# Fusion Strategies for Multi-Class Stock Movement Prediction: Balancing Temporal, Spatial, and Tabular Models

Yiwei Chang[1*]; Jinguo Lian[2]

Department of Mathematics & Statistics, University of Massachusetts Amherst, Amherst, MA, USA
*Corresponding Author

Received: 04 November 2025/ Revised: 13 November 2025/ Accepted: 19 November 2025/ Published: 30-11-2025

**Abstract—** *Accurate short-horizon stock-movement forecasting remains a central problem in computational finance, where even small directional errors can accumulate into significant trading risk. The most challenging regime is the neutral state—intervals with minor price changes that are easily masked by noise. To address this challenge, we compare three complementary learning paradigms and their combinations across multiple lookback horizons for three representative equities (AAPL, GOOG, TSLA). We evaluate Long Short-Term Memory (LSTM) networks for temporal dynamics, Convolutional Neural Networks (CNNs) on polar-transformed price images for spatial pattern extraction, and XGBoost on tabular technical indicators for structured feature learning.*

*Empirical results (Appendices A–C) reveal distinct horizon-dependent behaviors: CNNs excel at ultra-short windows (W = 1–3) with perfect accuracy and neutral-F1 ≈ 1.00 but deteriorate rapidly as horizons lengthen; LSTMs gain overall accuracy with longer windows (W = 30–60º) but lose sensitivity to neutral segments; and XGBoost remains the most stable single model, maintaining accuracy ≈ 0.89–0.93, low loss ≈ 0.4–0.6, and neutral-F1 ≈ 0.89–0.96 across assets.*

*Building on these complementary patterns, we propose fusion frameworks that integrate CNN and XGBoost outputs through weighted voting, cascaded thresholds, and probability-smoothed blending. The best configuration—probability-smoothed fusion—achieves roughly a 3–4 percentage-point improvement in neutral-F1 over the strongest standalone model while preserving comparable accuracy and calibration loss. The LSTM is retained solely as a benchmark to illustrate sequence-model trade-offs and is not included in the fusion.*

*Together, the results demonstrate that combining spatial and tabular perspectives yields more balanced recognition of neutral states without sacrificing directional accuracy. Accuracy measures overall correctness, loss captures probabilistic calibration, and F1 quantifies class-wise precision–recall balance. Viewed jointly, these metrics show that CNN–XGBoost fusion produces smoother and more interpretable predictions across assets and horizons. Such stability can reduce overtrading during ambiguous market phases, improving risk-adjusted decision-making in algorithmic trading strategies.*

***Keywords— Short-horizon stock prediction, CNN–XGBoost fusion, polar-coordinate transformation, financial time-series classification, probabilistic calibration, neutral-class F1.***

## I. INTRODUCTION

Predicting short-horizon stock direction remains a central challenge in financial modeling and algorithmic trading. Near-term price movements are affected by volatility, microstructure effects, and external shocks, producing highly non-stationary data that obscure clear predictive patterns [7]. Among the three outcome classes—up, down, and neutral—the neutral state is the most difficult to detect because its price changes are small and easily masked by random market noise. Misclassifying neutral periods as directional often results in unnecessary trades, increasing turnover and reducing risk-adjusted returns.

Recent machine-learning methods have advanced the ability to model financial time series from multiple perspectives. Long Short-Term Memory (LSTM) networks capture sequential dependencies and are widely used for temporal forecasting [1, 8]. Convolutional Neural Networks (CNNs), when applied to time-series segments rendered as polar-coordinate images, recognize local spatial patterns resembling technical chart structures [2]. XGBoost, a gradient-boosted ensemble of decision trees, remains a strong baseline for structured financial indicators and is known for its stability and resilience to class imbalance [3].

Each modeling family offers a distinct view of the market but also exhibits horizon-specific limitations, as demonstrated by the empirical findings in Appendices A–C. The LSTM's accuracy improves with longer lookback windows, yet its ability to identify neutral movements deteriorates. The CNN captures short-term spatial cues effectively—particularly at the one-day horizon—but its performance degrades sharply as the temporal window expands. XGBoost, by contrast, maintains steady accuracy, low loss, and high neutral-F1 scores across horizons, making it the most reliable standalone option in this study.

These complementary strengths motivate a targeted fusion design. In this work, we combine CNN and XGBoost outputs to integrate short-horizon pattern recognition with stable probabilistic calibration, while the LSTM is retained solely as a non-fused baseline. This separation clarifies the specific contribution of sequential modeling and helps benchmark fusion performance against traditional temporal approaches [4].

Our contributions are fourfold:

1) We conduct a systematic evaluation of LSTM, CNN, and XGBoost across multiple lookback windows (1–60 days) for three representative equities—Apple (AAPL), Alphabet (GOOG), and Tesla (TSLA).
2) We develop fusion strategies—weighted voting, cascaded thresholds, and probability smoothing—to merge CNN and XGBoost predictions.
3) We show that the best fusion configurations improve neutral-class F1 by up to approximately four percentage points compared to the strongest single model, while maintaining comparable accuracy and loss.
4) We interpret accuracy, loss, and F1 jointly: accuracy measures global correctness, loss reflects probability calibration, and F1 quantifies precision–recall balance, particularly for neutral detection.

The remainder of this paper is organized as follows: Section 2 describes the data sources, preprocessing pipeline, and model structures; Section 3 presents results for single models and fusion strategies across assets and time horizons; and Section 4 concludes with practical implications for trading and risk management, limitations, and avenues for future work

## II. MATERIALS AND METHODS

### 2.1 Data Source:

We employ daily OHLCV (open, high, low, close, volume) data for three highly liquid technology equities—Apple Inc. (AAPL), Alphabet Inc. (GOOG), and Tesla, Inc. (TSLA)—spanning June 10, 2020 to June 10, 2025. Data were retrieved from Yahoo Finance, a widely used source for academic research in quantitative finance. These stocks were selected to represent actively traded, high-volatility securities within the same sector, providing sufficiently diverse dynamics for cross-model evaluation while avoiding issues of thin trading or missing records.

### 2.2 Data Segmentation and Label Definition:

To study short-term predictability at different temporal scales, each price series is segmented into overlapping lookback windows of length $W \in \{1, 3, 7, 15, 30, 60\}$ trading days. These windows capture multiple horizons:

- **1–3 days:** ultra-short-term reactions to microstructure events and news shocks;
- **7–15 days:** short-term swings reflecting sentiment cycles and mean reversion;
- **30–60 days:** broader mini-trends incorporating regime and seasonal effects.

For each window ending at time $t$, the next-day return is:

$$r_{t+1} = \frac{P_{t+1} - P_t}{P_t} \tag{1}$$

Classes are assigned as

up if $r_{t+1} > +0.03$, neutral if $|r_{t+1}| \leq 0.03$, down if $r_{t+1} < -0.03$

This ±3% threshold reflects short-horizon volatility typical of large-cap U.S. equities.

