Deep Learning Final Project Proposal

Multi-Modal Stock Market Prediction: Integrating Technical Analysis with Financial News Sentiment

Group Number: 5

Team Members:

Adarsh Singh
Venkatesh Nagarjuna
Mayur Patil

Course: DATS 6303 - Deep Learning (Fall 2025)

Instructor: Dr. Amir Jafari

Date: November 7, 2025

GitHub Repository: https://github.com/drsh0755/FinalProject-Group5

Problem Selection and Motivation

Stock price prediction remains challenging due to markets being influenced by both quantitative patterns and qualitative information. Traditional approaches focus on either technical analysis (price history) or sentiment analysis (news) in isolation, missing valuable complementary signals. We propose a multi-modal deep learning system that fuses technical indicators (LSTM), financial news sentiment (Transformers), and market context (Dense Networks) to improve prediction accuracy.

Our project employs a two-phase validation approach: (1) historical backtesting on 2020-2025 data to demonstrate the model works, and (2) live deployment starting mid-December to validate performance on truly unseen future data. This dual approach provides both rigorous academic validation and practical real-world testing.

This problem is compelling because: (1) it demonstrates real-world application of multiple architectures covered in this course, (2) professional traders use multi-factor analysis that AI should replicate, and (3) it addresses the limitation of single-source prediction models by synthesizing diverse information streams. Our approach directly mirrors the methodology in Akita et al. (2016), who demonstrated that combining numerical and textual information significantly outperforms single-source models.

Database/Dataset Description

Our multi-modal approach requires three distinct data streams, each serving a specific purpose in the prediction pipeline:

Stream 1: Technical Analysis Data (LSTM Input)

Data Element	Source	Link	Coverage	Samples
Stock Prices	Yahoo Finance (yfinance)	https://pypi.org/project/yfinance/	Jan 2020 - Nov 2025	1,460 days/stock

- Stocks Selected: AAPL, TSLA, JPM, MSFT, GOOGL
- Raw Features: OHLCV (Open, High, Low, Close, Volume) | Daily 5 features/day
- Technical Indicators: pandas_ta | https://pypi.org/project/ft-pandas-ta/ | 20 indicators/day

Technical Indicators Calculated:

- Moving Averages: SMA (5, 20, 50, 200-day), EMA (12, 26-day)
- Momentum: RSI (14-day), MACD (12-26-9), Stochastic Oscillator
- Volatility: Bollinger Bands (upper, middle, lower), ATR
- Volume: On-Balance Volume (OBV)
- Returns: Daily return, 5-day return, volatility (20-day rolling std)

Input Shape: 60-day sequence × 20 features = (60, 20) per sample

Total Samples: 5 stocks × 1,460 days = 7,300 samples

Stream 2: News Sentiment Data (Transformer Input)

Data Element	Source	Link	Coverage	Volume
Historical News	Kaggle: Daily Financial News	https://www.kaggle.com/datasets/miguelaenlle/massive-stock-news-analysis-db-for-nlpbacktests	2009-2020	~1.75M headlines
Recent News	Alpha Vantage News API	https://www.alphavantage.co/documentation/#news-sentiment	2024-2025	~350 articles
Fine-tuning Data	Financial Phrase Bank	https://huggingface.co/datasets/takala/financial_phrasebank	Labeled sentences	4,840 sentences
Sentiment Model	FinBERT (Pre-trained)	https://huggingface.co/ProsusAI/finbert	-	-

Data Processing Pipeline:

1. Download Kaggle dataset (CSV format: Date, Stock, Headline, Source)
2. Filter for 5 target stocks (AAPL, TSLA, JPM, MSFT, GOOGL)
3. Fine-tune FinBERT on Financial Phrase Bank (optional, can use pre-trained)
4. Generate 256-dimensional sentiment embeddings per trading day
5. Aggregate multiple headlines per day using average pooling
6. Handle missing news days: forward-fill with previous day's sentiment

Input Shape: 256-dimensional embedding per trading day

Coverage: 1,460 days aligned with stock prices

Stream 3: Market Context Data (Dense Network Input)

Data Element	Source	Link	Coverage	Features
Market Indices	Yahoo Finance (yfinance)	https://pypi.org/project/yfinance/	Jan 2020 - Nov 2025	5 indices
Economic	FRED	https://fred.stlouisfed.org/docs/api/fred/	Jan 2020 -	5

Indicators	(Federal Reserve)		Nov 2025	indicators

Market Indices (Daily):

- S&P 500 Index (^GSPC): Broad market benchmark
- VIX Volatility Index (^VIX): Market fear gauge
- Technology Sector ETF (XLK): Tech sector performance
- Financial Sector ETF (XLF): Financial sector performance
- US Dollar Index (DX-Y.NYB): Currency strength

Economic Indicators (Daily/Monthly interpolated):

- Federal Funds Rate (FRED: DFF): Monetary policy
- 10-Year Treasury Yield (FRED: DGS10): Risk-free rate
- Consumer Price Index (FRED: CPIAUCSL): Inflation
- Unemployment Rate (FRED: UNRATE): Economic health
- Industrial Production Index (FRED: INDPRO): Manufacturing activity

Input Shape: 10 features per trading day

Coverage: 1,460 days aligned with stock prices

Dataset Size Assessment

Metric	Value
Total Samples	7,300 (5 stocks × 1,460 days)
Training Set (70%)	5,110 samples
Validation Set (15%)	1,095 samples
Test Set (15%)	1,095 samples

All data sources are FREE with no cost barriers. Multi-modal nature provides rich feature representations (286 total features). Regularization techniques prevent overfitting.

