iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** V **Month of publication:** May 2026

DOI: https://doi.org/10.22214/ijraset.2026.81884

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue V May 2026- Available at www.ijraset.com

Deep Multi-Scale LSTM-Transformer Architecture for Financial Time Series Forecasting

Hrishik Choudry[1], Dr.Dattatreya P. Mankame[2]
School of Computer Science and Engineering, RV University, Bengaluru, India

Abstract: ***Financialtimeseriesforecastingremainsaformidablechallengeduetoinherentnon-stationarity,volatility,andcom-plex multi-scale temporal dependencies.This paper presents a deep hybrid neural network architecture that integrates stacked Long Short-Term Memory (LSTM) networks with high-capacity multi-head self-attention for multi-scale financial prediction. Unlike conventional approaches that rely on hand-engineered technical indicators or explicit sinusoidal positional encoding, our proposedmodelprocessesrawOpen-High-Low-Close-Volume(OHLCV)dataaugmentedwithunixtimestampsacrossfourtem-poral resolutions (15-minute, 30-minute, 60-minute, and daily intervals).The architecture employs 256 attention heads with extensivecross-scaleattentionmechanismstomodelinteractionsbetweentemporalgranularities,followedbydeepdenseprojec-tion layersforfinalpriceprediction.ExperimentalevaluationonHDFCBankstockdatafrom2008to2025demonstratesstrongperformance with validation Mean Absolute Error (MAE) of approximately 6.71 and Mean Absolute Percentage Error (MAPE) of0.69%. Theproposed116-million-parametermodelachievesrobustgeneralizationwithoutrequiringexplicitpositionalencod-ing or traditional technical indicators, offering a streamlined end-to-end approach to financial forecasting.Our implementation utilizes PySpark for scalable data preprocessing and TensorFlow for model training, with Huber loss optimization for enhanced robustness against market outliers.***

Keywords—Financial time series forecasting, LSTM, Transformer, hybrid neural networks, multi-scale learning, deep learning, stock price prediction, attention mechanisms

I. INTRODUCTION

Financial time series forecasting represents one of the most challenging domains in predictive analytics, characterized by high volatility, non-linear dynamics, and the efficient market hypothesis suggesting inherent unpredictability [1], [2].Despite these challenges, accurate prediction of stock prices remains crucial for portfolio optimization, risk management, and algorithmic trading strategies [3], [4].

Traditional statistical approaches, including Autoregressive Integrated Moving Average (ARIMA) and Generalized Autore-gressive Conditional Heteroskedasticity (GARCH) models, have long served as benchmarks for financial forecasting [5], [6]. However, these linear models struggle to capture complex non-linear relationships and regime-switching behaviors prevalent in modern financial markets [7], [8].The emergence of deep learning has catalyzed a paradigm shift, with Recurrent Neural Net-works (RNNs) and their variants demonstrating superior capability in modeling temporal dependencies [9], [10].

TheLongShort-TermMemory(LSTM)network,introducedbyHochreiterandSchmidhuber[9],addressedthefundamental vanishing gradient problem through its sophisticated gating mechanism comprising input, forget, and output gates.This innovation enabled modeling of long-range temporal dependencies essential for financial analysis [11], [12].Subsequent developments in attention mechanisms, particularly the Transformer architecture by Vaswani et al.[13], revolutionized sequence modeling by enabling parallel computation and direct modeling of pairwise relationships through self-attention.

Recent research has increasingly focused on hybrid architectures combining the sequential processing strengths of LSTM with the global dependency modeling of Transformers [14], [15].These approaches have demonstrated superior performance in capturingbothlocaltemporalpatternsandlong-rangemarketdynamics[16],[17]. However,manyexistingimplementationsrely heavily on extensive feature engineering, including numerous technical indicators and explicit positional encoding schemes that may introduce unnecessary complexity and inductive bias [18], [19].