### 2.3 Polar Coordinate Transformation for CNN Input:

Following Sezer and Ozbayoglu [2], each price window $P = \{p_1, p_2, \ldots, p_n\}$ is normalized and mapped into polar coordinates to transform sequential variations into spatial structures.

$$p_i^{norm} = \frac{p_i - min(P)}{max(P) - min(P)}, \theta_i = \frac{2\pi i}{n}, r_i = p_i^{norm} \tag{2}$$

This representation converts the one-dimensional sequence into a two-dimensional polar image where radius encodes relative price and angle encodes time. The polar layout preserves order and scale while exposing local curvatures that resemble chart formations commonly used in technical analysis. Empirically (Appendix A–C), such images produce clear distinctions in very short windows ($W$ =1–3) but lose sharpness as $W$ grows, explaining the CNN's degradation at long horizons.

### 2.4 Feature Construction for XGBoost:

For the gradient-boosted tree model, we derive structured tabular indicators from the same price data: daily returns, moving averages, momentum, relative strength index (RSI), Bollinger Bands, and rolling volatility. All features are standardized to zero mean and unit variance to ensure balanced influence during tree growth. This tabular design captures aggregated financial signals that complement the CNN's local pattern recognition [6].

### 2.5 LSTM Baseline:

The Long Short-Term Memory (LSTM) network serves as an **independent baseline** rather than a fused component. It models sequential dependencies directly from the raw normalized price series. Appendix A–C show that the LSTM achieves higher accuracy at longer windows but declining Neutral-F1, highlighting its horizon-dependent bias toward directional trends. Including it as a reference provides a benchmark for assessing how fusion (CNN + XGBoost) differs from traditional sequence modeling [9].

### 2.6 Handling Class Imbalance:

Neutral days dominate most stock series, creating an imbalanced classification task. To address this:

- **CNN:** applies class-weighted focal loss [5] emphasizing underrepresented classes (especially neutral) and uses early stopping to prevent overfitting;
- **XGBoost:** incorporates class weights directly into the objective function to maintain sensitivity to minority classes.

Both strategies aim to prevent the models from being biased toward majority up/down classes.

### 2.7 Preprocessing Workflow:

Figure 1 outlines the end-to-end workflow:

1) Collect OHLCV data for AAPL, GOOG, TSLA (2020–2025);
2) Segment each series into overlapping windows for six horizons;
3) Convert each window into two representations—polar images (CNN) and engineered features (XGBoost);
4) Apply normalization, class weighting, and focal loss
5) Split 80/20 for training/testing with validation monitoring.

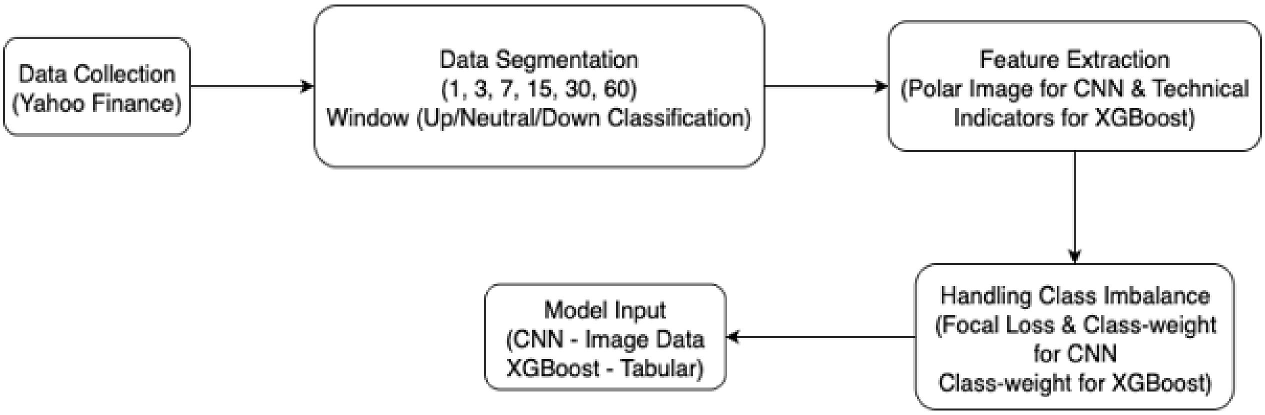
Data Collection (Yahoo Finance)
Data Segmentation (1, 3, 7, 15, 30, 60) Window (Up/Neutral/Down Classification)
Feature Extraction (Polar Image for CNN & Technical Indicators for XGBoost)
Handling Class Imbalance (Focal Loss & Class-weight for CNN Class-weight for XGBoost)
Model Input (CNN - Image Data XGBoost - Tabular)

**FIGURE 1: Data preprocessing pipeline showing window segmentation, polar image creation for CNN, and tabular feature generation for XGBoost**

### 2.8 Model Structures:

**CNN with Polar Input.** Polar images (H × Wc × 3 tensors) are standardized to the range [0, 1] and lightly augmented using rotation, scaling, and brightness jitter. We adopt a compact convolutional backbone with two variants:

- **Tri-class CNN:** outputs probabilities for *down*, *neutral*, and *up*.
- **Binary UD-CNN:** estimates the probability of *up* for fusion cascades.

Training uses the Adam optimizer with early stopping based on validation loss.

- **XGBoost.** We train a multiclass gradient-boosted tree model using the multi:softprob objective to output calibrated class probabilities. Typical hyperparameters include: learning rate = 0.1, max_depth = 6, n_estimators = 100, subsample = 0.8, and column sample = 0.8. Evaluation metrics include multi-class log-loss and overall accuracy.
- **Fusion Scope.** Consistent with the results in Appendices A–C, the fusion framework integrates only CNN and XGBoost. The LSTM remains separate as a sequential benchmark. This choice is empirically motivated: CNN provides strong sensitivity to short-horizon spatial signals, XGBoost offers stable probabilistic calibration, and combining them yields complementary improvements in neutral-class performance.

### 2.9 Evaluation Metrics:

To evaluate classification performance, we adopt three complementary metrics: **Accuracy**, **Cross-Entropy Loss**, and **F1 Score**. Accuracy reflects overall correctness; loss measures probabilistic calibration; and F1 captures the precision–recall balance, especially critical under class imbalance where the neutral class dominates. Formally:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}, Loss = -\frac{1}{N}\sum_{i,k} y_{ik} \log(\hat{p}_{ik}), F1 = \frac{2 \times (Precision \times Recall)}{Precision + Recall}$$

A model with high accuracy, low loss, and balanced F1 across classes is considered both well-calibrated and suitable for decision-making.

