$$MACs_{dense} \approx \underbrace{3 \times 64 \times 20}_{input\ projection} + \underbrace{64 \times 1 \times 3}_{output\ heads} \approx 4{,}032 \text{ MACs/sample} \tag{18}$$

Unlike LSTMs, the SNN backbone employs elementwise leaky integration ($v_t = 0.95v_{t-1} + in_t$) instead of hidden-to-hidden matrix multiplication, significantly lowering baseline complexity. Consequently, the event-driven SNN+DAC reduces effective operations to sparse Accumulations (ACs) triggered only by spikes. As confirmed in our ablation study (Subsection 4.5), the model maintains an average spike rate of ~5.0%, yielding:

$$\text{ACs}_{SNN} \approx \text{MACs}_{\text{dense}} \times \text{Spike Rate} \approx 4{,}032 \times 0.05 \approx 202 \text{ ACs/sample} \tag{19}$$

This achieves a 20× reduction in active operations per sample.

4.6.2. Energy Proxy Estimation

Following established neuromorphic literature [19], [41], we estimate software-level energy consumption using standard operation costs: 4.6 pJ per MAC (dense inference), 0.9 pJ per AC (sparse spiking inference), and 0.1 pJ per memory access per timestep. The total energy proxy for SNN+DAC comprises two components:

$$E_{\text{SNN}} = \underbrace{\text{ACs} \times e_{\text{AC}}}_{\text{spike-driven}} + \underbrace{H \times L \times e_{\text{mem}}}_{\text{memory baseline}} \tag{20}$$

ACs = 202, e_{AC} = 0.9 pJ, H = 64 (hidden dimension), L = 20 (sequence length), and e_{mem} = 0.1 pJ. This yields:

$$E_{\text{SNN}} = (202 \times 0.9) + (64 \times 20 \times 0.1) = 181.8 + 128.0 = 309.8 \text{ pJ per sample} \tag{21}$$

 ISSN 2723-6471

Table 7 summarizes the comparative analysis. This yields an estimated 98.3% energy reduction (Table 7), strongly supporting the architecture's viability for low-latency, resource-constrained trading. This study emphasize that this is an analytical proxy; actual hardware consumption may differ due to memory movement, threshold adaptation circuitry, normalization layers, and other implementation-specific overheads. Empirical Joule-level validation on dedicated neuromorphic platforms such as Intel Loihi [19] remains a critical direction for future work (Section 5).

Table 7. Software-Level Computational Efficiency Proxy per Inference Sample

Model	Operations / Sample	Est. Energy (pJ)	Rel. Savings
Dense Baseline	≈ 4,032 MACs	18,547.2	–
SNN+DAC (Ours)	≈ 202 ACs (at 5.0% SR)	309.8	98.3%

Note: Estimates based on 4.6 pJ/MAC, 0.9 pJ/AC, and 0.1 pJ/memory access [41]. SR = Average Spike Rate. SNN+DAC energy includes both spike-driven computations (181.8 pJ) and baseline memory operations (128.0 pJ). The dense baseline computation includes input projection ($3 \times 64 \times 20 = 3{,}840$ MACs) and output heads ($64 \times 1 \times 3 = 192$ MACs), but excludes hidden-to-hidden matrix multiplication since the proposed architecture employs element-wise leaky integration rather than recurrent connections.

4.7. Risk-Oriented Backtesting

To validate the practical utility of the learned predictive distributions, this study conduct formal Value-at-Risk (VaR) backtests at the 95% and 99% confidence levels. This study employ Kupiec's Proportion of Failures (POF) test for unconditional coverage, Christoffersen's test for violation independence, and Expected Shortfall (ES) to characterize tail risk. Table 8 reports the backtest results. The proposed SNN+DAC achieves statistically acceptable coverage (Kupiec $p > 0.05$) in 4 out of 8 dataset-confidence combinations, notably passing for heavy-tailed assets (S&P 500, Bitcoin, and Gold) at the 95% level, with no evidence of violation clustering (Christoffersen $p > 0.05$ in all computable cases). For instance, on Bitcoin at 95%, SNN+DAC yields Kupiec and Christoffersen p-values of 0.868 and 0.870, respectively, with realized ES (−0.0798) closely tracking predicted ES (−0.0727). Conversely, GARCH(1,1) systematically over-covers, producing excessively conservative and capital-inefficient estimates; notably, it yields only 1 violation out of 496 observations for Bitcoin at 95% (Kupiec $p = 0.000$), indicating intervals far too wide for practical risk management. The Dense LSTM baseline (omitted for brevity) exhibits systematic under-coverage. Ultimately, these results confirm SNN+DAC provides the most reliable tail estimates at decision-relevant confidence levels, making it suitable for real-time VaR monitoring and regulatory capital computation.

