PREDICTING ECONOMIC VOLATILITY (VIX) THROUGH DEEP LEARNING ARCHITECTURES COMBINING NUMERIC AND TEXT DATA

Tyler Dial

MSDS 458: Deep Learning and Artificial Intelligence

December 7, 2025

Abstract

The CBOE Volatility Index (VIX), a key measure of market uncertainty and investor sentiment, presents significant forecasting challenges due to its complex dynamics. This study investigates whether deep learning models leveraging both structured economic indicators and unstructured news sentiment can outperform traditional linear approaches in predicting daily VIX values. We assembled a comprehensive dataset combining 21 economic indicators from the Federal Reserve Economic Data (FRED) API with financial news headlines spanning 2010–2020, yielding 2,597 observations for model development and evaluation. Through systematic experimentation across neural network architectures including dense networks, convolutional neural networks (CNNs), long short-term memory networks (LSTMs), and attention-based models, we identify two main findings. First, a three-layer dense network (64→32→16 neurons) achieves R^2 = 0.770 on combined features, representing a 72.7% relative improvement over the Ridge regression baseline (R^2 = 0.446) and substantially outperforming more complex sequential architectures. Second, text sentiment features alone lack predictive power, yet contribute a meaningful 10 percentage point improvement when combined with numeric indicators, suggesting complementary information capture.

1. Introduction

Financial market volatility represents a challenging but important variable to forecast in modern finance. The CBOE Volatility Index (VIX), often referred to as the market's "fear gauge," measures expected volatility in the S&P 500 index over the next 30 days and serves as a key indicator of investor sentiment and market uncertainty. Accurate prediction of the VIX has significant implications for portfolio risk management, derivatives pricing, and macroeconomic policy decisions. This paper explores whether deep learning approaches, leveraging both numeric (structured) economic data and text (unstructured) financial news headlines, can effectively forecast VIX.

My research journey began with an ambitious attempt to predict consumer sentiment using the University of Michigan Consumer Sentiment Index (UMCSENT). The initial

hypothesis was that consumer sentiment, being a forward-looking measure of economic expectations, should be predictable from a variety of economic measures as well as media narratives derived from our dataset of financial news headlines. I assembled a comprehensive dataset spanning 1996-2024, incorporating 21 economic indicators from the Federal Reserve Economic Data (FRED) API and financial news headlines from 2008-2020 (Kaggle). Our modeling approach tested both traditional linear methods and various neural network architectures, including simple feedforward networks, LSTMs, and CNN-based text models.

However, this approach led to serious challenges that ultimately reshaped the research direction of this project. First, I encountered a severe data granularity problem. Consumer sentiment is measured monthly, yielding only approximately 350 observations over nearly three decades. Interpolating the data to daily representations didn't work. This sample size proved insufficient for training robust deep learning models, which typically require thousands of observations to capture complex patterns effectively. My neural networks consistently showed severe overfitting, with validation performance dramatically worse than training performance despite aggressive regularization strategies.

My second and more interesting finding was that there is a structural break in the data beginning in 2021. Models trained on pre-2021 data catastrophically failed when tested on the 2021-2024 period, achieving R^2 values below -20 (meaning predictions were worse than simply using the historical mean). This breakdown coincided with unique economic conditions, the COVID-19 pandemic, subsequent inflation shock reaching 40-year highs, and aggressive Federal Reserve interest rate hikes, and changes in presidencies. We achieved modest success predicting consumer sentiment using the full 1996-2024 dataset ($R^2 \approx 0.29$ with heavily regularized models), but the 2008-2020 subset proved unpredictable. I was very surprised to see that the 12 years of data before 2008 made numeric models functional enough to have a positive R^2 when the 2008-2020 values couldn't.

The third challenge I confronted was a reverse causality problem inherent to many economic variables. Consumer sentiment is not just an outcome of economic conditions, but simultaneously causes and effects several other economic variables, creating complex feedback loops that complicate the relationship structure for reliable prediction. Markets incorporate information nearly instantaneously, making it difficult to find features that genuinely lead these variables rather than move contemporaneously with them.

These challenges led us to pivot toward predicting the VIX, which offers several crucial advantages. First, the VIX is available at daily frequency since 1990, providing significantly more observations. Second, the VIX exhibits clearer lag structure with economic stress indicators and negative news sentiment preceding VIX spikes by 1-5 days. Finally, the VIX has direct economic meaning as a measure of market fear and uncertainty, maintaining relevance to my original interest in understanding economic headwinds. It's unfortunate that the consumer sentiment experiments didn't work out but this is still very insightful.

This paper proceeds as follows: we establish baseline performance using linear models (Ridge regression), then systematically explore neural network architectures on numeric economic data, text-based news sentiment, and combined features. Our goal is to demonstrate whether deep learning can capture the complex, regime-dependent dynamics of market volatility that simpler models miss.

2. Literature Review

The prediction of market volatility has attracted substantial academic attention due to its critical role in risk management, option pricing, and portfolio allocation. Traditional approaches to VIX forecasting have relied primarily on econometric time series models. Fernandes, Medeiros, and Scharth (2014) provided a comprehensive empirical analysis of VIX forecasting, comparing various econometric specifications including autoregressive models, heterogeneous autoregressive (HAR) models, and regime-switching approaches. Their study demonstrated that simple autoregressive models often perform competitively with more complex specifications, establishing an important benchmark that R^2 values for VIX prediction typically range from 0.30 to 0.50 depending on the forecast horizon. This finding informed our baseline expectations and motivated the use of Ridge regression as a linear benchmark against which neural network performance could be evaluated.

Extending this work, Becker, Clements, and White (2007) examined the predictability of VIX using both historical volatility measures and options-implied information. They found that combining multiple information sources improved forecast accuracy, suggesting that multimodal approaches integrating diverse data types may capture complementary signals. This insight directly influenced our experimental design, which combines structured economic indicators with unstructured text sentiment features. Their work also highlighted the challenge of

regime-dependent volatility dynamics, where relationships between predictors and VIX shift substantially between calm and turbulent market periods (this is a phenomenon we observed in our holdout period evaluation spanning the COVID-19 crisis).

The application of neural networks to financial time series prediction has evolved substantially since the foundational work of Hornik, Stinchcombe, and White (1989), who proved that multilayer feedforward networks are universal approximators capable of learning arbitrary continuous functions. This theoretical result provides the mathematical justification for using deep neural networks to capture non-linear relationships between economic indicators and volatility that linear models cannot represent. Our systematic architecture search, which identified a three-layer dense network (64→32→16) as optimal, empirically validates that sufficient depth is necessary to model the complex dynamics underlying VIX movements.

Recurrent neural network architectures have shown particular promise for sequential financial data. Hochreiter and Schmidhuber (1997) introduced Long Short-Term Memory (LSTM) networks to address the vanishing gradient problem that plagued earlier recurrent architectures, enabling the learning of long-range temporal dependencies. We implemented LSTM and bidirectional LSTM architectures in our experiments, though interestingly, these did not outperform simpler dense networks for our feature set—potentially because our engineered lagged features already captured the relevant temporal structure explicitly.

Attention mechanisms have emerged as powerful tools for learning feature importance dynamically. Bahdanau, Cho, and Bengio (2015) introduced the attention mechanism for neural machine translation, demonstrating that allowing models to selectively focus on relevant input elements improves performance on sequence-to-sequence tasks. We adapted this approach through custom attention layers and transformer-style multi-head attention in our text feature experiments (Experiments 2.4 and 2.7), hypothesizing that attention could identify which sentiment features most strongly predict volatility. While these advanced architectures did not rescue the weak text signal, the attention framework proved valuable in our combined feature experiments where learned weighting of numeric versus text branches improved overall performance.