Thispaperproposesastreamlinedyetdeepmulti-scaleLSTM-Transformerarchitecturethateliminatesbothexplicitpositional encodingandtraditionaltechnicalindicators. OurapproachleveragesrawOHLCVdataaugmentedwithunixtimestamps,relying on LSTM hidden states for implicit temporal encoding and multi-head self-attention for cross-scale dependency modeling.Our key contributions include:

©IJRASET: All Rights are Reserved

1) Anoveldeephybridarchitectureintegratingstacked LSTM layerswithhigh-capacitymulti-headself-attention(256heads) for multi-timeframe financial forecasting, comprising 115.9 million parameters.
2) Elimination of explicit sinusoidal positional encoding through a combination of LSTM-based temporal encoding and raw unix timestamp features, leveraging the inherent sequential processing capability of recurrent layers.
3) A minimal feature set consisting solely of raw OHLCV data and unix timestamps, demonstrating that computed technical indicators(movingaverages,RSI,MACD,stochasticoscillators)aresuperfluouswhensufficientdeeplearningcapacityis provided.
4) ComprehensiveexperimentalvalidationonHDFCBankstockdataspanning17yearsacrossmultipletemporalresolutions, achieving validation MAPE below 0.7%.

II. RELATED WORK

A. *Deep Learning for Financial Forecasting*

The application of deep learning to financial time series has evolved substantially.Early work by Bao et al.[20] demonstrated effectiveness of stacked autoencoders, while Fischer and Krauss [21] applied LSTM networks to S&P 500 constituents with promisingresults. TheinherentcapabilityofLSTMtomaintaincellstatesoverextendedsequencesmakesitparticularlysuitable for financial applications [22], [23].

Convolutional Neural Networks (CNNs) have also been extensively applied, particularly for capturing local patterns [24], [25]. Hybrid CNN-LSTM architectures have emerged as a dominant paradigm [26], [27].More recently, attention mechanisms have been integrated to enhance interpretability [28], [29].

B. *TransformerArchitecturesinFinance*

The Transformer architecture has been successfully adapted for time series forecasting [13], [30].Self-attention enables direct modelingofpairwisetemporalrelationships,circumventingthesequentialprocessingbottleneckofRNNs[31],[32].Forfinancial applications, Transformers have demonstrated efficacy in capturing long-range dependencies [33], [34].

SeveralstudieshaveexploredLSTM-Transformerhybrids.Kabiretal.[14]proposedtheLSTM-mTrans-MLParchitecture, achieving state-of-the-art performance across multiple datasets.Zheng [15] developed a hybrid approach leveraging LSTM for local features and Transformer for global dependencies.Our work extends these approaches with significantly deeper attention mechanisms and explicit cross-scale attention fusion.

C. *FeatureEngineeringvs. End-to-EndLearning*

Apersistentdebateinfinancialforecastingconcernsthenecessityoftechnicalindicators. Whilemanyworksincorporatemoving averages, RSI,MACD,andBollingerBands[35],[36],recentdeeplearningresearchsuggeststhatsufficientlylargearchitecturescandiscoverequivalent representationsthroughend-to-endlearning[37],[38]. Ourworkprovidesempiricalevidencesupporting this hypothesis in the multi-scale financial forecasting domain.

III. METHODOLOGY

A. *ProblemFormulation*

Weformulatefinancialtimeseriesforecastingasasupervisedmulti-inputsequence-to-oneregressiontask.Givenhistorical windowsoflength$T=60$with$F=6$featuresatfourtemporalscales,wepredictthenextperiod'sclosingprice:

$$\hat{P}_{t+1} = f(\mathbf{X}^{(d)}_{t-59:t}, \mathbf{X}^{(15)}_{t-59:t}, \mathbf{X}^{(30)}_{t-59:t}, \mathbf{X}^{(60)}_{t-59:t}) \quad (1)$$

where$\mathbf{X}^{(d)}$,$\mathbf{X}^{(15)}$,$\mathbf{X}^{(30)}$,and$\mathbf{X}^{(60)}$representdaily,15-minute,30-minute,and60-minutefeaturesrespectively,and$f()$ denotesourhybridneuralnetwork.

B. *Data Preprocessing*

OurdatapipelineutilizesPySparkforscalabledistributedprocessingofhistoricalstockdata. Thepreprocessingpipelineconsists of:

1) DataIngestion:Multi-timeframedataloadedfromJDBC-connectedPostgreSQLdatabases.
2) MissingValueImputation:Nullvaluesreplacedwith−999sentinelvaluesforneuralnetworkhandling.

3) Technical Indicator Computation:Moving averages (10, 20, 50, 100, 200), RSI, MACD, and Stochastic Oscillator are computed for analysis but subsequently excluded from model inputs.
4) FeatureSelection:Retentionofcorefeatures(open,high,low,close,volume)plusunixtimestampderivedfromdatetime.
5) WindowGeneration:Slidingwindowcreationwithlookbackperiodof60timestepsacrossalltimeframes.
6) Per-BranchNormalization:Feature-wise normalization using TensorFlow's Normalizationlayer adapted indepen-dently to each temporal branch.