Why This is Sufficient:

- Multi-modal nature provides rich feature representations (20 + 256 + 10 = 286 features)
- Temporal sequences (60-day windows) add effective samples
- Regularization techniques (dropout, batch norm, early stopping) prevent overfitting
- Comparable to Akita et al. (2016) who used 50 companies × ~200 days

Data Accessibility: All sources are FREE with no cost barriers:

- Yahoo Finance: Unlimited API calls
- Kaggle: Free download (requires account)

- Alpha Vantage: 25 calls/day free tier (sufficient)
- FRED: Unlimited API calls (free key)

Deep Learning Network Architecture (Initial Stage)

Custom Three-Stream Architecture with Late Fusion:

Stream 1: Technical Analysis (LSTM Network)

- 3-layer stacked LSTM (128 units per layer)
- Input: 60-day sequences of 20 technical indicators
- Dropout: 0.2 between layers
- Layer normalization after final LSTM
- Output: 128-dimensional temporal features
- *Course Coverage: Week 10 - LSTM/GRU*

Stream 2: News Sentiment Analysis (Transformer)

- Pre-trained FinBERT (BERT fine-tuned for financial text)
- Freeze first 12 layers (transfer learning)
- Fine-tune last 12 layers (optional)
- Projection: 768 → 256 dimensions, ReLU, dropout 0.3
- Output: 256-dimensional sentiment features
- *Course Coverage: Week 11 - Transformers*

Stream 3: Market Context (Dense Network)

- 2-layer fully connected network
- Architecture: 10 → 64 (ReLU, BatchNorm, Dropout 0.3) → 64 (ReLU)
- Output: 64-dimensional market context features
- *Course Coverage: Week 6 - Training Deep Networks*

Fusion Architecture

- Concatenate all streams: [128 + 256 + 64] = 448 dimensions
- Fusion layers: 448 → 256 (ReLU, BatchNorm, Dropout 0.4) → 128 (ReLU, BatchNorm, Dropout 0.3)
- Multi-task output heads:
  - Price prediction: Linear(128 → 1) for regression
  - Direction classification: Linear(128 → 2) + Softmax for up/down
  - Volatility estimation: Linear(128 → 1) for regression
- Combined loss: 0.4 × MSE(price) + 0.4 × CrossEntropy(direction) + 0.2 × MSE(volatility)

Why Custom: Integrates standard components (LSTM, BERT) with novel late fusion and multi-task learning. Not available as pre-built model.

Course Coverage: Week 9 - PyTorch Custom Dataloaders for multi-input handling

Framework Selection: PyTorch

Rationale:

1. Multi-Modal Support: Easy implementation of custom DataLoader for 3 input streams (Week 9 material)
2. FinBERT Integration: Native Hugging Face transformers library support
3. Flexibility: Dynamic computation graphs ideal for custom fusion architectures
4. Debugging: Pythonic, can use standard Python debugger

Alternative Considered: TensorFlow/Keras functional API would work but requires more boilerplate for multi-input models and lacks seamless FinBERT integration.

Reference Materials

Core Academic Papers :

1. Akita, R., Yoshihara, A., Matsubara, T., & Uehara, K. (2016)
"Deep learning for stock prediction using numerical and textual information"
2016 IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS), pp. 1-6
DOI: 10.1109/ICIS.2016.7550882
Links:

- IEEE Xplore: https://ieeexplore.ieee.org/document/7550882
- ResearchGate (Free): https://www.researchgate.net/publication/306925671

2. Araci, D. (2019)
"FinBERT: Financial Sentiment Analysis with Pre-trained Language Models"
arXiv preprint arXiv:1908.10063
Link: https://arxiv.org/abs/1908.10063

3. Ngiam, J., Khosla, A., Kim, M., Nam, J., Lee, H., & Ng, A. Y. (2011)
"Multimodal deep learning"
Proceedings of ICML 2011, pp. 689-696
Link: https://people.csail.mit.edu/khosla/papers/icml2011_ngiam.pdf

Technical Documentation:

- FinBERT: https://huggingface.co/ProsusAI/finbert
- PyTorch: https://pytorch.org/docs/stable/