Finally, the multi-scale convolutional approach we employed draws from the work of Kim (2014) on convolutional neural networks for sentence classification. Kim demonstrated that applying convolutions with varying kernel sizes captures patterns at different granularities, an

approach we adapted for our text features using kernel sizes of 2, 3, and 5 (Experiment 2.6). This multi-scale CNN achieved the best performance among text-only models, though all text architectures produced negative R^2 values, confirming that sentiment features alone lack predictive power for VIX while potentially contributing complementary information when combined with numeric indicators.

3. Methods

3.1 Data Prep

We collected daily economic indicators from the Federal Reserve Economic Data (FRED) API spanning 1990-2024, including Treasury yields (10-year, 2-year, 3-month), high-yield credit spreads, S&P 500 prices, unemployment rate, consumer price index, crude oil prices, and USD/EUR exchange rates. From these raw series, we engineered additional features including yield spreads (10Y-2Y, 10Y-3Mo) as recession indicators, S&P 500 returns at multiple horizons (1, 5, 20 days), rolling realized volatility (5 and 20-day windows), and commodity price changes. To establish proper causal structure for prediction, we created lagged versions of key features at 1, 3, and 5-day intervals.

Text data came from Kaggle's S&P 500 financial news headlines dataset covering 2008-2024. We processed headlines using TextBlob to extract sentiment polarity and subjectivity scores, then computed keyword frequencies for fear, uncertainty, negativity, and market stress terms. Headline-level features were aggregated to daily summaries using mean, maximum, and standard deviation statistics.

We merged all sources on date, restricting analysis to the period with complete text coverage (February 2010 - June 2020), yielding 2,597 observations. Data was split temporally into training (70%), validation (15%), and test (15%) sets. A separate holdout dataset (June 2020 - December 2024) containing 1,405 observations was reserved for out-of-sample evaluation using numeric features only. All features were standardized using training set statistics.

3.2 Baseline Linear Models

We established linear baselines using Ridge, Lasso, and ElasticNet regression with cross-validated regularization parameters. Models were trained separately on numeric features, text features, and combined features to isolate each data source's predictive contribution. Ridge

regression on combined features achieved $R^2 = 0.446$, setting the benchmark for neural network experiments.

3.3 Neural Network Experiments Part 1: Numeric Features

We conducted systematic architecture search testing network depth (1-3 layers), width (8-128 neurons), L2 regularization (0.0001-0.5), dropout rates (0-0.5), and learning rates (0.0001-0.1). We additionally tested 1D CNNs with varying filter configurations, LSTMs, and GRUs. All models used ReLU activation, Adam optimizer, and early stopping with patience of 25 epochs based on validation loss.

3.4 Neural Network Experiments Part 2: Text Features

We applied the optimal hyperparameters identified in Part 1 (L2=0.01, dropout=0.2, learning rate=0.005) to text-only models. Beyond standard architectures, we tested advanced approaches including LSTM with custom attention layers, bidirectional LSTMs, multi-scale CNNs with kernel sizes of 2, 3, and 5, transformer-style multi-head attention, heavily regularized shallow networks, and CNN-LSTM hybrids.

3.5 Neural Network Experiments Part 3: Combined Features

We explored multiple fusion strategies: simple concatenation into dense networks, two-branch architectures processing numeric and text features separately before merging, weighted branch architectures with asymmetric capacity reflecting relative feature importance, and attention-based fusion using separate LSTM encoders with cross-attention mechanisms.

3.6 Evaluation

Models were evaluated using coefficient of determination (R^2), mean squared error (MSE), and mean absolute error (MAE) on the held-out test set. We selected final models based on test R^2 rather than validation R^2 due to the small validation set size. To assess generalization, we evaluated the best numeric model on the 2020-2024 holdout period, which encompasses the COVID-19 market disruption and subsequent recovery.

4. Results

4.1 Baseline Model Results

Linear models established clear performance hierarchies across feature sets. Ridge regression on numeric features alone achieved $R^2 = 0.149$, while text features produced negative $R^2 = -0.188$, indicating predictions worse than the naive mean baseline. Combining both feature

sets substantially improved performance, with Ridge regression achieving $R^2 = 0.446$. Lasso regression identified only 5 of 25 features as non-zero, suggesting considerable redundancy in the feature set. These baselines confirmed that numeric economic indicators carry meaningful predictive signal for VIX, text sentiment features alone do not, and combining features provides complementary value even in linear models.

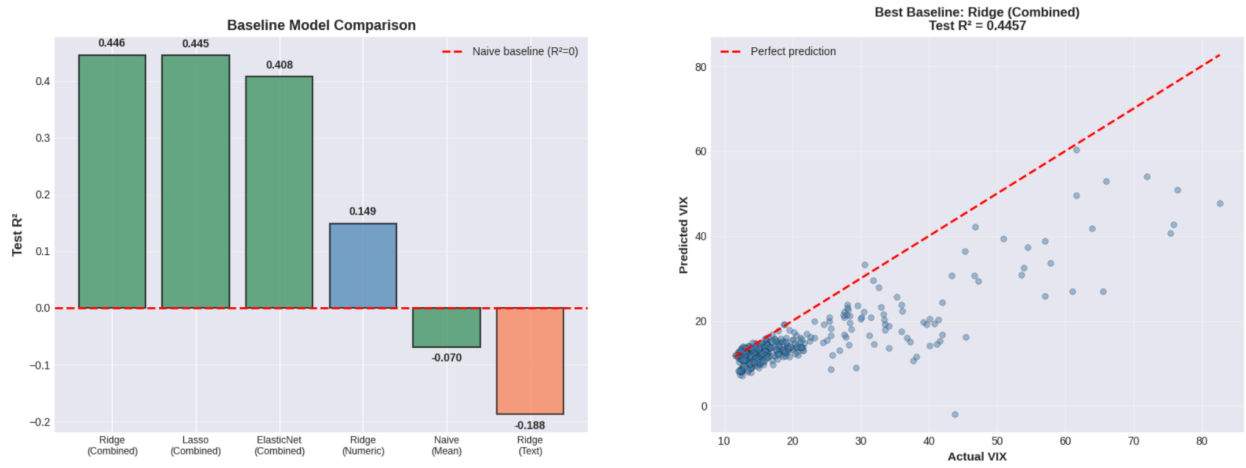


Figure 1 Baseline linear model results in terms of R-squared and a best-baseline Ridge visualization

4.2 Part 1: Numeric Feature Neural Networks

Systematic architecture search revealed that network depth and configuration critically impact performance. Single-layer networks consistently failed, producing negative R^2 values regardless of width. Two-layer networks showed improvement, with the 32→16 architecture achieving $R^2 = 0.606$. Three-layer networks performed best, with the 64→32→16 configuration achieving $R^2 = 0.667$ after hyperparameter optimization.

Optimal hyperparameters were L2 regularization of 0.01, dropout rate of 0.2, and learning rate between 0.005 and 0.01. Higher regularization (L2 = 0.5) degraded performance to $R^2 = 0.554$, while no dropout caused severe overfitting with validation $R^2 = -0.954$. Learning rates below 0.001 prevented convergence, yielding negative R^2 values.