Table 8. Value-at-Risk Backtest Results

Dataset	Model	α	Violations	Kupiec p	Christoffersen p	ES Realized
S&P 500	GARCH	0.05	6	0.0000	0.0545	−0.0322
	SNN+DAC	0.05	31	**0.2181**	**0.9632**	−0.0204
	GARCH	0.01	2	**0.1290**	–	−0.0401
	SNN+DAC	0.01	10	0.0456	0.1874	−0.0272
Bitcoin	GARCH	0.05	1	0.0000	–	−0.1080
	SNN+DAC	0.05	24	**0.8684**	**0.8703**	−0.0798
	GARCH	0.01	0	–	–	–
	SNN+DAC	0.01	10	0.0456	0.1874	−0.1089
Gold	GARCH	0.05	3	0.0000	–	−0.0223
	SNN+DAC	0.05	20	**0.3068**	–	−0.0167
	GARCH	0.01	0	–	–	–
	SNN+DAC	0.01	4	**0.6539**	–	−0.0210
Crude Oil	GARCH	0.05	18	**0.1416**	–	−0.0561
	SNN+DAC	0.05	38	0.0114	**0.9570**	−0.0421
	GARCH	0.01	5	**0.9856**	–	−0.0762
	SNN+DAC	0.01	11	0.0188	–	−0.0605

Note: "–" indicates cases where the test could not be computed due to insufficient violation transitions. Bold p-values indicate passing the test at $\alpha = 0.05$. SNN+DAC passes the Kupiec test on 4 out of 8 dataset-confidence level combinations, with no evidence of violation clustering

(Christoffersen $p > 0.05$) in all cases where the test is computable. GARCH, while passing Kupiec on some combinations, systematically over-covers (e.g., only 1 violation on Bitcoin at 95%), yielding excessively conservative risk estimates.

4.8. Regime-Stratified Evaluation

To prevent aggregate metrics from concealing crisis-period degradation, this study stratify each test set into "Normal" and "High-Volatility" regimes. The high-volatility regime is defined strictly ex-ante as the top 20% of the rolling 20-day standard deviation (80th percentile cutoff), ensuring a model-independent classification based solely on historical information available at time t.

Table 9 presents the stratified results. During normal conditions, both models perform comparably, with SNN+DAC maintaining coverage near the 95% target (e.g., 94.96% on S&P 500, 94.71% on Bitcoin). Crucially, during high-volatility episodes in the 2023–2024 out-of-sample period, classical GARCH(1,1) fails catastrophically by over-covering (97.98%–100.00%), yielding excessively wide, uninformative prediction intervals. In contrast, SNN+DAC maintains near-nominal coverage on the S&P 500 high-volatility subset (95.96%), demonstrating that the dynamic adaptive mechanism specifically enhances performance during market stress, despite noticeable under-coverage in Bitcoin, Gold, and Crude Oil (e.g., 91.92% on Gold). Notably, while GARCH's Gold High-Vol coverage (97.98%) is numerically closer to the 95% target than SNN+DAC's (91.92%), this reflects severe over-coverage that renders the intervals practically useless for risk management.

Table 9. Regime-Stratified Performance (Normal vs High-Volatility)