### 2.10 Fusion Strategies:

#### 2.10.1 Soft Voting:

Let $p_{cnn}, p_{xgb} \in \mathbb{R}^3$denote class-probability vectors for (down, neutral, up). Weighted averaging forms the fused prediction:

$$\hat{p} = normalize(Wp_{cnn} + (I - W)p_{xgb}) \quad (3)$$

where $W = diag(w_{\text{down}}, w_{\text{neutral}}, w_{\text{up}})$. Weights $w_{\text{down}}, w_{\text{up}} > 0.5$ emphasize CNN directional detection, while $w_{\text{neutral}} < 0.5$ assigns greater confidence to XGBoost for neutral identification.

### 2.10.2 Hybrid Fusion Model:

**Cascaded Thresholds:** A binary CNN provides $p_{up}$, and XGBoost outputs $p_{neutral}$. The confidence score $conf_{ud} = \max(p_{up}, 1 - p_{up})$ controls the cascade:

- If $conf_{ud} \geq \tau_{ud}$, choose CNN's up/down decision.
- Else if $p_{neutral} \geq \tau_{neu}$, predict neutral.
- Otherwise, fall back to the higher-confidence CNN output.

Thresholds $(\tau_{ud}, \tau_{neu})$ are grid-searched per asset.

### 2.10.3 Probability-Smoothed Fusion:

To further stabilize predictions, cascaded outputs are passed through a probability-smoothing calibration stage. By applying softmax temperature scaling or a confidence threshold $\tau_{neu}$, the resulting probability distribution becomes smoother and better calibrated across all three classes. This reduces overconfidence and aligns CNN and XGBoost probability scales, improving multi-model consistency.

To avoid hard thresholds, we generate a continuously smoothed tri-class distribution:

$$p'_{\text{down}} = 1 - p_{Pup} \tag{4}$$

$$p_{\text{not-neutral}} = 1 - p_{neutral} \tag{5}$$

$$\tilde{p}_{down} = p_{\text{not-neutral}} \cdot \frac{p'_{down}}{p_{up}+p'_{down}}, \tilde{p}_{up} = p_{\text{not-neutral}} \cdot \frac{p_{up}}{p_{up}+p'_{down}} \tag{6}$$

$$\hat{P} = normalize\{\tilde{p}_{down}, p_{neutral}, \tilde{p}_{up}\} \tag{7}$$

Together, these steps form the **Cascaded and Probability-Smoothed Fusion** model, where the cascaded decision structure ensures directional selectivity, and the smoothing process enforces consistent probabilistic calibration across all three classes.

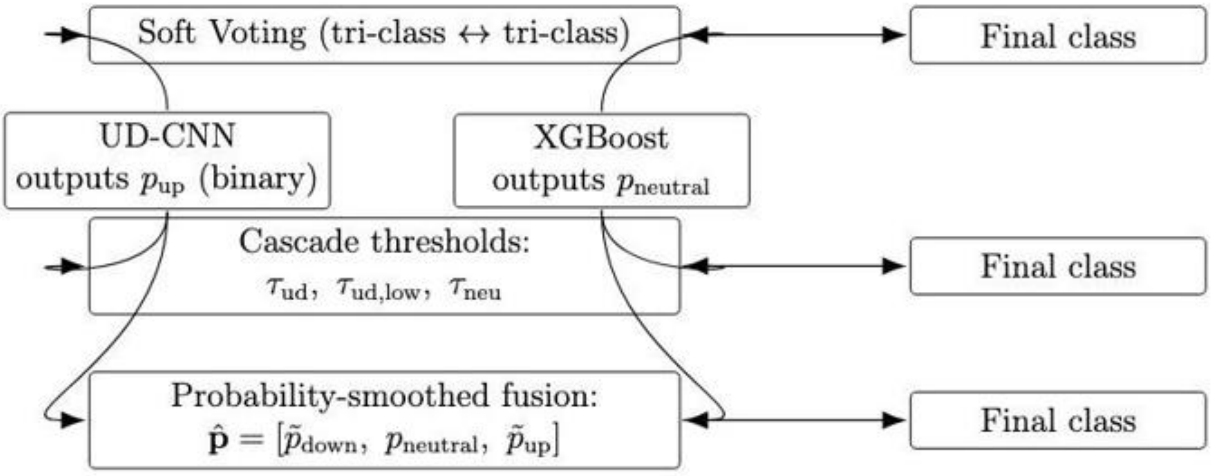
Soft Voting (tri-class ↔ tri-class)
Final class
UD-CNN outputs $p_{\text{up}}$ (binary)
XGBoost outputs $p_{\text{neutral}}$
Cascade thresholds: $\tau_{\text{ud}}$, $\tau_{\text{ud,low}}$, $\tau_{\text{neu}}$
Final class
Probability-smoothed fusion: $\hat{\mathbf{p}} = [\tilde{p}_{\text{down}}, p_{\text{neutral}}, \tilde{p}_{\text{up}}]$
Final class

**FIGURE 2: Fusion strategies tested: weighted voting, cascaded thresholds, and probability smoothing. All combine CNN and XGBoost outputs; the LSTM is evaluated separately as a reference baseline**

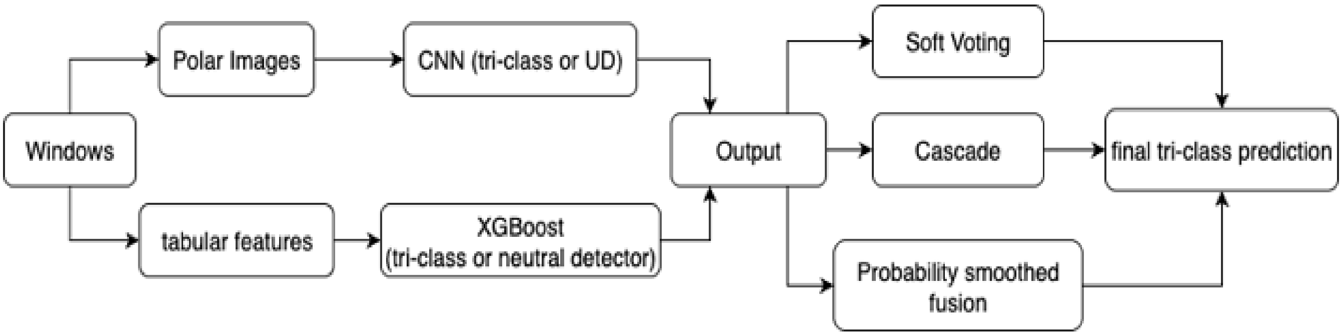

**FIGURE 3: Overall architecture: polar images feed the CNN, tabular indicators feed XGBoost, and their probability outputs are fused. The LSTM baseline is run independently for comparison**

# III. RESULTS AND DISCUSSION

## 3.1 Single-Model Baselines Across Horizons:

### 3.1.1 CNN with Polar-Image Input:

When short windows are used, the CNN trained on polar-coordinate images achieves exceptionally strong performance. At $W = 1$ day, it attains perfect accuracy and a Neutral-F1 score of **1.00** across all assets (Appendix A and E). This confirms that the polar transformation enables the network to capture immediate geometric cues in price movement—such as sharp reversals or local plateaus—that are highly discriminative at ultra-short horizons.