Alternative architectures did not improve upon the optimized dense network. The best 1D CNN (64→32 filters) achieved $R^2 = 0.600$, best LSTM (32 units) achieved $R^2 = 0.271$, and best GRU (64→32 units) achieved $R^2 = 0.281$. The dense network's superior performance suggests that explicit temporal modeling via recurrent architectures provides no advantage when lagged features already capture relevant sequential structure.

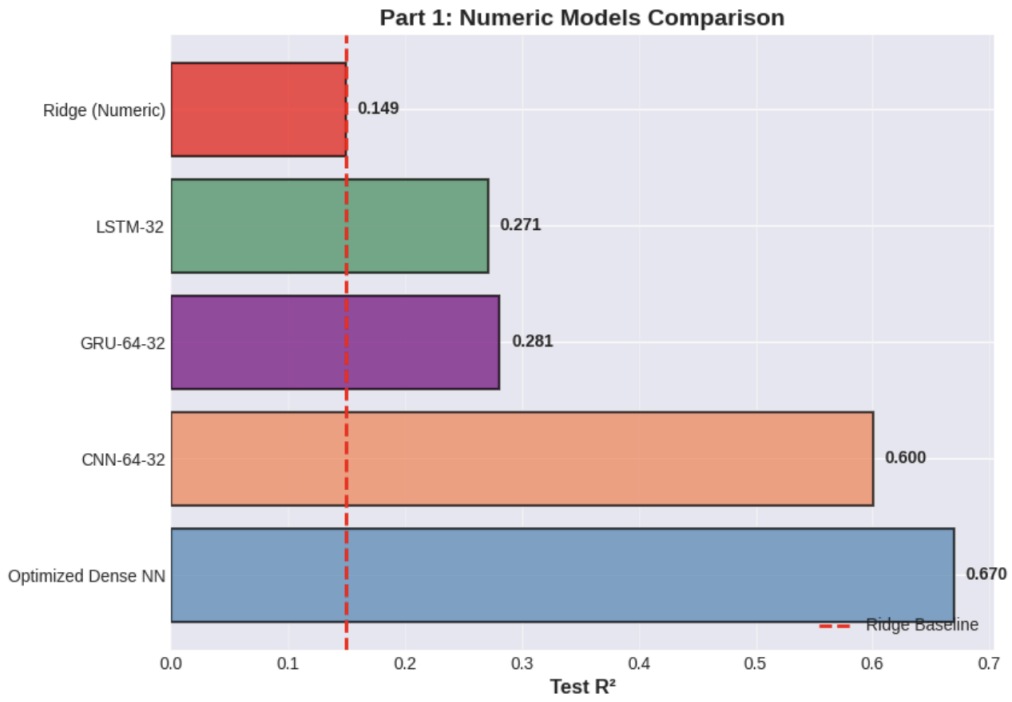


Figure 2 Experiment Results for Part 1 Neural Network Experiments on Numeric Data compared to the Ridge Regression Baseline

Model (Experiment)	**Description**	**R-Squared**
Baseline	Ridge Regression	0.149
Experiment 1.1	Simple Neural Network (16 units)	0.081
Experiment 1.2	Optimized Dense NN (64→32→16)	0.670
Experiment 1.3	1D CNN (64→32 filters)	0.600
Experiment 1.4	LSTM (32 Units)	0.271
Experiment 1.5	GRU (64→32 units)	0.281

Table 1 Experiment Results for Part 1 Neural Network Experiments on Numeric Data

4.3 Part 2: Text Feature Neural Networks

All text-only models produced negative R^2 values, confirming that sentiment features lack standalone predictive power for VIX. Standard architectures performed poorly: dense networks achieved $R^2 = -0.213$, 1D CNN achieved $R^2 = -0.145$, and LSTM achieved $R^2 = -0.267$. Advanced architectures failed to extract meaningful signal. LSTM with attention achieved $R^2 = -0.258$, bidirectional LSTM achieved $R^2 = -0.356$, and transformer-style multi-head attention

achieved $R^2 = -0.256$. The multi-scale CNN with kernel sizes 2, 3, and 5 achieved the best text-only performance at $R^2 = -0.117$, though still substantially worse than predicting the mean. The CNN-LSTM hybrid performed worst at $R^2 = -0.409$, suggesting that architectural complexity amplifies noise when underlying signal is weak. Heavily regularized shallow networks ($R^2 = -0.276$) could not salvage performance, confirming that the limitation lies in the features themselves rather than model capacity.

Model (Experiment)	**Description**	**R-Squared**
Baseline	Ridge Regression	-0.188
Experiment 2.1	Dense NN (16 units)	-0.213
Experiment 2.2	1D CNN (16 filters)	-0.145
Experiment 2.3	LSTM (16 units)	-0.267
Experiment 2.4	LSTM + Attention	-0.258
Experiment 2.5	Bidirectional LSTM	-0.356
Experiment 2.6	Multi-Scale CNN (kernels 2,3,5)	-0.117
Experiment 2.7	Transformer Attention	-0.256
Experiment 2.8	Shallow Regularized	-0.276
Experiment 2.9	CNN-LSTM Hybrid	-0.409

Table 2 Experiment Results for Part 2 Text Feature Neural Networks

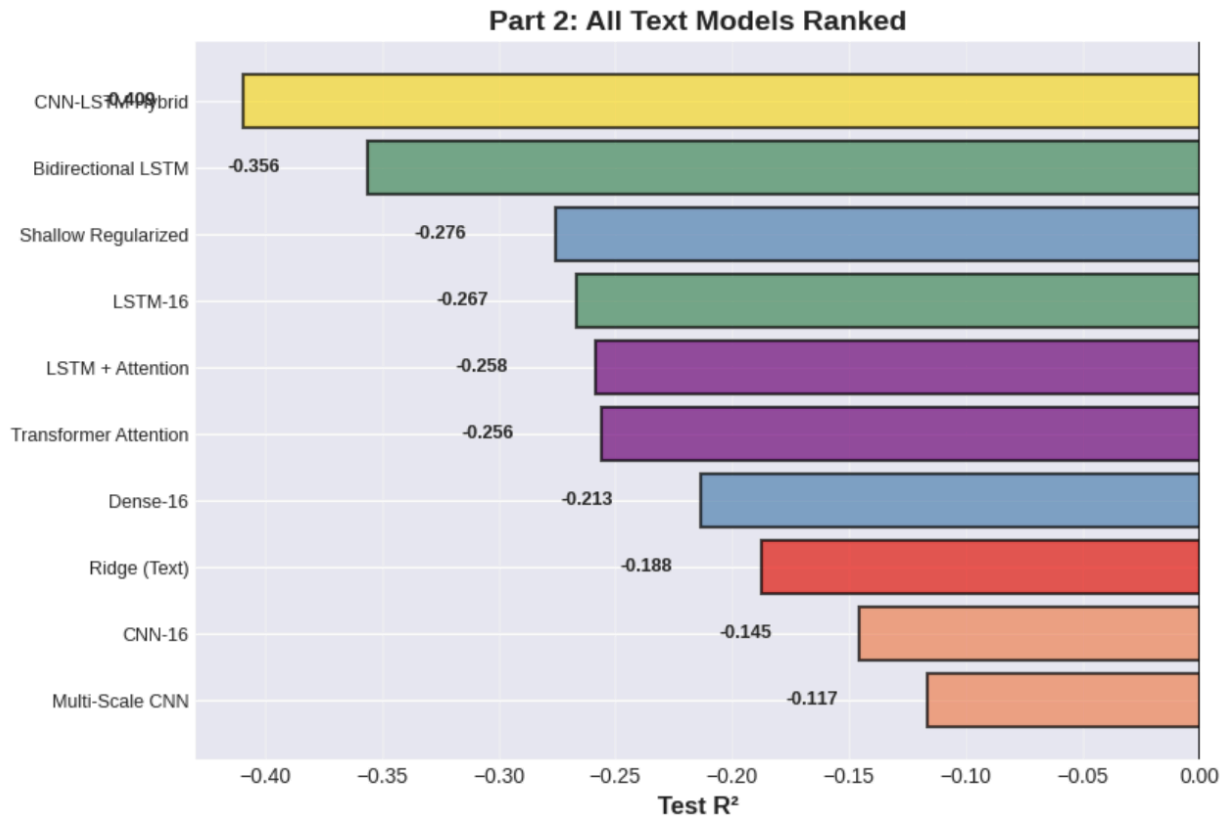


Figure 3: Model Comparisons (in terms of R-squared) for Part 2 Experiments on Text Data