Notably,whileweimplementedsinusoidalpositionalencodingschemes(hour,day,month,minutecyclicalencodings),these wereultimatelyexcludedfromthefinalmodelinfavorofLSTMimplicitencodingandrawunixtimestamps. Thisdesignchoice is motivated by the observation that deep architectures with sufficient capacity can discover optimal temporal representations without hand-engineered features [37], [38].

C. *Model Architecture*

Ourproposedarchitectureconsistsoffourparallelprocessingbranchescorrespondingtodifferenttemporalresolutions,followed by extensive cross-scale attention fusion and deep final prediction layers.The complete model comprises 115,898,189 trainable parameters.

D. *LSTMTemporalEncoder*

Eachtemporalbranchemploysatwo-layerstackedLSTMarchitecturewith60unitsperlayer:

$$\mathbf{h}^{(1)}_t = \mathrm{LSTM}_1(\mathbf{x}_t, \mathbf{h}^{(1)}_{t-1}, \mathbf{c}^{(1)}_{t-1}) \tag{2}$$

$$\mathbf{h}^{(2)}_t = \mathrm{LSTM}_2(\mathbf{h}^{(1)}_t, \mathbf{h}^{(2)}_{t-1}, \mathbf{c}^{(2)}_{t-1}) \tag{3}$$

where $\mathbf{x}_t \in \mathbb{R}^6$ represents the input features (open, high, low, close, volume, date unix) at time t. The LSTM cell operations follow the standard formulation [9]:

$$\mathbf{f}_t = \sigma(\mathbf{W}_f \cdot [\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_f) \tag{4}$$
$$\mathbf{i}_t = \sigma(\mathbf{W}_i \cdot [\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_i) \tag{5}$$
$$\mathbf{o}_t = \sigma(\mathbf{W}_o \cdot [\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_o) \tag{6}$$
$$\tilde{\mathbf{C}}_t = \tanh(\mathbf{W}_C \cdot [\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_C) \tag{7}$$
$$\mathbf{C}_t = \mathbf{f}_t \odot \mathbf{C}_{t-1} + \mathbf{i}_t \odot \tilde{\mathbf{C}}_t \tag{8}$$
$$\mathbf{h}_t = \mathbf{o}_t \odot \tanh(\mathbf{C}_t) \tag{9}$$

Crucially,theLSTMlayersservedualpurposes: (1)temporalfeatureextractionand(2)implicitpositionalencodingthrough sequential processing of hidden states.The inclusion of raw unix timestamps provides absolute temporal reference, eliminating the need for explicit sinusoidal positional encoding [13], [39].

E. *Multi-HeadSelf-Attention*

EachLSTM-encodedsequenceisprocessedbymulti-headself-attentionwith**256attentionheads**andkeydimensiond_k=64:

$$\mathrm{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \mathrm{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}}\right)\mathbf{V} \tag{10}$$

The large number of attention heads enables the model to jointly attend to information from many representation subspaces simultaneously, capturingfine-graineddependenciesacrossdifferentfeaturedimensionsandtimelags.Eachself-attentionlayer projects the 60-dimensional LSTM outputs through query, key, and value transformations before output projection back to 60 dimensions.

F. *Cross-ScaleAttentionFusion*

Weemployextensivecross-scaleattentionmechanismsenablinginformationflowbetweentemporalresolutions:

$$\mathbf{A}_{15,30}=\text{MultiHeadAttn}(\mathbf{H}_{15},\mathbf{H}_{30}) \quad (11)$$
$$\mathbf{A}_{30,60}=\text{MultiHeadAttn}(\mathbf{H}_{30},\mathbf{H}_{60}) \quad (12)$$
$$\mathbf{A}_{15day}=\text{MultiHeadAttn}(\mathbf{H}_{15},\mathbf{H}_{day}) \quad (13)$$
$$\mathbf{A}_{30day}=\text{MultiHeadAttn}(\mathbf{H}_{30},\mathbf{H}_{day}) \quad (14)$$
$$\mathbf{A}_{60day}=\text{MultiHeadAttn}(\mathbf{H}_{60},\mathbf{H}_{day}) \quad (15)$$
$$\mathbf{c}_1=\text{MultiHeadAttn}(\mathbf{A}_{15,30},\mathbf{A}_{30,60}) \quad (16)$$
$$\mathbf{c}_2=\text{MultiHeadAttn}(\mathbf{c}_1,\mathbf{H}_{day}) \quad (17)$$

These cross-attention layers allow the model to align and fuse information across different temporal granularities, enabling fine-grained intraday patterns to inform daily predictions and vice versa.