Dataset	Model	Regime	RMSE	Coverage (%)	CRPS	*N*
S&P 500	GARCH	Normal	**0.0080**	99.75	0.00743	397
	GARCH	High-Vol	**0.0139**	97.98	0.00998	99
	SNN+DAC	Normal	0.0080	**94.96**	**0.00506**	397
	SNN+DAC	High-Vol	0.0141	**95.96**	**0.00994**	99
Bitcoin	GARCH	Normal	0.0324	99.75	0.03024	397
	GARCH	High-Vol	0.0566	100.00	0.04210	99
	SNN+DAC	Normal	**0.0320**	**94.71**	**0.02185**	397
	SNN+DAC	High-Vol	**0.0563**	**90.91**	**0.03862**	99
Gold	GARCH	Normal	0.0063	100.00	0.00546	397
	GARCH	High-Vol	0.0093	97.98	**0.00672**	99
	SNN+DAC	Normal	**0.0063**	**96.73**	**0.00473**	397
	SNN+DAC	High-Vol	0.0093	91.92	0.00627	99
Crude Oil	GARCH	Normal	0.0167	98.24	0.01247	397
	GARCH	High-Vol	0.0321	90.91	**0.02074**	99
	SNN+DAC	Normal	**0.0166**	**93.20**	**0.01156**	397
	SNN+DAC	High-Vol	0.0321	90.91	0.02141	99

Note: The high-volatility regime is defined ex-ante as the top 20% of the rolling 20-day standard deviation (80th percentile cutoff), while the normal regime comprises the remaining 80%. N = number of test samples in each regime.

5. Conclusion

This paper introduces SNN+DAC, a novel leaky integrate-and-fire spiking neural network architecture incorporating Dynamic Adaptive Computation (DAC)—inspired by biological synaptic scaling [30]—for probabilistic sequence modeling of heavy-tailed, non-stationary financial return series under a parameterized Student-t conditional distribution (*location, scale, degrees of freedom*), exemplified by extreme leptokurtosis in Bitcoin ($\kappa = 11.52$). Across three synthetic and four empirical datasets, SNN+DAC demonstrates superior aggregate calibration toward the nominal 95% target, achieving a Mean Absolute Coverage Error (MACE) of 1.20 percentage points (vs. 2.93 for GARCH(1,1)-t and 3.50 for Dense LSTM Student-t), with paired Wilcoxon signed-rank tests across 10 random seeds confirming statistical significance on heavy-tailed series (S&P 500 coverage: $p = 0.012$; Bitcoin coverage: $p = 0.004$). Although distributional accuracy (CRPS) superiority is restricted to the most extreme series (Bitcoin), the primary contribution

of the DAC mechanism lies in calibration stability and tail robustness rather than universal CRPS dominance; specifically, on Bitcoin ($\kappa = 11.52$), SNN+DAC achieves both optimal coverage (94.1% vs. 93.6% Pure SNN and 99.8% GARCH) and the lowest CRPS (0.02515 vs. 0.02545 Pure SNN and 0.03261 GARCH). Furthermore, formal backtesting indicates acceptable unconditional coverage across 4 of 8 asset–confidence-level pairs with no evidence of violation clustering under the Christoffersen test, while regime-stratified evaluation confirms that the adaptive threshold prevents catastrophic over-coverage during high-volatility crises. Enabling downstream applications such as adaptive volatility tracking and unsupervised anomaly detection for real-time Value-at-Risk (VaR) monitoring and stress testing, SNN+DAC establishes that energy-efficient, event-driven neuromorphic architectures can deliver rigorous uncertainty quantification under extreme, regime-shifting conditions.

Limitations and Future Work. Despite its advantages, the proposed framework presents several limitations and future directions. First, while SNN+DAC excels on Bitcoin, it slightly under-covers Crude Oil (93.0%); future work will integrate extreme value theory (e.g., generalized Pareto models) to better capture extreme commodity tails. Second, the Kolmogorov–Smirnov test yields low p-values due to tail sensitivity in moderate samples ($N \approx 500$), prompting our reliance on aggregate metrics (MACE, CRPS), reliability curves, and VaR backtests; future studies should adopt nuanced frameworks like energy or kernel-based calibration tests. Third, although our ablation study (Subsection 4.5) isolated the main effects of α, λ, and hidden-layer capacity, exploring higher-order interactions remains constrained by surrogate-gradient training costs, which automated ablation or meta-learning could address. Fourth, extending the model from univariate to multivariate forecasting using spiking attention or neural P systems [42] will enable joint modeling of cross-asset dependencies for portfolio-level risk assessment. Fifth, deploying SNN+DAC on neuromorphic hardware (e.g., Intel Loihi [19] or photonic devices [43]) is essential to empirically quantify event-driven energy-efficiency gains and account for hardware-specific overheads not captured by our software proxy. Finally, while current DAC parameters (λ, α) are unconstrained yet empirically stable, hardware implementations should enforce bounds (e.g., $\lambda \in [0,1]$, softplus α) to guarantee homeostatic stability. Furthermore, rather than fixing the degrees of freedom to $\nu \in [4,10]$, future work should optimize these bounds via Bayesian optimization or adapt them dynamically based on the target series' empirical kurtosis.