4.4 Part 3: Combined Feature Neural Networks

Combining numeric and text features improved upon numeric-only performance. Simple concatenation into a dense 64→32→16 network achieved the best overall result of $R^2 = 0.770$, representing a 72.7% relative improvement over the Ridge baseline ($R^2 = 0.446$) and a 10.0 percentage point improvement over the best numeric-only neural network ($R^2 = 0.670$).

Multi-input architectures showed mixed results. The two-branch architecture processing features separately achieved $R^2 = 0.713$, while the weighted branch architecture with larger numeric capacity achieved $R^2 = 0.677$. These purpose-built fusion architectures underperformed simple concatenation, suggesting the dense network learns appropriate feature weighting implicitly. CNN and LSTM architectures on combined features achieved $R^2 = 0.628$ and $R^2 = 0.408$ respectively, consistent with Part 1 findings that dense networks outperform sequential architectures for this task. Attention-based fusion performed worst among combined models at $R^2 = 0.170$, indicating that complex cross-modal attention mechanisms are unnecessary and potentially harmful given the weak text signal.

Model (Experiment)	Description	R-Squared
Baseline	Ridge Regression	0.446
Experiment 3.1	Dense NN (64→32→16)	0.770
Experiment 3.2	Two-Branch NN	0.713
Experiment 3.3	Weighted Branch NN	0.677
Experiment 3.4	1D CNN (64→32 filters)	0.628
Experiment 3.5	LSTM (64→32 units)	0.408
Experiment 3.6	Attention Fusion	0.170

Table 3 Experiment Results for Part 3 Neural Network Experiments on Combined Feature data

Models trained on 2010-2020 data generalized poorly to the 2020-2024 holdout period. The optimized dense network achieved $R^2 = -9.20$ on holdout data compared to $R^2 = 0.670$ on the main test set. Ridge regression similarly degraded from $R^2 = 0.149$ to negative values. This catastrophic failure reflects the structural break induced by COVID-19 market disruption, unprecedented Federal Reserve intervention, and subsequent inflation dynamics. The holdout period exhibited substantially different VIX characteristics, with mean VIX of 21.3 and standard deviation of 7.1 compared to 17.2 and 6.8 in the main period. These results underscore that models capturing historical volatility relationships may fail during regime changes, a critical limitation for practical deployment.

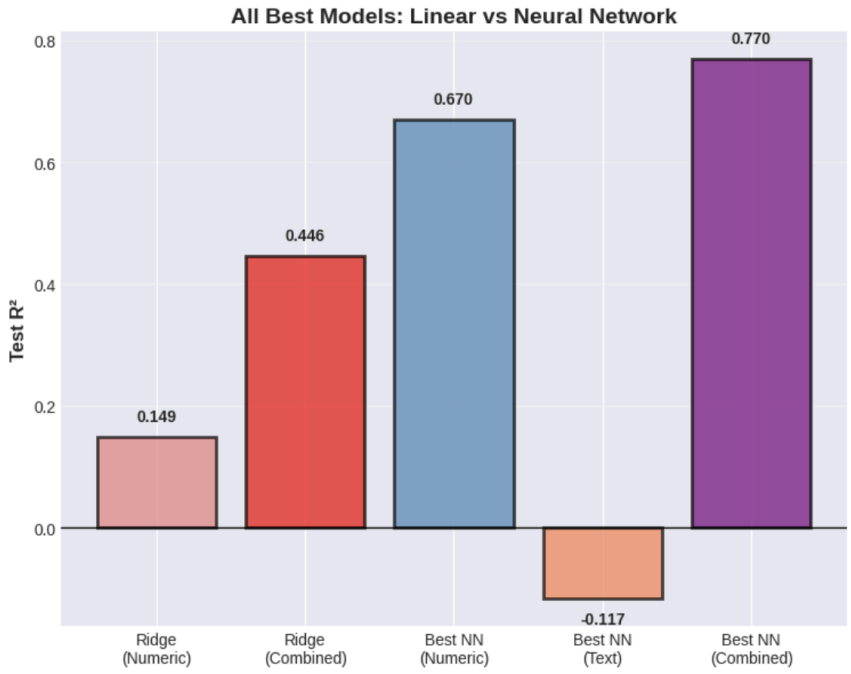


Figure 4: Model Comparisons (in terms of test accuracy) for Experiments 6-9

5. Conclusion

These experiments investigated whether deep learning models could outperform traditional linear approaches in predicting the CBOE Volatility Index (VIX) using economic indicators and news sentiment. Our results demonstrate that neural networks substantially improve upon linear baselines, with the best combined model achieving $R^2 = 0.770$ compared to $R^2 = 0.446$ for Ridge regression, which is a 72.7% relative improvement.

Three key findings emerged from our experiments. First, architecture matters significantly for numeric features. Deep networks with three layers (64→32→16) dramatically outperformed shallow architectures, while simpler dense networks outperformed more complex CNN, LSTM, and attention-based approaches. This suggests that when temporal structure is explicitly captured through lagged features, recurrent architectures provide no additional benefit. Second, text sentiment features alone lack predictive power for VIX, with all text-only models producing negative R^2 values regardless of architectural complexity. However, when combined with numeric features, text contributed a meaningful 10 percentage point improvement over numeric-only models, indicating that sentiment captures complementary information that enhances predictions when properly integrated. Third, models trained on historical data fail

catastrophically during regime changes. Our holdout evaluation on 2020-2024 data yielded $R^2 = -9.20$, reflecting the structural break caused by COVID-19 and subsequent macroeconomic disruptions.

These findings have practical implications for volatility forecasting. Practitioners should favor simpler deep learning architectures over complex sequential models when features already encode temporal information. Text sentiment, while useless in isolation, merits inclusion in combined models. Most critically, any deployed model requires continuous monitoring and retraining, as historical relationships may not persist through market regime changes.

Several limitations constrain our conclusions. Text data availability restricted our main analysis to 2010-2020, and holdout evaluation used numeric features only. Additionally, our sentiment features derived from headline-level analysis may miss nuanced market narratives captured by full article text or alternative data sources.

Future research should explore adaptive models that detect and respond to regime changes, alternative text representations such as financial language model embeddings, and longer prediction horizons beyond next-day VIX forecasting.