However, this advantage diminishes rapidly as the lookback window increases. For AAPL at $W = 30$, accuracy falls to **0.723** and Neutral-F1 to **0.52**, and at $W = 60$ the Neutral class collapses to **0**. Loss values (Appendix B) simultaneously rise toward **0.9**, indicating poor probability calibration at longer horizons. This degradation arises because longer polar sequences lose spatial coherence—images become noisy, overlapping, and visually ambiguous, making them harder for convolutional filters to separate effectively. Therefore, CNNs perform best for ultra-short forecasts but are unreliable for medium- or long-term prediction windows.

### 3.1.2 LSTM (Sequential Baseline):

The LSTM provides a meaningful temporal benchmark. Accuracy improves steadily with longer horizons—from approximately **0.80** at $W = 3$ to **0.86** at $W = 60$(Appendix A)—indicating that additional temporal context strengthens directional prediction. However, Neutral-F1 decreases sharply over the same range (AAPL: **0.95** → **0.56**; GOOG: **0.95** → **0.43**), demonstrating an increasing tendency to classify ambiguous days as directional movement.

Loss values decrease with increasing window length (Appendix B), suggesting improved probabilistic calibration, but this does not correct the declining sensitivity to neutral periods. As a result, the LSTM is retained only as a comparative baseline to illustrate the horizon-dependent trade-off between accuracy and class balance.

### 3.1.3 XGBoost (Tabular Features):

Across all horizons, XGBoost exhibits the most stable and balanced performance. For AAPL at $W = 30$, it achieves an accuracy of **0.89**, loss of **0.50**, and Neutral-F1 of **0.94**—levels that remain similarly strong for GOOG and TSLA. Appendix A reports accuracy levels between **0.89–0.93** for AAPL and GOOG, with moderately lower values (**0.53–0.62**) for TSLA due to greater volatility.

Loss remains consistently low (**0.4–0.6**, Appendix B), confirming strong probability calibration. Appendix E shows Neutral-F1 values remaining between **0.89–0.96** across horizons, making XGBoost the most dependable single model for neutral detection.

These results align with prior findings on ensemble-method advantages reported by Chen and Guestrin [3] and other studies on hybrid financial forecasting frameworks [4].

TABLE 1
REPRESENTATIVE SINGLE-MODEL RESULTS FOR AAPL (30-DAY WINDOW).

| Model | Window | Accuracy | Loss | Macro-F1 | Up-F1 | Neutral-F1 | Down-F1 |
|---|---|---|---|---|---|---|---|
| LSTM | 30 | 0.776 | 0.446 | 0.68 | 0.83 | 0.51 | 0.73 |
| CNN | 30 | 0.723 | 0.576 | 0.71 | 0.81 | 0.52 | 0.8 |
| XGBoost | 30 | 0.89 | 0.497 | 0.31 | 0 | 0.94 | 0 |

### 3.2 Fusion of CNN and XGBoost:

All fusion experiments combine the outputs of the CNN and XGBoost models, while the LSTM is excluded from fusion and used only as a reference baseline. Three fusion mechanisms were evaluated: soft voting, cascaded thresholds, and probability smoothing. The goal was to assess whether combining spatial representations from CNNs with tabular feature learning from XGBoost could improve prediction stability, calibration, and neutral-state detection across forecasting horizons.

### 3.2.1 Soft Voting:

In the soft-voting approach, weighted probabilities from CNN and XGBoost are combined based on performance characteristics. Higher weights were assigned to the CNN for movement direction (up and down), while XGBoost received higher emphasis for detecting neutral periods. Grid-based hyperparameter tuning identified optimal balance points, yielding Macro-F1 scores of approximately 0.86 and Neutral-F1 scores near 0.96 in the strongest configuration (GOOG, W = 15). Despite these improvements, gains remained inconsistent across assets and horizons. This variability indicates that fixed weighting strategies cannot fully adapt to changing market volatility or asset-specific behavior.

### 3.2.2 Cascaded Thresholds:

The cascaded framework introduces conditional routing of predictions. The CNN first evaluates directional confidence; if confidence exceeds a defined threshold, the prediction is accepted. Otherwise, the decision is deferred to XGBoost, particularly for potential neutral cases. The best-performing thresholds were (0.8, 0.65), producing highly competitive results in isolated configurations. For example, the cascade achieved a Neutral-F1 of 0.99 for AAPL at W = 60 and 0.96 for GOOG at W = 15. However, performance consistency decreased when evaluating pooled three-class predictions across all assets. Test loss also fluctuated notably, suggesting that cascaded logic is highly sensitive to hyperparameter selection and asset volatility characteristics.

### 3.2.3 Probability-Smoothed Fusion:

Probability smoothing replaces binary routing logic with a continuous blending mechanism that mixes CNN directional confidence with XGBoost's neutral probability. This eliminates abrupt decision boundaries and produces smoother predicted distributions. The approach achieved the most stable cross-asset performance. For example, GOOG at W = 30 achieved accuracy of 0.919, loss of 0.38, and Neutral-F1 of 0.96. Performance remains robust even at longer windows (W = 60), particularly for more volatile assets such as TSLA. On average, across all assets and horizons, Neutral-F1 improvements of approximately 3–4 percentage points were observed relative to the strongest individual model, while accuracy and calibration loss remained comparable. These results indicate that integrating spatial and tabular learning perspectives through probability smoothing yields more balanced three-class performance without compromising overall predictive reliability.

TABLE 2
REPRESENTATIVE FUSION RESULTS FOR GOOG (15-DAY WINDOW)

| Model | Window | Accuracy | Loss | Macro-F1 | Up-F1 | Neutral-F1 | Down-F1 |
|---|---|---|---|---|---|---|---|
| Soft Voting | 15 | 0.938 | 0.457 | 0.86 | 0.75 | 0.96 | 0.86 |
| Hybrid Fusion Model | 15 | 0.857 | 0.852 | 0.85 | 0.85 | 0.91 | 0.8 |

### 3.3 Comparative Interpretation:

Accuracy reflects the proportion of correct predictions across all three classes. As shown in Appendix A, XGBoost and the fused approaches maintain the most stable accuracy across forecasting windows. In contrast, the CNN performs strongly only at very short horizons and deteriorates rapidly beyond W = 3. The LSTM shows increasing accuracy as the temporal window expands, although this does not translate into balanced classification across all classes.

