STOCK MARKET PREDICTION USING NEURAL NETWORKS

- Sheela Hansda (hansda.s@northastern.edu)
- Kaushiki Ambi (ambi.k@northeastern.edu)
- Fatima Salim (salim.f@northeatsern.edu)

OBJECTIVES

- Develop a deep-learning model to predict the direction of stock market price movements
- Extract useful features from diverse sources of information
- Identify correlations between features
- Train a prediction model based on extracted features
- Make predictions using the trained model to guide investment decisions

DATASETS

- Unrestricted stock market data from UCI Machine Learning Repository
- Daily closing prices of S&P 500, NASDAQ, Dow Jones, NYSE, and RUSSELL 2000
- 82 features: technical indicators, economic data, world indices, currency exchange rates, commodities, and futures contracts
- Data split: 60% training, 20% validation, 20% testing

CURRENT STATE-OF-THE-ART

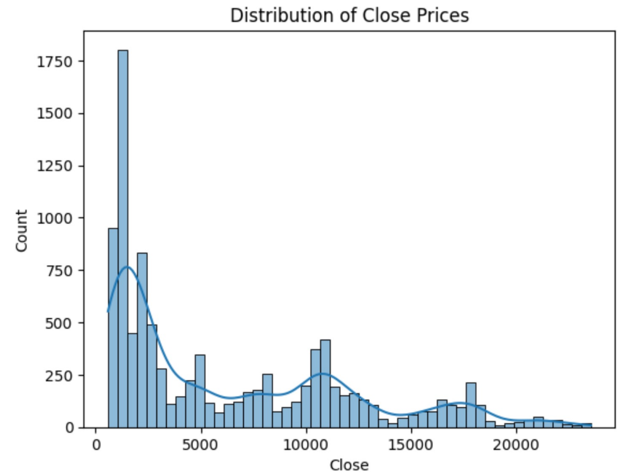
- Advanced neural networks like LSTM for time-series pattern recognition
- Blending various models
- Enhancing data features
- Sentiment analysis using NLP
- Transfer learning for refined predictions
- Model transparency
- Real-time data analysis
- Ethical and regulatory compliance

APPROACH

- Data preprocessing: handle missing values, normalize features, format for CNN
- Exploratory Data Analysis (EDA): trends, correlations, patterns
- Model design and architecture selection: 2D CNN for temporal and cross-sectional dependencies
- Model implementation and training: TensorFlow/PyTorch, hyper-parameter tuning, regularization
- Model evaluation and testing: accuracy, precision, recall, F1-score
- Refinement and optimization: architecture adjustments, ensemble learning
- Potential enhancement: 3D CNN for capturing multi-dimensional interactions

DATA PREPARATION AND PREPROCESSING

- Data acquisition (UCI Machine Learning Repository)
- Preprocessing techniques
  - Data cleaning (handling missing values, removing outliers)
  - Feature engineering (normalization)
  - Data augmentation (jittering, scaling, time shifting)
  - Data transformation (windowing technique)
- Data splitting (80% training, 20% testing)



MODEL ARCHITECTURE AND TRAINING

- Initial CNN model and shift to LSTM
- Architecture details
  - LSTM layers (64 units, 32 units)
  - Dropout layers
  - Dense layer (64 neurons, ReLU activation)
  - Output layer (sigmoid activation)
- Optimization and loss function (Adam, binary cross-entropy)
- Model evaluation (accuracy, validation)

Model: "sequential_41"

Layer (type)	Output Shape	Param #
lstm_26 (LSTM)	(None, 8, 64)	19200
dropout_55 (Dropout)	(None, 8, 64)	0
lstm_27 (LSTM)	(None, 32)	12416
dropout_56 (Dropout)	(None, 32)	0
dense_80 (Dense)	(None, 64)	2112
dropout_57 (Dropout)	(None, 64)	0
dense_81 (Dense)	(None, 1)	65

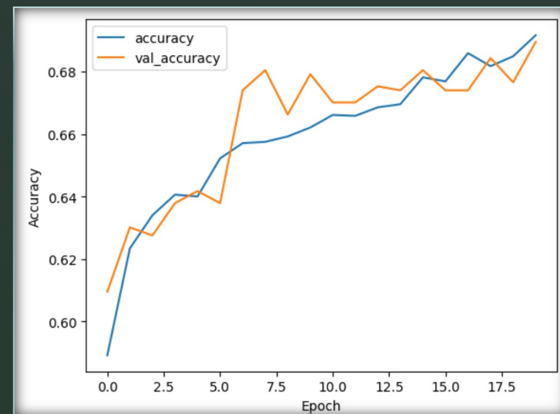
RESULTS

Training performance (accuracy graph)

Test performance

- Accuracy: 67.44%
- Classification report (precision, recall, F1-score)

Comparison with previous CNN-based models



	precision	recall	f1-score	support
0	0.67	0.70	0.69	970
1	0.69	0.65	0.67	971
accuracy			0.68	1941
macro avg	0.68	0.68	0.68	1941
weighted avg	0.68	0.68	0.68	1941

CONCLUSION AND FUTURE WORK

- Promising results with LSTM model
- Potential advantages over CNN-based models
- Limitations and considerations
- Future explorations (additional data sources, feature engineering, model architectures)

ANY QUESTIONS?

THANK-YOU