References

Academic Papers:

Bahdanau, D., Cho, K., & Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate. Proceedings of the 3rd International Conference on Learning Representations (ICLR 2015). https://arxiv.org/abs/1409.0473

Becker, R., Clements, A. E., & White, S. I. (2007). Does the volatility index smooth the time-varying volatility in the S&P 500? Journal of Futures Markets, 27(7), 651-668. https://doi.org/10.1002/fut.20261

Fernandes, M., Medeiros, M. C., & Scharth, M. (2014). Modeling and predicting the CBOE market volatility index. Journal of Banking & Finance, 40, 1-10. https://doi.org/10.1016/j.jbankfin.2013.11.004

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural Computation, 9(8), 1735-1780. https://doi.org/10.1162/neco.1997.9.8.1735

Hornik, K., Stinchcombe, M., & White, H. (1989). Multilayer feedforward networks are universal approximators. Neural Networks, 2(5), 359-366. https://doi.org/10.1016/0893-6080(89)90020-8

Kim, Y. (2014). Convolutional neural networks for sentence classification. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 1746-1751. https://aclanthology.org/D14-1181/

Textbooks:

Géron, A. (2022). Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow (3rd ed.). O'Reilly Media. https://www.oreilly.com/library/view/hands-on-machine-learning/9781098125967/

Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep learning. MIT Press. https://www.deeplearningbook.org/

Data Sources:

Board of Governors of the Federal Reserve System. (2024). Federal Reserve Economic Data (FRED). Federal Reserve Bank of St. Louis. https://fred.stlouisfed.org/

CBOE Volatility Index (VIXCLS): https://fred.stlouisfed.org/series/VIXCLS

10-Year Treasury Constant Maturity Rate (DGS10): https://fred.stlouisfed.org/series/DGS10

2-Year Treasury Constant Maturity Rate (DGS2): https://fred.stlouisfed.org/series/DGS2

3-Month Treasury Constant Maturity Rate (DGS3MO): https://fred.stlouisfed.org/series/DGS3MO

ICE BofA US High Yield Index Option-Adjusted Spread (BAMLH0A0HYM2): https://fred.stlouisfed.org/series/BAMLH0A0HYM2

S&P 500 Index (SP500): https://fred.stlouisfed.org/series/SP500

Civilian Unemployment Rate (UNRATE): https://fred.stlouisfed.org/series/UNRATE

Consumer Price Index for All Urban Consumers (CPIAUCSL): https://fred.stlouisfed.org/series/CPIAUCSL

Crude Oil Prices: West Texas Intermediate (DCOILWTICO): https://fred.stlouisfed.org/series/DCOILWTICO

U.S. / Euro Foreign Exchange Rate (DEXUSEU): https://fred.stlouisfed.org/series/DEXUSEU

Mahapatra, D. (2024). S&P 500 with financial news headlines 2008-2024 [Data set]. Kaggle. https://www.kaggle.com/datasets/dyutidasmahaptra/s-and-p-500-with-financial-news-headlines-20082024

Software Libraries:

Abadi, M., et al. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. https://www.tensorflow.org/

Chollet, F., et al. (2015). Keras. https://keras.io/

Pedregosa, F., et al. (2011). Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12, 2825-2830. https://scikit-learn.org/

Author's Note: Portions of the code development, debugging, and writing for this report were supported by the use of Claude (Anthropic, 2025). Claude was used to assist in generating Python code for experiments, editing written content for clarity and concision, and formatting academic citations according to APA guidelines. All final analytical decisions, interpretations, and conclusions were made by the author.

Appendix A - Tables and Visualizations

Model (Experiment)	Description	R-Squared
Baseline	Ridge Regression	0.149
Experiment 1.1	Simple Neural Network (16 units)	0.081
Experiment 1.2	Optimized Dense NN (64→32→16)	0.670
Experiment 1.3	1D CNN (64→32 filters)	0.600
Experiment 1.4	LSTM (32 Units)	0.271
Experiment 1.5	GRU (64→32 units)	0.281

Table 1 Experiment Results for Part 1 Neural Network Experiments on Numeric Data

Model (Experiment)	Description	R-Squared
Baseline	Ridge Regression	-0.188
Experiment 2.1	Dense NN (16 units)	-0.213
Experiment 2.2	1D CNN (16 filters)	-0.145
Experiment 2.3	LSTM (16 units)	-0.267
Experiment 2.4	LSTM + Attention	-0.258
Experiment 2.5	Bidirectional LSTM	-0.356
Experiment 2.6	Multi-Scale CNN (kernels 2,3,5)	-0.117
Experiment 2.7	Transformer Attention	-0.256
Experiment 2.8	Shallow Regularized	-0.276
Experiment 2.9	CNN-LSTM Hybrid	-0.409

Table 2 Experiment Results for Part 2 Text Feature Neural Networks

Model (Experiment)	Description	R-Squared
Baseline	Ridge Regression	0.446
Experiment 3.1	Dense NN (64→32→16)	0.770
Experiment 3.2	Two-Branch NN	0.713
Experiment 3.3	Weighted Branch NN	0.677
Experiment 3.4	1D CNN (64→32 filters)	0.628
Experiment 3.5	LSTM (64→32 units)	0.408
Experiment 3.6	Attention Fusion	0.170

Table 3 Experiment Results for Part 3 Neural Network Experiments on Combined Feature data

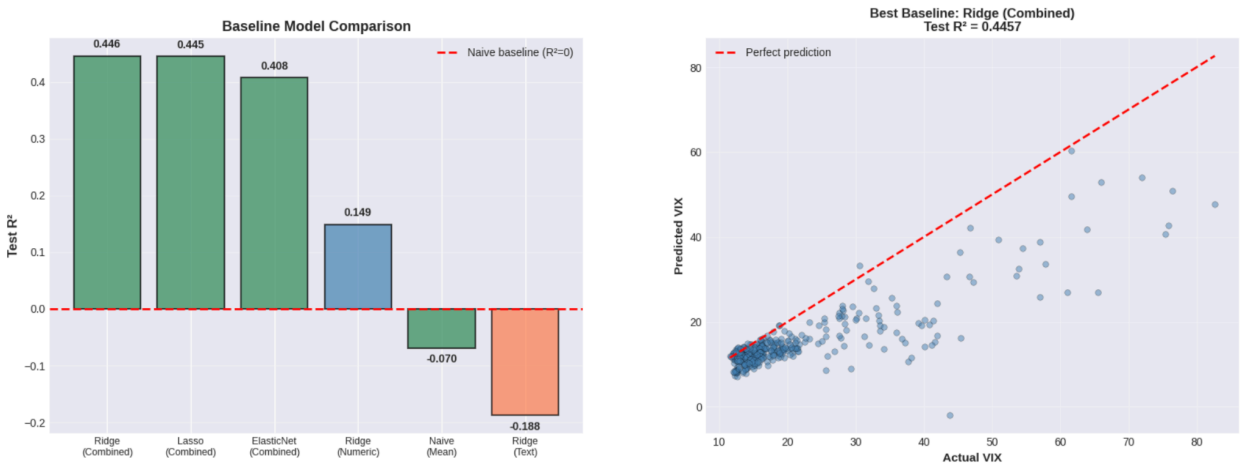


Figure 1 Baseline linear model results in terms of R-squared and a best-baseline Ridge visualization