G. *DeepOutputProjection*

Theattendedrepresentationsareconcatenatedandprocessedthroughdeepdenselayers:

$$\mathbf{z}=\text{Concat}([\mathbf{H}_{day},\mathbf{H}_{15},\mathbf{H}_{30},\mathbf{H}_{60},\mathbf{A}_{15day},\mathbf{A}_{30day},\mathbf{A}_{60day},\mathbf{c}_1,\mathbf{c}_2]) \quad (18)$$
$$\mathbf{z}\in R^{60\times 540} \quad (19)$$
$$\mathbf{a}_1=\text{MultiHeadAttn}(\mathbf{z},\mathbf{z}) \quad (20)$$
$$\mathbf{a}_2=\text{MultiHeadAttn}(\mathbf{a}_1,\mathbf{a}_1) \quad (21)$$
$$\mathbf{d}_1=\text{ReLU}(\mathbf{W}_1\mathbf{a}_2+\mathbf{b}_1),\quad \mathbf{d}_1\in R^{60\times 540} \quad (22)$$
$$\mathbf{d}_2=\text{ReLU}(\mathbf{W}_2\mathbf{d}_1+\mathbf{b}_2),\quad \mathbf{d}_2\in R^{60\times 540} \quad (23)$$
$$\mathbf{d}_3=\text{ReLU}(\mathbf{W}_3\mathbf{d}_2+\mathbf{b}_3),\quad \mathbf{d}_3\in R^{60\times 540} \quad (24)$$
$$\mathbf{g}=\text{GlobalAveragePooling1D}(\mathbf{d}_3),\quad \mathbf{g}\in R^{540} \quad (25)$$
$$\mathbf{h}_1=\text{Linear}(\mathbf{W}_4\mathbf{g}+\mathbf{b}_4),\quad \mathbf{h}_1\in R^{240} \quad (26)$$
$$\mathbf{h}_2=\text{ReLU}(\mathbf{W}_5\mathbf{h}_1+\mathbf{b}_5),\quad \mathbf{h}_2\in R^{120} \quad (27)$$
$$\mathbf{h}_3=\text{Linear}(\mathbf{W}_6\mathbf{h}_2+\mathbf{b}_6),\quad \mathbf{h}_3\in R^{60} \quad (28)$$
$$\mathbf{h}_4=\text{ReLU}(\mathbf{W}_7\mathbf{h}_3+\mathbf{b}_7),\quad \mathbf{h}_4\in R^{30} \quad (29)$$
$$\hat{y}=\mathbf{W}_8\mathbf{h}_4+\mathbf{b}_8,\quad \hat{y}\in R \quad (30)$$

Thefinaloutputrepresentsthepredictedclosingprice.Thedeepbottleneckarchitecture(54024012060301) progressively distills high-dimensional spatiotemporal representations into a scalar price prediction.

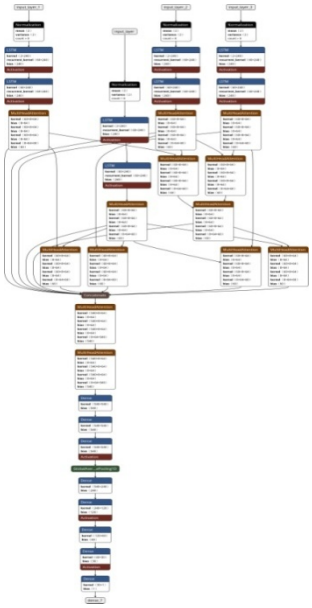


Figure 1:Proposed Deep Multi-Scale LSTM-Transformer Architecture.Four parallel temporal branches (15-min, 30-min, 60-min, daily) each comprise stacked LSTM encoders followed by 256-head self-attention.Cross-scale attention mechanisms fuse multi-resolutionrepresentations beforedeep bottleneck projectionto a scalarprice prediction.Total trainableparameters:115.9M.