- Loss represents probabilistic calibration, where lower values indicate that predicted probabilities correspond meaningfully to observed class frequencies. Appendix B demonstrates that XGBoost and the probability-smoothed fusion method achieve the lowest and most consistent loss values, typically between 0.3 and 0.6. This indicates that both models not only predict correctly but also express confidence in a manner aligned with empirical outcomes.
- Neutral-F1 evaluates precision and recall specifically for the neutral class, which is operationally important for preventing unnecessary directional trades. Results in Appendix C show that fusion models improve Neutral-F1 by approximately three to four percentage points compared with the strongest individual model, confirming that combining CNN's sensitivity to short-term structure with XGBoost's feature stability produces measurable classification benefits.

Taken together, the metrics reveal a consistent pattern in model behavior. The CNN is most effective at ultra-short temporal horizons but declines as sequences become longer. The LSTM benefits from extended temporal context but loses precision in distinguishing neutral states. XGBoost remains the most stable and well-calibrated single model across assets and horizons. The fusion models, particularly those incorporating probability smoothing, provide incremental yet consistent improvements in Neutral-F1 while maintaining comparable accuracy and improved calibration behavior.

These results demonstrate that integrating spatially derived representations with structured tabular learning can mitigate, though not fully eliminate, the inherent trade-off between directional accuracy and neutral-state detection in short-horizon financial forecasting.

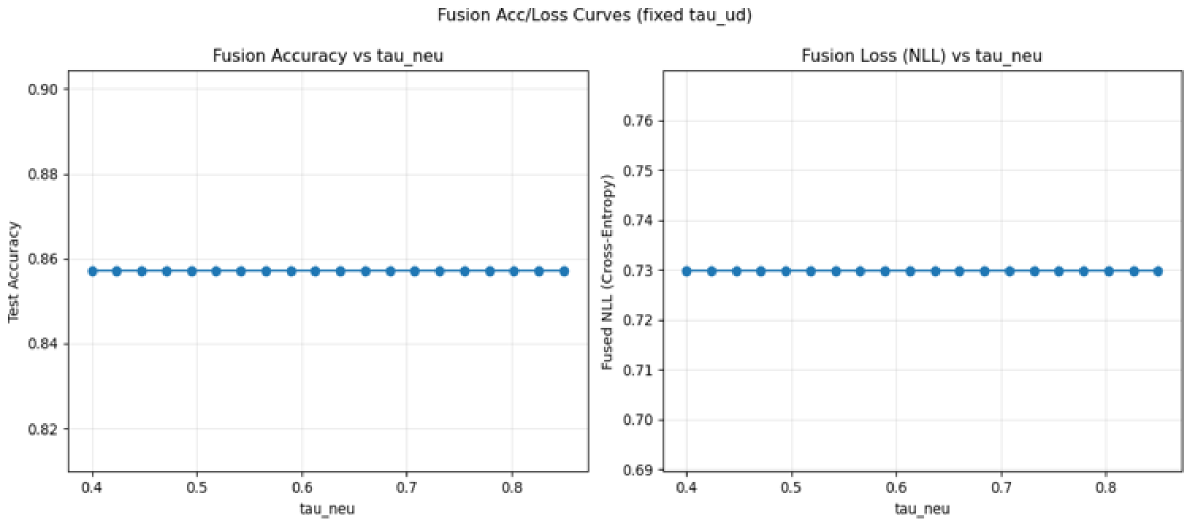

**FIGURE 4: Accuracy and loss sensitivity to the neutral threshold $\tau$ neu in the Hybrid Fusion**

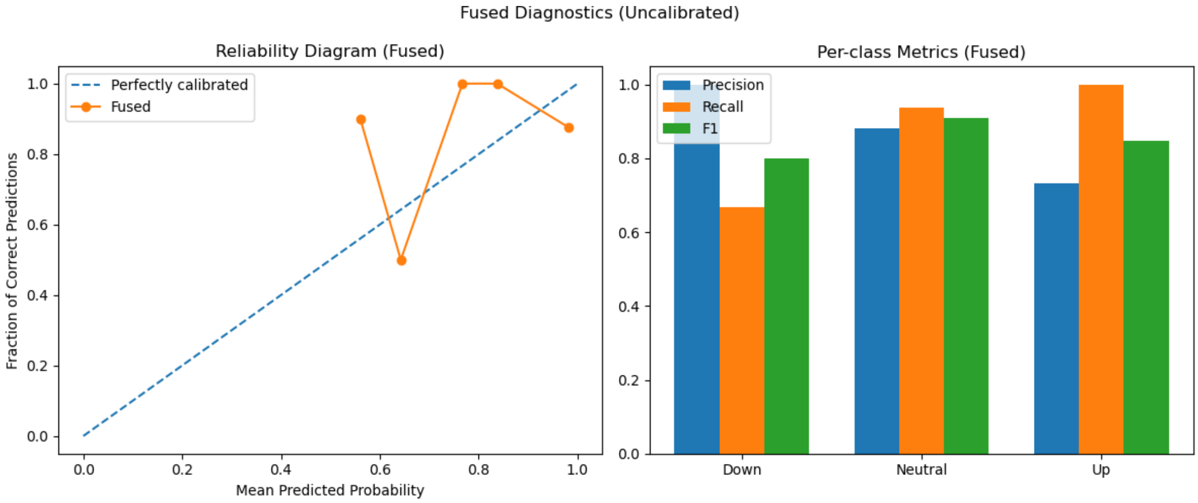

**FIGURE 5: Reliability and per-class performance for representative fusion configurations. Probability-smoothed fusion shows the most stable calibration across classes**

## IV. CONCLUSION AND FUTURE SCOPE

This study investigated short-horizon stock movement prediction with a specific focus on accurately identifying the neutral class across multiple forecasting intervals. Three complementary modeling approaches were analyzed: Convolutional Neural Networks (CNNs) using polar-transformed price images, Long Short-Term Memory (LSTM) networks for sequence learning, and XGBoost trained on structured financial indicators. Results across AAPL, GOOG, and TSLA (Appendices A–F) show that each method exhibits distinct behavior depending on the prediction window, motivating the proposed hybrid design.

The CNN demonstrated exceptional performance at ultra-short windows (W = 1–3), achieving perfect accuracy and Neutral-F1 scores near 1.00. However, its effectiveness declined sharply as the temporal window expanded due to increasing spatial complexity in the polar representation. The LSTM exhibited the opposite trend: accuracy improved with longer horizons (W = 30–60), yet sensitivity to neutral patterns weakened, leading to directional overbias. XGBoost delivered the most stable and well-balanced performance across all settings, maintaining accuracy between approximately 0.89 and 0.93, loss values near 0.4–0.6, and consistently high Neutral-F1 scores between 0.89 and 0.96. This made XGBoost the most reliable standalone approach.