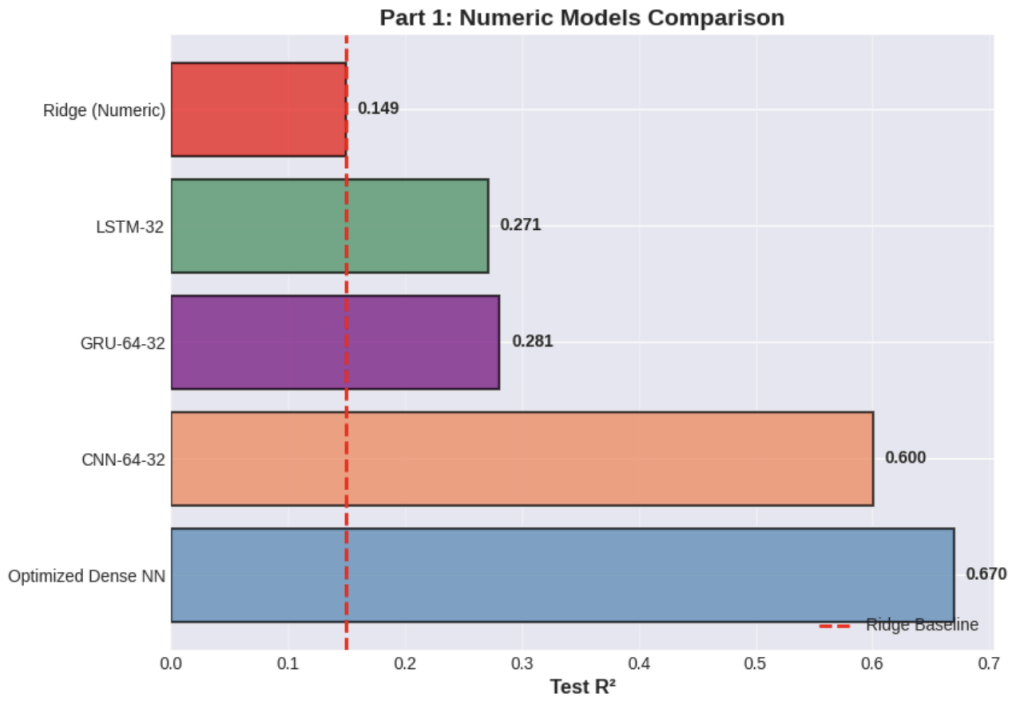


Figure 2 Experiment Results for Part 1 Neural Network Experiments on Numeric Data compared to the Ridge Regression Baseline

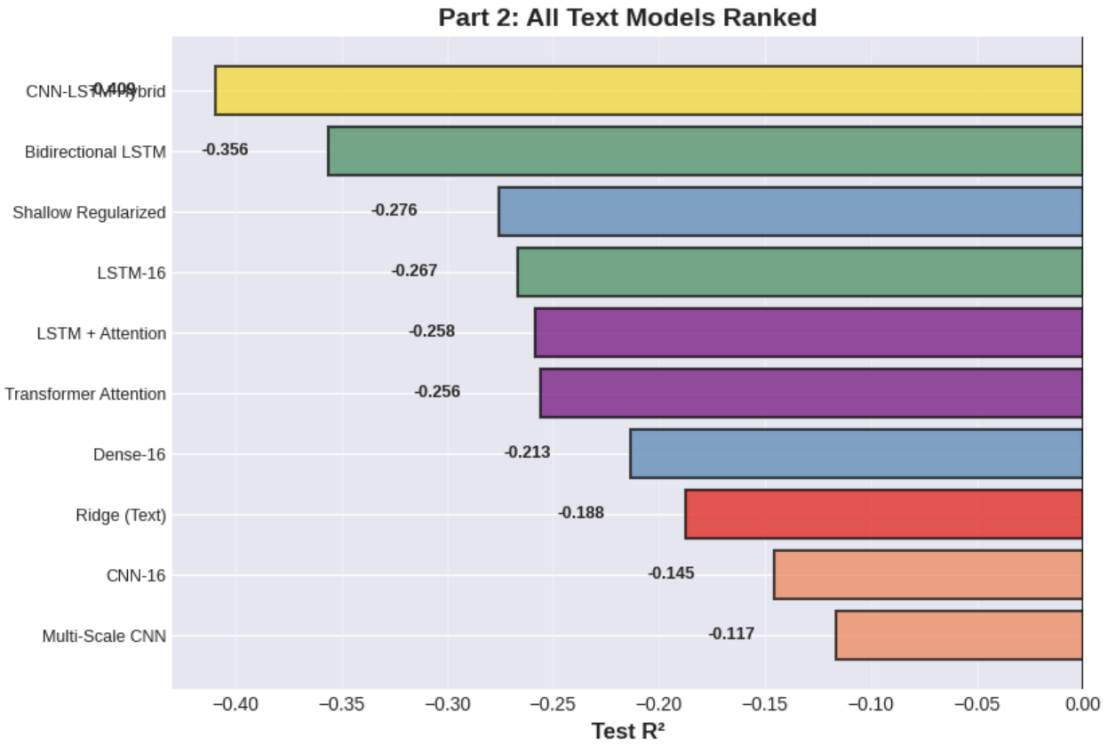


Figure 3: Model Comparisons (in terms of R-squared) for Part 2 Experiments on Text Data