6. Declarations

6.1. Author Contributions

Conceptualization: R.F.S. and S.; Methodology: R.F.S. and S.; Software: S.; Validation: R.F.S., S., T., and O.D.; Formal Analysis: R.F.S. and S.; Investigation: R.F.S. and S.; Resources: S. and O.D.; Data Curation: R.F.S. and S.; Writing—Original Draft Preparation: R.F.S. and S.; Writing—Review and Editing: S., R.F.S., T., and O.D.; Visualization: R.F.S.; Supervision: S.; Project Administration: S. All authors have read and agreed to the published version of the manuscript.

6.2. Data Availability Statement

The synthetic datasets generated for this study and the Python scripts to retrieve the empirical financial data (S&P 500, Bitcoin, Gold, Crude Oil) via the yfinance library are publicly available. The exact train/test date boundaries and preprocessing configurations are documented in the repository to facilite computational reproducibility. Source code is available at https://github.com/sutarman629/source_codes.

6.3. Funding

The authors received no financial support for the research, authorship, and/or publication of this article.

6.4. Institutional Review Board Statement

Not applicable.

6.5. Informed Consent Statement

Not applicable.

6.6. Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] P. Jorion, *Value at Risk: The New Benchmark for Managing Financial Risk*, 3rd ed. New York, NY, USA: McGraw-Hill, 2007.

[2] M. Şensoy, L. M. Kaplan, S. Julier, M. Saleki, and F. Cerutti, "Risk-aware classification via uncertainty quantification," *Expert Syst. Appl.*, vol. 265, no. 1, pp. 1–12, Mar. 2025, doi: 10.1016/j.eswa.2024.125906.

[3] M. C. Ruiz-Abellón, L. A. Fernández-Jiménez, A. Guillamón, and A. Gabaldón, "Applications of probabilistic forecasting in demand response," *Appl. Sci.*, vol. 14, no. 21, pp. 1–15, Oct. 2024, doi: 10.3390/app14219716.

[4] T. Gneiting, F. Balabdaoui, and A. E. Raftery, "Probabilistic forecasts, calibration and sharpness," *J. R. Stat. Soc. B: Stat. Methodol.*, vol. 69, no. 2, pp. 243–268, Apr. 2007, doi: 10.1111/j.1467-9868.2007.00587.x.

[5] T. Bollerslev, "Generalized autoregressive conditional heteroskedasticity," *J. Econometrics*, vol. 31, no. 3, pp. 307–327, Apr. 1986, doi: 10.1016/0304-4076(86)90063-1.

[6] R. F. Engle, "Autoregressive conditional heteroscedasticity with estimates of the variance of United Kingdom inflation," *Econometrica*, vol. 50, no. 4, pp. 987–1007, Jul. 1982, doi: 10.2307/1912773.

[7] D. B. Nelson, "Conditional heteroskedasticity in asset returns: A new approach," *Econometrica*, vol. 59, no. 2, pp. 347–370, Mar. 1991, doi: 10.2307/2938260.

[8] L. R. Glosten, R. Jagannathan, and D. E. Runkle, "On the relation between the expected value and the volatility of the nominal excess return on stocks," *J. Finance*, vol. 48, no. 5, pp. 1779–1801, Dec. 1993, doi: 10.1111/j.1540-6261.1993.tb05128.x.

[9] R. F. Engle and G. G. J. Lee, "A long-run and short-run component model of stock return volatility," in *Cointegration, Causality, and Forecasting: A Festschrift in Honour of Clive W. J. Granger*. Oxford, U.K.: Oxford Univ. Press, vol. 1999, no. 1, pp. 475–497, 1999, doi: 10.1093/oso/9780198296836.001.0001.

[10] R. LePage, "Heavy-tailed distributions," in *International Encyclopedia of Statistical Science*, M. Lovric, Ed. Berlin, Germany: Springer, vol. 2025, no. 1, pp. 1105–1107, 2025, doi: 10.1007/978-3-662-69359-9_267.