- yfinance: https://pypi.org/project/yfinance/
- Alpha Vantage: https://www.alphavantage.co/documentation/
- FRED API: https://fred.stlouisfed.org/docs/api/fred/

Performance Metrics

Primary Metrics:

1. Directional Accuracy: % correct up/down predictions (target: 60-62% vs 50% random)
2. Price Error: RMSE (target: <$3.00), MAE, MAPE
3. Trading Performance: Sharpe ratio (target: >0.8), total returns vs buy-and-hold

Ablation Studies: Systematically compare:

- LSTM only, Transformer only, Dense only
- All two-stream combinations
- Full three-stream model (expected best: 60-62% accuracy)

Project Schedule

Total Duration: November 10 - December 8, 2025 (4 weeks)

Pre-Work (Nov 7-9):

Download all datasets (stock prices, news, market data) and set up Python environment

Week 1 (Nov 10-16): Core Implementation

Data preprocessing, implement 3-layer LSTM, train baseline model

Week 2 (Nov 17-23): Multi-Modal Integration

Process news sentiment with FinBERT, implement fusion architecture, train full model on GPU

Week 3 (Nov 24-30): Evaluation & Deployment Setup

Comprehensive testing, ablation studies, AND deploy real-time system (Nov 25-30 to start collecting live predictions)

Target: Backtesting complete + live system deployed and collecting data by Dec 1

Week 4 (Dec 1-8): Live Data Collection & Finalization (Tentative/ Future Scope)

- Dec 1: Streamlit dashboard
- Dec 4-5: Continue live monitoring (Day 4-5), start analyzing results, begin final report
- Dec 6: Collect Day 6-7 predictions, complete live vs backtest analysis, finish final report
- Dec 7 evening: Finalize individual reports
- Dec 8: FINAL SUBMISSION & PRESENTATION (includes 5-7 days of live predictions from Dec 1-7)

- Minimum 3 days of live data acceptable (proves concept)

- Manual data collection if automation fails (backup plan)

- Reduce to 3 stocks instead of 5 (fewer API calls)

- Use local computer instead of GCP (eliminates server issues)

If real-time deployment faces issues:

Contingency Plans: If behind schedule, reduce to 3 stocks or 2-stream model (LSTM + News)

Expected Outcomes

Technical Goals:

- Directional accuracy: 60-62% (vs 56% LSTM-only baseline)
- RMSE: <$3.00
- Sharpe ratio: >0.8
- Demonstrate multi-modal superiority via ablation studies
- Demonstrate automated pipeline working reliably
- Compare backtest vs live performance, analyze any degradation
- Live directional accuracy: 55-58% (demonstrating generalization)
- Collect 5-7 days of live predictions before Dec 8 submission

Learning Objectives:

- Master multi-modal deep learning
- Gain practical LSTM, Transformer, PyTorch experience
- Develop end-to-end ML pipeline skills
- Create professional deliverables

Real-Time Deployment and Live Validation (Tentative/ Future Scope)

Our project includes live deployment to validate the model on truly unseen future data. After completing historical backtesting, we will deploy the trained model to make daily predictions, providing real-world validation of our multi-modal architecture.

Daily Prediction System

Automated Pipeline (Weekdays 4:30 PM EST):

- Data Collection: Fetch closing prices, news headlines, market indices via APIs
- Feature Engineering: Calculate technical indicators, process news with FinBERT
- Prediction: Load trained model, generate next-day price/direction predictions
- Storage: Log predictions in SQLite database with timestamp
- Notification: Send daily predictions via Telegram/Email

Next-Day Verification

Accuracy Tracking (Next day 4:35 PM):

- Fetch actual results, compare with predictions
- Calculate daily/rolling accuracy metrics
- Update performance database, send accuracy report

Live Performance Metrics

- Daily directional accuracy (up/down prediction correctness)
- 7-day and 30-day rolling accuracy
- RMSE, MAE on price predictions
- Cumulative hypothetical trading returns

Real-Time Deployment Timeline

Nov 25-30: Deploy system, test pipeline, begin daily predictions

Dec 1-6: Collect 5-7 trading days of live predictions

Dec 7: Analyze live vs backtest performance, finalize report

Dec 8: Submit project with BOTH historical AND live results (5-7 days of live data)

Expected Live Performance

With only 5-7 days of live predictions, we expect directional accuracy of 50-60% (small sample size increases variance). The key deliverable is demonstrating the automated system works reliably, not achieving high accuracy on limited data. A longer deployment period would provide more stable metrics.

Conclusion

This project applies state-of-the-art deep learning to stock market prediction by integrating LSTM, Transformers, and dense networks in a multi-modal framework. Our two-phase approach—historical backtesting (2020-2025) followed by live deployment (December 2025)—provides comprehensive validation. With a realistic 4-week timeline, free data sources, automated daily predictions, and clear contingency plans, we are confident in delivering a high-quality submission that demonstrates both academic rigor and practical real-world applicability.