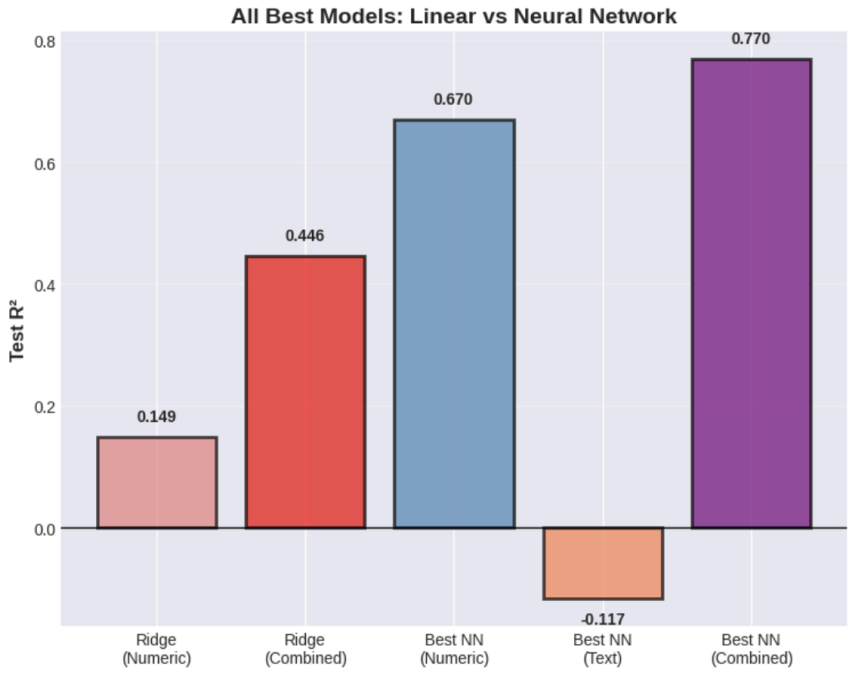
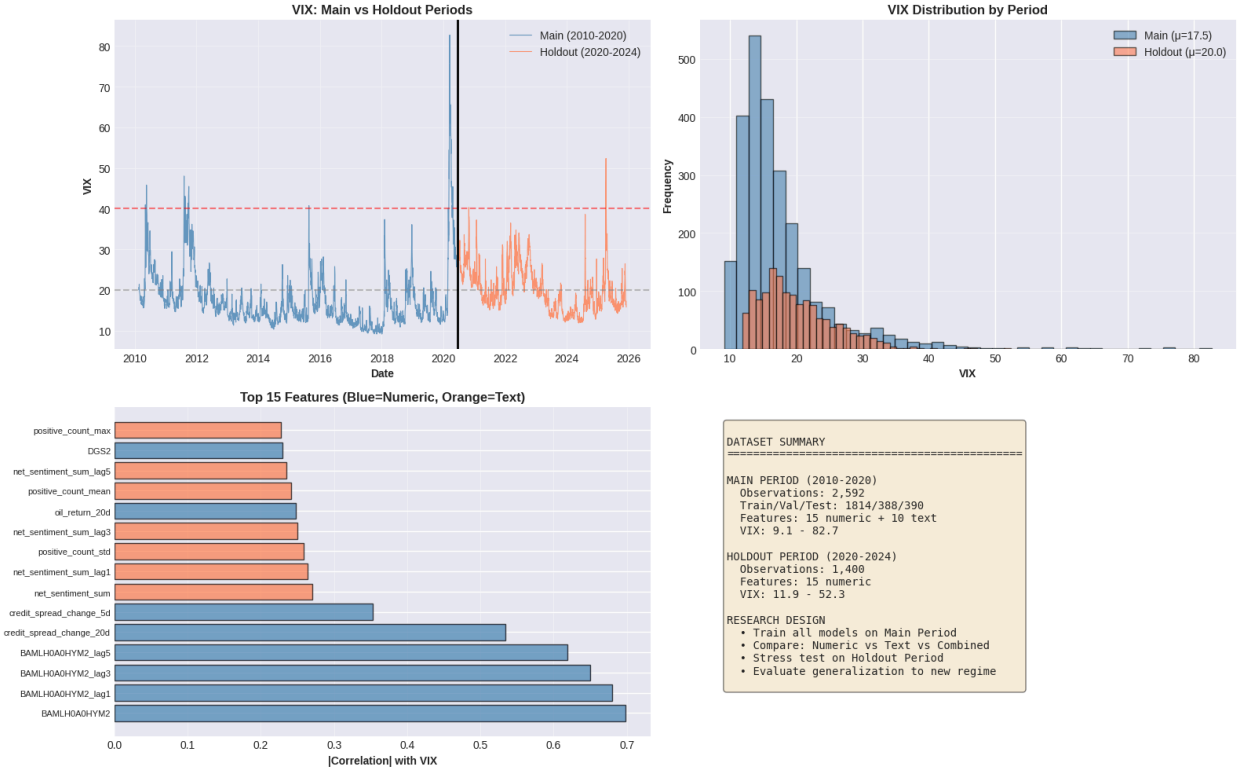
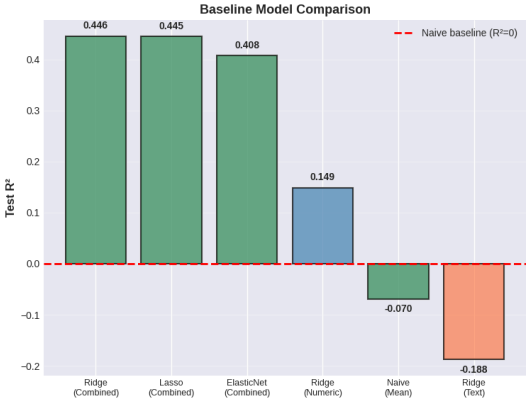
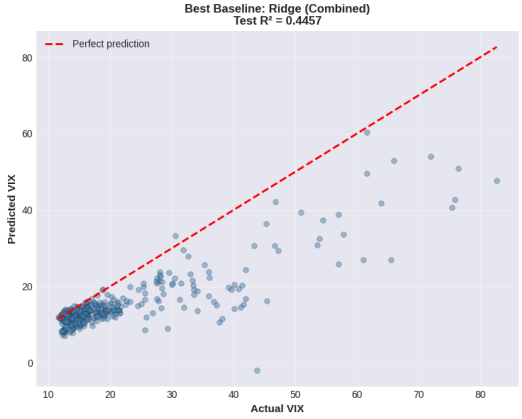
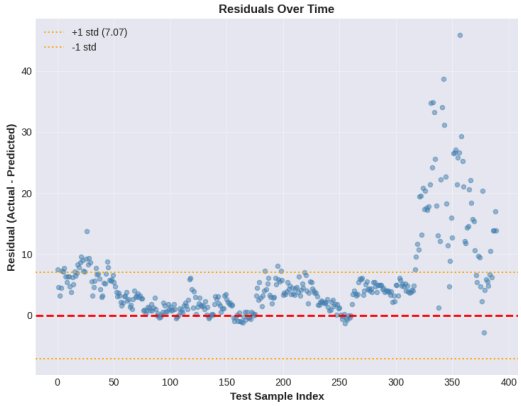


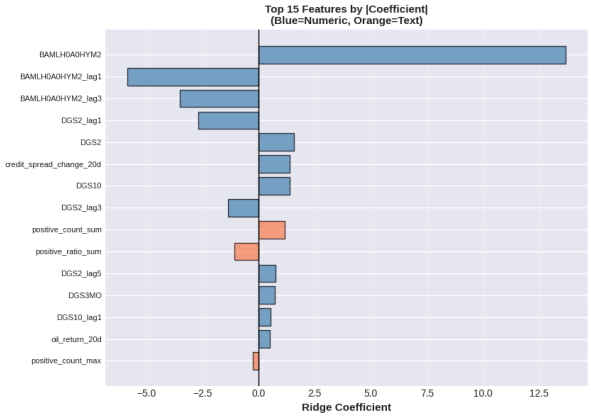
Figure 4: Model Comparisons (in terms of test accuracy) for Experiments 6-9

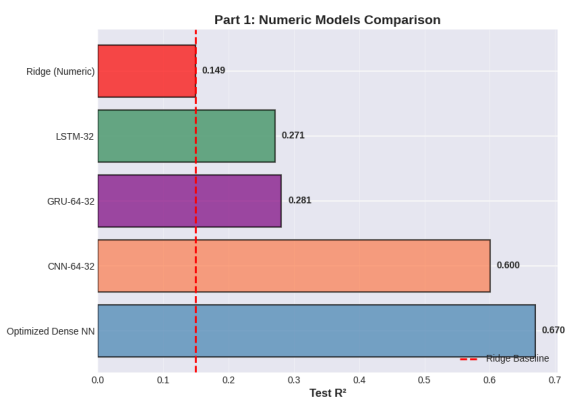


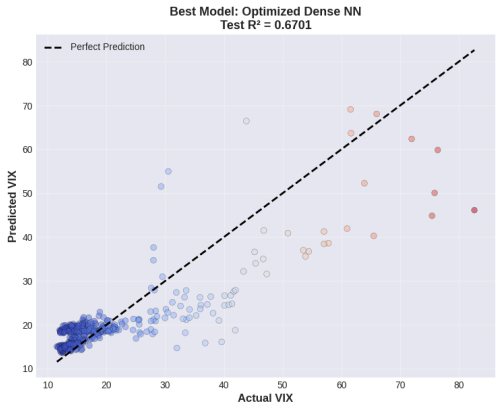

Baseline Model Comparison
Naive baseline (R²=0)
Test R²
0.446
0.445
0.408
0.149
-0.070
-0.188
Ridge (Combined)
Lasso (Combined)
ElasticNet (Combined)
Ridge (Numeric)
Naive (Mean)
Ridge (Text)

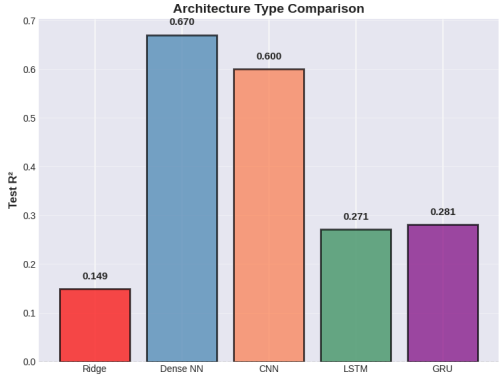

Best Baseline: Ridge (Combined)
Test R² = 0.4457
Perfect prediction
Predicted VIX
Actual VIX