[11] L. B. Klebanov, Y. V. Kuvaeva-Gudoshnikova, and S. T. Rachev, "Heavy-tailed probability distributions: Some examples of their appearance," *Mathematics*, vol. 11, no. 14, pp. 1–15, Jul. 2023, doi: 10.3390/math11143094.

[12] D. Salinas, V. Flunkert, J. Gasthaus, and T. Januschowski, "DeepAR: Probabilistic forecasting with autoregressive recurrent networks," *Int. J. Forecasting*, vol. 36, no. 3, pp. 1181–1191, Jul. 2020, doi: 10.1016/j.ijforecast.2019.07.001.

[13] B. Lim, S. Ö. Arık, N. Loeff, and T. Pfister, "Temporal fusion transformers for interpretable multi-horizon time series forecasting," *Int. J. Forecasting*, vol. 37, no. 4, pp. 1748–1764, Oct. 2021, doi: 10.1016/j.ijforecast.2021.03.012.

[14] Y. Santur, "A novel financial forecasting approach using deep learning framework," *Comput. Econ.*, vol. 62, no. 3, pp. 1341–1392, Oct. 2023, doi: 10.1007/s10614-023-10403-5.

[15] L. Xu and Z. Dong, "Deep learning for financial forecasting and strategic business optimisation in enterprises," *Int. J. Inf. Commun. Technol.*, vol. 26, no. 19, pp. 1–15, 2025, doi: 10.1504/IJICT.2025.10071438.

[16] S. Giantsidi and C. Tarantola, "Deep learning for financial forecasting: A review of recent trends," *Int. Rev. Econ. Finance*, vol. 104, no. 1, pp. 1–15, Dec. 2025, doi: 10.1016/j.iref.2025.104719.

[17] G. Huang, "Dynamic neural networks: Advantages and challenges," *Natl. Sci. Rev.*, vol. 11, no. 8, pp. 1–10, Jul. 2024, doi: 10.1093/nsr/nwae088.

[18] W. Maass, "Networks of spiking neurons: The third generation of neural network models," *Neural Netw.*, vol. 10, no. 9, pp. 1659–1671, Dec. 1997, doi: 10.1016/S0893-6080(97)00011-7.

[19] M. Davies et al., "Loihi: A neuromorphic manycore processor with on-chip learning," *IEEE Micro*, vol. 38, no. 1, pp. 82–99, Jan. 2018, doi: 10.1109/MM.2018.112130359.

[20] S. K. Esser et al., "Convolutional networks for fast, energy-efficient neuromorphic computing," *Proc. Natl. Acad. Sci. U.S.A.*, vol. 113, no. 41, pp. 11441–11446, Oct. 2016, doi: 10.1073/pnas.1604850113.

[21] F. Zenke and S. Ganguli, "SuperSpike: Supervised learning in multilayer spiking neural networks," *Neural Comput.*, vol. 30, no. 6, pp. 1514–1541, Jun. 2018, doi: 10.1162/neco_a_01086.

[22] S. B. Shrestha and G. Orchard, "SLAYER: Spike layer error reassignment in time," *Adv. Neural Inf. Process. Syst.*, vol. 2018, no. Dec., pp. 1419–1428, Dec. 2018, doi: 10.48550/arXiv.1810.08646.

[23] P. U. Diehl, D. Neil, J. Binas, M. Cook, S.-C. Liu, and M. Pfeiffer, "Fast-classifying, high-accuracy spiking deep networks through weight and threshold balancing," *Proc. Int. Joint Conf. Neural Netw.*, vol. 2015, no. Jul., pp. 1–8, Jul. 2015, doi: 10.1109/IJCNN.2015.7280696.

[24] X. Kong et al., "Deep learning for time series forecasting: A survey," *Int. J. Mach. Learn. Cybern.*, vol. 16, no. 7–8, pp. 5079–5112, Aug. 2025, doi: 10.1007/s13042-025-02560-w.

[25] S. Nazlioglu, I. Kucukkaplan, E. Kilic, and M. Altuntas, "Financial market integration of emerging markets: Heavy tails, structural shifts, nonlinearity, and asymmetric persistence," *Res. Int. Bus. Finance*, vol. 62, no. 1, pp. 1–15, Dec. 2022, doi: 10.1016/j.ribaf.2022.101742.