To leverage complementary strengths, three fusion strategies combining CNN and XGBoost outputs were evaluated: weighted soft voting, cascaded thresholds, and probability-smoothed blending. Among these, probability-smoothed fusion yielded the most robust results, improving Neutral-F1 by approximately 3–4 percentage points relative to the strongest single model while maintaining comparable accuracy and calibration. The LSTM remained part of the study only as a conceptual benchmark to highlight contrast with non-sequential architectures and to help interpret horizon-dependent trade-offs.

Across all experiments, accuracy, loss, and F1 score provided complementary evaluation perspectives. Accuracy measured overall correctness, loss captured probability calibration and confidence alignment, and F1 quantified the precision–recall balance essential for neutral prediction. Together, these metrics show that the CNN–XGBoost fusion approach produces more stable and interpretable forecasts across assets and horizons, particularly for applications where avoiding false directional signals is important.

From an applied standpoint, improved neutral detection can reduce unnecessary trades, thereby lowering transaction costs and improving risk-adjusted returns in algorithmic trading systems. The stability of XGBoost and the incremental yet consistent improvement from fusion methods provide practical insight: combining spatial and tabular representations mitigates overconfidence during volatile regimes, while sequence models remain relevant for broader directional forecasting tasks.

Several limitations warrant further study. The experiments were conducted on three individual equities; extending evaluation to diversified assets, market indices, and higher-frequency intraday data would strengthen generalization. Macroeconomic, sentiment, and options-derived variables were not incorporated and may provide additional predictive value. Finally, fusion weights were static; adaptive, regime-aware ensembles may better capture dynamic market behavior.

Future work may explore:

1) Incorporating external signals such as sentiment, macroeconomic indicators, or options-derived features into the polar-CNN architecture;

2) Testing alternative spatial encodings to preserve structure at longer windows;

3) Developing adaptive or self-tuning fusion systems responsive to volatility shifts; and

4) Extending the framework to portfolio-level or multi-asset forecasting.

Advancing these directions may establish the CNN–XGBoost hybrid as a generalizable and interpretable method for robust financial time-series prediction across diverse market environments.

## ACKNOWLEDGEMENTS

This research is supported by the UMass Amherst Research Support Fund (RSF). The authors thank the university for providing the computational resources and infrastructure that made this work possible.

## CONFLICT OF INTEREST

The authors declare no conflict of interest.

## REFERENCES

[1] T. Fischer and C. Krauss, “Deep learning with long short-term memory networks for financial market predictions,” *European Journal of Operational Research*, vol. 270, no. 2, pp. 654–669, 2018.

[2] O. B. Sezer and A. M. Ozbayoglu, “Algorithmic financial trading with deep convolutional neural networks: Time series to image conversion approach,” *Applied Soft Computing*, vol. 70, pp. 525–538, 2018.

[3] T. Chen and C. Guestrin, “XGBoost: A scalable tree boosting system,” in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 785–794.

[4] M. Ballings, D. Van den Poel, N. Hespeels, and R. Gryp, “Evaluating multiple classifiers for stock price direction prediction,” *Expert Systems with Applications*, vol. 42, no. 20, pp. 7046–7056, 2015.

[5] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, “Focal Loss for Dense Object Detection,” in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2980–2988.

[6] Y. Zhang and L. Wu, “Stock market prediction of S&P 500 via combination of improved BCO approach and BP neural network,” *Expert Systems with Applications*, vol. 36, no. 5, pp. 8849–8854, 2009.

[7] K. Kim, “Financial time series forecasting using support vector machines,” *Neurocomputing*, vol. 55, no. 1–2, pp. 307–319, 2003.

[8] W. Bao, J. Yue, and Y. Rao, “A deep learning framework for financial time series using stacked autoencoders and long short-term memory,” *PLOS ONE*, vol. 12, no. 7, p. e0180944, 2017.

[9] X. Ding, Y. Zhang, T. Liu, and J. Duan, “Deep learning for event-driven stock prediction,” in *Proceedings of the 24th International Conference on Artificial Intelligence*, 2015, pp. 2327–2333.

## APPENDIX

**Detailed Accuracy Results:**

**TABLE 3**
**TEST ACCURACY OF DIFFERENT MODELS ACROSS TIME WINDOWS FOR AAPL, GOOG, TSLA, AND THE AGGREGATED THREE-CLASS (3C) SETTING**

| Model | Window | AAPL Acc | GOOG Acc | TSLA Acc | 3C Acc |
|---|---|---|---|---|---|
| LSTM | 1 | 0.9 | 0.896 | 0.614 | 0.803 |
| | 3 | 0.809 | 0.777 | 0.709 | 0.754 |
| | 7 | 0.712 | 0.7 | 0.732 | 0.73 |
| | 15 | 0.762 | 0.734 | 0.815 | 0.808 |
| | 30 | 0.776 | 0.816 | 0.869 | 0.835 |
| | 60 | 0.858 | 0.862 | 0.895 | 0.861 |
| CNN | 1 | 1 | 1 | 1 | 1 |
| | 3 | 0.78 | 0.812 | 0.571 | 0.647 |
| | 7 | 0.625 | 0.653 | 0.812 | 0.718 |
| | 15 | 0.796 | 0.708 | 0.771 | 0.912 |
| | 30 | 0.723 | 0.771 | 0.792 | 0.856 |
| | 60 | 0.596 | 0.809 | 0.872 | 0.846 |
| XGBoost | 1 | 0.908 | 0.885 | 0.53 | 0.794 |
| | 3 | 0.892 | 0.9 | 0.55 | 0.811 |
| | 7 | 0.892 | 0.896 | 0.552 | 0.814 |
| | 15 | 0.887 | 0.907 | 0.621 | 0.805 |
| | 30 | 0.89 | 0.91 | 0.616 | 0.784 |
| | 60 | 0.912 | 0.929 | 0.561 | 0.803 |
| CNN+XGBoost (Soft Voting) | 1 | 0.74 | 0.82 | 0.46 | 0.467 |
| | 3 | 0.74 | 0.833 | 0.49 | 0.487 |
| | 7 | 0.812 | 0.837 | 0.5 | 0.487 |
| | 15 | 0.653 | 0.938 | 0.479 | 0.493 |
| | 30 | 0.809 | 0.646 | 0.562 | 0.452 |
| | 60 | 0.979 | 0.681 | 0.681 | 0.448 |
| CNN+XGBoost (Hybrid Fusion Model) | 3 | 0.804 | 0.784 | 0.765 | 0.583 |
| | 7 | 0.638 | 0.74 | 0.8 | 0.647 |
| | 15 | 0.61 | 0.857 | 0.86 | 0.647 |
| | 30 | 0.784 | 0.919 | 0.9 | 0.811 |
| | 60 | 0.875 | 0.871 | 0.917 | 0.891 |