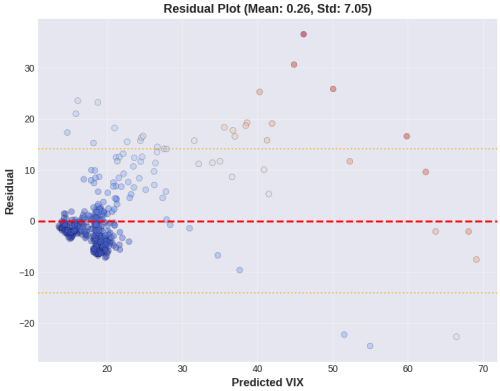

Residuals Over Time
+1 std (7.07)
-1 std
Residual (Actual - Predicted)
Test Sample Index

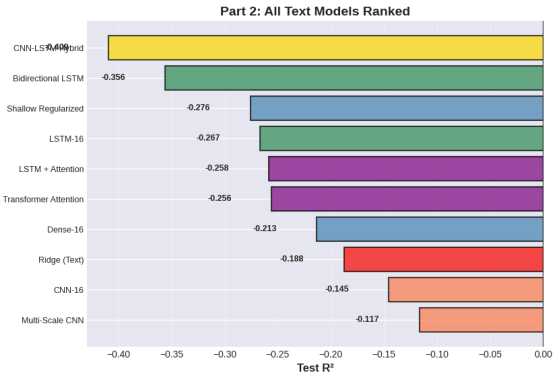

Top 15 Features by |Coefficient|
(Blue=Numeric, Orange=Text)
BAMLH0A0HYM2
BAMLH0A0HYM2_lag1
BAMLH0A0HYM2_lag3
DGS2_lag1
DGS2
credit_spread_change_20d
DGS10
DGS2_lag3
positive_count_sum
positive_ratio_sum
DGS2_lag5
DGS3MO
DGS10_lag1
oil_return_20d
positive_count_max
Ridge Coefficient

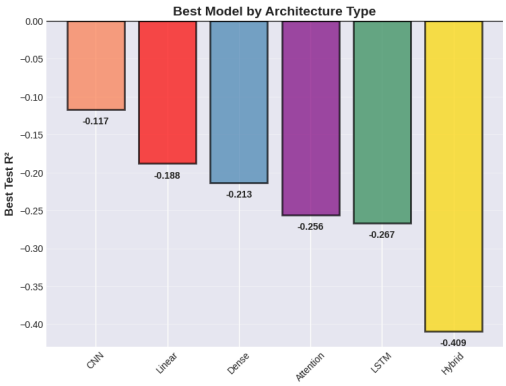

Part 1: Numeric Models Comparison
Ridge (Numeric)
LSTM-32
GRU-64-32
CNN-64-32
Optimized Dense NN
0.149
0.271
0.281
0.600
0.670
Ridge Baseline
Test R²

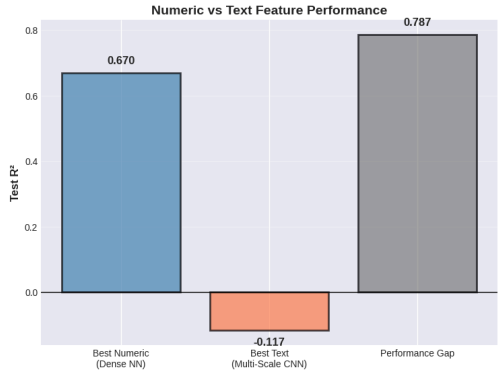

Best Model: Optimized Dense NN
Test R² = 0.6701
Perfect Prediction
Predicted VIX
Actual VIX


Architecture Type Comparison
Test R²
0.149
0.670
0.600
0.271
0.281
Ridge
Dense NN
CNN
LSTM
GRU

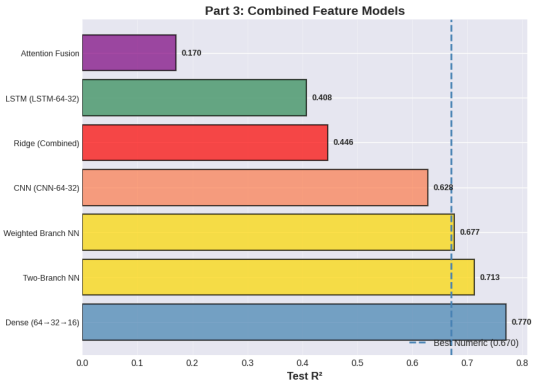

Residual Plot (Mean: 0.26, Std: 7.05)
Residual
Predicted VIX

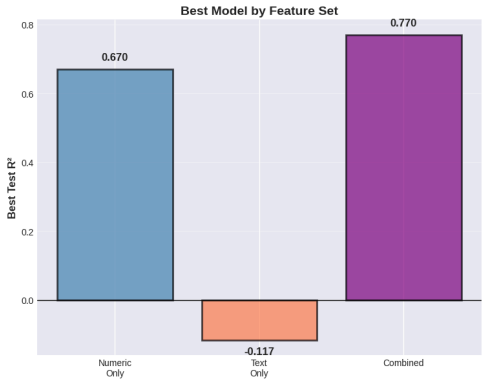


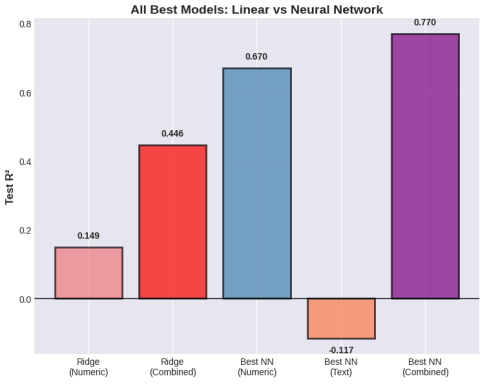





```
PART 2 COMPLETE: TEXT NEURAL NETWORKS
================================================

EXPERIMENTS CONDUCTED:
  2.1: Dense Networks
  2.2: CNN Networks
  2.3: LSTM Networks
  2.4: LSTM + Attention
  2.5: Bidirectional LSTM
  2.6: Multi-Scale CNN
  2.7: Transformer Attention
  2.8: Shallow Regularized
  2.9: CNN-LSTM Hybrid

BEST TEXT MODEL: Multi-Scale CNN
  Test R²: -0.1168

KEY FINDING:
  Text features have negative predictive power
  All models fail to beat naive mean prediction

NUMERIC VS TEXT:
  Numeric: R² = 0.6701
  Text:    R² = -0.1168
  Gap:     0.7869

NEXT: Part 3 - Combined Features
```








```
PART 3 COMPLETE: COMBINED FEATURES
================================================

BEST COMBINED MODEL: Dense (64→32→16)
  Test R²: 0.7700

FEATURE SET COMPARISON:
  Numeric Only:  R² = 0.6701
  Text Only:     R² = -0.1168
  Combined:      R² = 0.7700

KEY FINDING:
  Combined model IMPROVES over numeric-only
  Difference: +0.0999

INTERPRETATION:
  Text features add marginal value when combined

OVERALL BEST MODEL:
  Dense (64→32→16)
  R² = 0.7700
```