[26] A. J. Patton and H. Zhang, "Bespoke realized volatility: Tailored measures of risk for volatility prediction," *J. Econometrics*, vol. 254, no. 1, pp. 1–20, Mar. 2026, doi: 10.1016/j.jeconom.2025.106122.

[27] R. M. Neal, "Bayesian learning for neural networks," *Lecture Notes Stat.*, vol. 118, no. 1, pp. 1–204, 1996, doi: 10.1007/978-1-4612-0745-0.

[28] V. Kuleshov, N. Fenner, and S. Ermon, "Accurate uncertainties for deep learning using calibrated regression," *Proc. Int. Conf. Mach. Learn.*, vol. 2018, no. Jul., pp. 2796–2804, Jul. 2018, doi: 10.48550/arXiv.1807.00263.

[29] F. Ponulak and A. Kasiński, "Supervised learning in spiking neural networks with ReSuMe: Sequence learning, classification, and spike shifting," *Neural Comput.*, vol. 22, no. 2, pp. 467–510, Feb. 2010, doi: 10.1162/neco.2009.11-08-901.

[30] G. G. Turrigiano, "The self-tuning neuron: Synaptic scaling of excitatory synapses," *Cell*, vol. 135, no. 3, pp. 422–435, Oct. 2008, doi: 10.1016/j.cell.2008.10.008.

[31] A. Niculescu-Mizil and R. Caruana, "Predicting good probabilities with supervised learning," *Proc. Int. Conf. Mach. Learn.*, vol. 2005, no. Aug., pp. 625–632, Aug. 2005, doi: 10.1145/1102351.1102430.

[32] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," *Proc. Int. Conf. Mach. Learn.*, vol. 2017, no. Aug., pp. 1320–1330, Aug. 2017, doi: 10.48550/arXiv.1706.04599.

[33] A. P. Dawid, "Present position and potential developments: Some personal views: Statistical theory: The prequential approach," *J. R. Stat. Soc. A*, vol. 147, no. 2, pp. 278–292, 1984, doi: 10.2307/2981683.

[34] J. Bröcker and L. A. Smith, "Scoring probabilistic forecasts: The importance of being proper," *Weather Forecast.*, vol. 22, no. 2, pp. 382–388, Apr. 2007, doi: 10.1175/WAF966.1.

[35] A. Agresti, "The foundations of statistical science: A history of textbook presentations," *Braz. J. Probab. Stat.*, vol. 35, no. 4, pp. 1–15, Nov. 2021, doi: 10.1214/21-BJPS518.

[36] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *Proc. Int. Conf. Learn. Representations*, vol. 2015, no. May, pp. 1–15, May 2015, doi: 10.48550/arXiv.1412.6980.

[37] C. Lee, S. S. Sarwar, P. Panda, G. Srinivasan, and K. Roy, "Enabling spike-based backpropagation for training deep neural network architectures," *Front. Neurosci.*, vol. 14, no. Feb., pp. 1–15, Feb. 2020, doi: 10.3389/fnins.2020.00119.

[38] Y. Wu, L. Deng, G. Li, J. Zhu, and L. Shi, "Spatio-temporal backpropagation for training high-performance spiking neural networks," *Front. Neurosci.*, vol. 12, no. May, pp. 1–10, May 2018, doi: 10.3389/fnins.2018.00331.

[39] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.

[40] R. Araoussi, "yfinance: Market data downloader in Python," *GitHub*, vol. 2026, no. 1, pp. 1–10, 2026. [Online]. Available: https://github.com/ranaroussi/yfinance

[41] M. Horowitz, "1.1 Computing's energy problem (and what we can do about it)," *Proc. IEEE Int. Solid-State Circuits Conf.*, vol. 2014, no. Feb., pp. 10–14, Feb. 2014, doi: 10.1109/ISSCC.2014.6757323.

[42] J. Xu, J. Pan, S. Su, H. Li, R. Zhao, and F. Ding, "Multivariate time series forecasting with a numerical spiking neural P system-inspired network," *Neurocomputing*, vol. 700, no. 1, pp. 1–12, Nov. 2026, doi: 10.1016/j.neucom.2026.134476.

[43] C. Lu et al., "Accelerating spiking neural networks with photonic reconfigurable devices," *Nat. Commun.*, vol. 17, no. 1, pp. 1–12, Apr. 2026, doi: 10.1038/s41467-026-72119-y.