**Detailed Loss Results:**

### TABLE 4
### TEST LOSS OF DIFFERENT MODELS ACROSS TIME WINDOWS FOR AAPL, GOOG, TSLA, AND THE AGGREGATED THREE-CLASS (3C) SETTING

| Model | Window | AAPL Loss | GOOG Loss | TSLA Loss | 3C Loss |
|---|---|---|---|---|---|
| LSTM | 1 | 0.375 | 0.402 | 0.937 | 0.606 |
| | 3 | 0.465 | 0.518 | 0.675 | 0.601 |
| | 7 | 0.642 | 0.699 | 0.598 | 0.637 |
| | 15 | 0.544 | 0.557 | 0.389 | 0.492 |
| | 30 | 0.446 | 0.429 | 0.332 | 0.384 |
| | 60 | 0.316 | 0.36 | 0.303 | 0.345 |
| CNN | 1 | 0 | 0 | 0 | 0 |
| | 3 | 0.039 | 0.075 | 0.056 | 0.061 |
| | 7 | 0.076 | 0.05 | 0.06 | 0.057 |
| | 15 | 0.043 | 0.067 | 0.078 | 0.027 |
| | 30 | 0.576 | 0.47 | 0.524 | 0.036 |
| | 60 | 0.9 | 0.53 | 0.36 | 0.047 |
| XGBoost | 1 | 0.423 | 0.493 | 1.113 | 0.657 |
| | 3 | 0.432 | 0.409 | 1.065 | 0.598 |
| | 7 | 0.463 | 0.419 | 1.017 | 0.566 |
| | 15 | 0.44 | 0.389 | 0.937 | 0.585 |
| | 30 | 0.497 | 0.376 | 0.946 | 0.634 |
| | 60 | 0.399 | 0.302 | 1.091 | 0.589 |
| CNN+XGBoost (Soft Voting) | 1 | 0.135 | 0.108 | 0.248 | 0.284 |
| | 3 | 0.576 | 0.509 | 1.011 | 1.086 |
| | 7 | 0.565 | 0.51 | 0.819 | 1.086 |
| | 15 | 0.642 | 0.457 | 0.773 | 0.917 |
| | 30 | 0.527 | 0.586 | 0.672 | 0.871 |
| | 60 | 0.569 | 0.636 | 0.672 | 0.909 |
| CNN+XGBoost (Hybrid Fusion Model) | 3 | 0.721 | 0.467 | 0.865 | 0.736 |
| | 7 | 0.462 | 0.52 | 0.856 | 0.497 |
| | 15 | 0.912 | 0.852 | 0.768 | 0.497 |
| | 30 | 0.993 | 0.38 | 0.363 | 0.414 |
| | 60 | 0.747 | 0.256 | 0.357 | 0.304 |

**Detailed Macro-F1 Results:**

TABLE 5
TEST MACRO-F1 OF DIFFERENT MODELS ACROSS TIME WINDOWS FOR AAPL, GOOG, TSLA, AND THE AGGREGATED THREE-CLASS (3C) SETTING

| Model | Window | AAPL F1 | GOOG F1 | TSLA F1 | 3C F1 |
|---|---|---|---|---|---|
| LSTM | 1 | 0.32 | 0.32 | 0.25 | 0.3 |
| | 3 | 0.65 | 0.66 | 0.72 | 0.69 |
| | 7 | 0.7 | 0.69 | 0.66 | 0.73 |
| | 15 | 0.77 | 0.73 | 0.74 | 0.79 |
| | 30 | 0.68 | 0.7 | 0.61 | 0.72 |
| | 60 | 0.79 | 0.74 | 0.62 | 0.78 |
| CNN | 1 | 1 | 1 | 1 | 1 |
| | 3 | 0.29 | 0.59 | 0.72 | 0.38 |
| | 7 | 0.65 | 0.67 | 0.79 | 0.75 |
| | 15 | 0.79 | 0.73 | 0.71 | 0.9 |
| | 30 | 0.71 | 0.58 | 0.66 | 0.81 |
| | 60 | 0.24 | 0.77 | 0.72 | 0.73 |
| XGBoost | 1 | 0.32 | 0.35 | 0.29 | 0.3 |
| | 3 | 0.32 | 0.32 | 0.34 | 0.31 |
| | 7 | 0.35 | 0.38 | 0.28 | 0.35 |
| | 15 | 0.36 | 0.32 | 0.38 | 0.34 |
| | 30 | 0.31 | 0.32 | 0.35 | 0.32 |
| | 60 | 0.32 | 0.32 | 0.31 | 0.38 |
| CNN+XGBoost (Soft Voting) | 1 | 0.28 | 0.3 | 0.29 | 0.23 |
| | 3 | 0.28 | 0.3 | 0.33 | 0.25 |
| | 7 | 0.48 | 0.41 | 0.48 | 0.25 |
| | 15 | 0.5 | 0.86 | 0.47 | 0.46 |
| | 30 | 0.7 | 0.57 | 0.56 | 0.43 |
| | 60 | 0.98 | 0.58 | 0.68 | 0.45 |
| CNN+XGBoost (Hybrid Fusion Model) | 3 | 0.42 | 0.47 | 0.77 | 0.58 |
| | 7 | 0.46 | 0.52 | 0.78 | 0.61 |
| | 15 | 0.58 | 0.85 | 0.77 | 0.61 |
| | 30 | 0.79 | 0.91 | 0.73 | 0.7 |
| | 60 | 0.86 | 0.86 | 0.79 | 0.73 |

**Detailed Up-F1 Results:**

TABLE 6
TEST UP-F1 OF DIFFERENT MODELS ACROSS TIME WINDOWS FOR AAPL, GOOG, TSLA, AND THE AGGREGATED THREE-CLASS (3C) SETTING

| Model | Window | AAPL F1 | GOOG F1 | TSLA F1 | 3C F1 |
|---|---|---|---|---|---|
| LSTM | 1 | 0 | 0 | 0 | 0 |
| | 3 | 0.58 | 0.54 | 0.75 | 0.61 |
| | 7 | 0.75 | 0.67 | 0.84 | 0.79 |
| | 15 | 0.8 | 0.81 | 0.89 | 0.88 |
| | 30 | 0.83 | 0.87 | 0.9 | 0.88 |
| | 60 | 0.93 | 0.91 | 0.94 | 0.93 |
| CNN | 1 | - | - | - | - |
| | 3 | 0 | 0.57 | 0.67 | 0.17 |
| | 7 | 0.62 | 0.92 | 0.88 | 0.76 |
| | 15 | 0.92 | 0.81 | 0.91 | 0.93 |
| | 30 | 0.81 | 0.92 | 0.87 | 0.93 |
| | 60 | 0.71 | 0.92 | 0.92 | 0.93 |
| XGBoost | 1 | 0 | 0.12 | 0.06 | 0 |
| | 3 | 0 | 0 | 0.17 | 0 |
| | 7 | 0.1 | 0.2 | 0.06 | 0.02 |
| | 15 | 0.12 | 0 | 0.15 | 0.06 |
| | 30 | 0 | 0 | 0.1 | 0.02 |
| | 60 | 0 | 0 | 0.1 | 0.09 |
| CNN+XGBoost (Soft Voting) | 1 | 0 | 0 | 0.1 | 0 |
| | 3 | 0 | 0 | 0 | 0.05 |
| | 7 | 0 | 0 | 0.36 | 0.05 |
| | 15 | 0.12 | 0.75 | 0.45 | 0.44 |
| | 30 | 0.5 | 0.27 | 0.53 | 0.4 |
| | 60 | 1 | 0.38 | 0.67 | 0.42 |
| CNN+XGBoost (Hybrid Fusion Model) | 3 | 0.36 | 0.53 | 0.77 | 0.54 |
| | 7 | 0.56 | 0.67 | 0.8 | 0.78 |
| | 15 | 0.6 | 0.85 | 0.93 | 0.78 |
| | 30 | 0.1 | 0.94 | 0.95 | 0.87 |
| | 60 | 0.8 | 0.9 | 0.95 | 0.91 |

**Detailed Neutral-F1 Results:**

TABLE 7
TEST NEUTRAL-F1 OF DIFFERENT MODELS ACROSS TIME WINDOWS FOR AAPL, GOOG, TSLA, AND THE AGGREGATED THREE-CLASS (3C) SETTING

| Model | Window | AAPL F1 | GOOG F1 | TSLA F1 | 3C F1 |
|---|---|---|---|---|---|
| LSTM | 1 | 0.95 | 0.95 | 0.76 | 0.89 |
| | 3 | 0.88 | 0.85 | 0.64 | 0.82 |
| | 7 | 0.73 | 0.72 | 0.34 | 0.7 |
| | 15 | 0.65 | 0.64 | 0.46 | 0.66 |
| | 30 | 0.51 | 0.5 | 0.07 | 0.46 |
| | 60 | 0.56 | 0.43 | 0 | 0.49 |
| CNN | 1 | 1 | 1 | 1 | 1 |
| | 3 | 0.86 | 0.9 | 0.67 | 0.77 |
| | 7 | 0.65 | 0.8 | 0.67 | 0.71 |
| | 15 | 0.71 | 0.63 | 0.58 | 0.82 |
| | 30 | 0.52 | 0.33 | 0.29 | 0.69 |
| | 60 | 0 | 0.5 | 0.33 | 0.41 |
| XGBoost | 1 | 0.95 | 0.94 | 0.7 | 0.89 |
| | 3 | 0.95 | 0.95 | 0.71 | 0.9 |
| | 7 | 0.94 | 0.95 | 0.71 | 0.9 |
| | 15 | 0.94 | 0.95 | 0.76 | 0.89 |
| | 30 | 0.94 | 0.95 | 0.76 | 0.89 |
| | 60 | 0.95 | 0.96 | 0.72 | 0.89 |
| CNN+XGBoost (Soft Voting) | 1 | 0.85 | 0.9 | 0.63 | 0.63 |
| | 3 | 0.85 | 0.91 | 0.63 | 0.65 |
| | 7 | 0.91 | 0.91 | 0.59 | 0.65 |
| | 15 | 0.8 | 0.96 | 0.52 | 0.65 |
| | 30 | 0.89 | 0.77 | 0.59 | 0.58 |
| | 60 | 0.99 | 0.78 | 0.72 | 0.48 |
| CNN+XGBoost (Hybrid Fusion Model) | 3 | 0.9 | 0.88 | 0.73 | 0.6 |
| | 7 | 0.82 | 0.89 | 0.64 | 0.33 |
| | 15 | 0.77 | 0.91 | 0.5 | 0.33 |
| | 30 | 0.87 | 0.96 | 0.33 | 0.33 |
| | 60 | 0.78 | 0.83 | 0.5 | 0.32 |

**Detailed Down-F1 Results:**

TABLE 8
TEST DOWN-F1 OF DIFFERENT MODELS ACROSS TIME WINDOWS FOR AAPL, GOOG, TSLA, AND THE AGGREGATED THREE-CLASS (3C) SETTING

| Model | Window | AAPL F1 | GOOG F1 | TSLA F1 | 3C F1 |
|---|---|---|---|---|---|
| LSTM | 1 | 0 | 0 | 0 | 0 |
| | 3 | 0.48 | 0.61 | 0.76 | 0.64 |
| | 7 | 0.61 | 0.68 | 0.79 | 0.7 |
| | 15 | 0.84 | 0.75 | 0.87 | 0.83 |
| | 30 | 0.73 | 0.75 | 0.87 | 0.83 |
| | 60 | 0.89 | 0.87 | 0.9 | 0.92 |
| CNN | 1 | - | - | - | - |
| | 3 | 0 | 0.29 | 0.81 | 0.2 |
| | 7 | 0.67 | 0.29 | 0.81 | 0.77 |
| | 15 | 0.74 | 0.75 | 0.64 | 0.94 |
| | 30 | 0.8 | 0.5 | 0.83 | 0.82 |
| | 60 | 0 | 0.89 | 0.9 | 0.85 |
| XGBoost | 1 | 0 | 0 | 0.12 | 0.03 |
| | 3 | 0 | 0 | 0.13 | 0.03 |
| | 7 | 0 | 0 | 0.07 | 0.13 |
| | 15 | 0 | 0 | 0.22 | 0.07 |
| | 30 | 0 | 0 | 0.18 | 0.05 |
| | 60 | 0 | 0 | 0.1 | 0.15 |
| CNN+XGBoost (Soft Voting) | 1 | 0 | 0 | 0.15 | 0.05 |
| | 3 | 0 | 0 | 0.38 | 0.05 |
| | 7 | 0.53 | 0.33 | 0.5 | 0.05 |
| | 15 | 0.59 | 0.86 | 0.43 | 0.31 |
| | 30 | 0.71 | 0.67 | 0.56 | 0.32 |
| | 60 | 0.94 | 0.57 | 0.64 | 0.44 |
| CNN+XGBoost (Hybrid Fusion Model) | 3 | 0 | 0 | 0.81 | 0.58 |
| | 7 | 0 | 0 | 0.89 | 0.74 |
| | 15 | 0.38 | 0.8 | 0.88 | 0.74 |
| | 30 | 0.76 | 0.82 | 0.92 | 0.89 |
| | 60 | 1 | 0.86 | 0.92 | 0.96 |