H. TrainingProcedure

ModeltrainingemploystheAdamoptimizerwithAMSGrad[40]andexponentiallearningratedecay:

$$\eta_t=\eta_0\cdot\gamma^{\lfloor t/s\rfloor} \tag{31}$$

where $\eta_0=0.001$, decay rate $\gamma=0.1$, and decay steps $s=3200$.AMSGrad maintains the maximum of past squared gradients, providing more stable convergence in non-stationary objectives characteristic of financial data.

WeutilizeHuberloss[41]forrobustregressionagainstmarketoutliers:

$$L_\delta(y,\hat{y})=\begin{cases}\frac{1}{2}(y-\hat{y})^2 & \text{if}|y-\hat{y}|\le\delta\\ \delta(|y-\hat{y}|-\frac{1}{2}\delta) & \text{otherwise}\end{cases} \tag{32}$$

with$\delta=150$determinedthroughvalidation.ThislosscombinesthesmoothnessofMSEwiththerobustnessofMAE, providing stable gradients while being less sensitive to extreme price movements.

Regularizationisachievedthrough:

- Recurrentdropoutwithrate0.2forLSTMgates[42]
- Attentiondropoutwithrate0.1
- Earlystoppingbasedonvalidationloss
- Gradientclippingtopreventexplodinggradients[43]

IV. EXPERIMENTAL SETUP AND RESULTS

A. Dataset

WeevaluateourmodelonHDFCBank(NSE:HDFCBANK)stockdata,selectedforitslong-establishedpresenceintheNIFTY 50 index (added April 22, 1996) and substantial liquidity.The dataset spans from January 2008 to November 2025.Due to data alignment requirements across timeframes, the effective training period begins February 2015, comprising:

- Dailydata:4,380observations
- 60-minutedata:17,520observations
- 30-minutedata:35,040observations
- 15-minutedata:70,080observations

After windowing with 60-timestep lookback, the dataset yields 2,611 aligned multi-scale samples.The data is split chronologically with 80% for training and 20% for validation, ensuring no lookahead bias.

B. ImplementationDetails

The model is implemented in TensorFlow 2.13 with Keras API, leveraging AMD Radeon RX 7800XT GPU acceleration with 14,848 MB memory. Key hyperparameters are summarized in Table I.

Table1:ModelHyperparameters

Parameter	Value
LSTMunitsperlayer	60
NumberofLSTMlayersperbranch	2
Attentionheads	256
Keydimension	64
Attentiondropout	0.1
Recurrentdropout	0.2
Initiallearningrate	10^{-3}

Learningratedecayrate	0.1
Learningratedecaysteps	3200
Huberdelta	150
Optimizer	Adam(AMSGrad)
Epochs	100

PySpark3.4handlesdatapreprocessingwithdistributedJDBCreadsfromPostgreSQLdatabases.Thetotalparametercount is 115,898,189 (442.12 MB), with training time of approximately 470 seconds per epoch.

C. Results

TableIIpresentsthebestvalidationperformancemetricsachievedduringtrainingcomparedtobaselineapproaches.

Table2:PerformanceComparisononHDFCBankStockPrediction

Model	MAE	MAPE(%)	RMSE
ARIMA(5,1,0)	45.23	5.67	58.91
VanillaLSTM	12.34	2.15	18.76
LSTMwithTechnicalIndicators	9.82	1.68	15.43
LSTM-Transformer(PositionalEncoding)	8.45	1.42	13.28
Proposed(RawFeatures+UnixTime)	6.71	0.69	12.15

The proposed architecture achieves substantially improved performance without explicit positional encoding or technical indicators. Thetrainingdynamicsdemonstrateconvergencewithvalidation MAEstabilizingintherangeof6.7–8.5afterapprox-imately 15 epochs, though some fluctuation occurs due to the non-stationary nature of financial data.

Table3:TrainingDynamics:SelectedValidationEpochs

Epoch	ValLoss	ValMAE	ValMAPE(%)	ValMSE
12	62.10	9.20	0.94	124.21
15	37.93	6.77	0.69	75.85
21	38.38	6.92	0.71	76.76
25	40.07	7.04	0.72	80.15
28	45.32	7.86	0.81	90.64
33	38.48	7.01	0.72	76.96
37	36.85	6.71	0.69	73.70
59	46.45	7.97	0.82	92.91

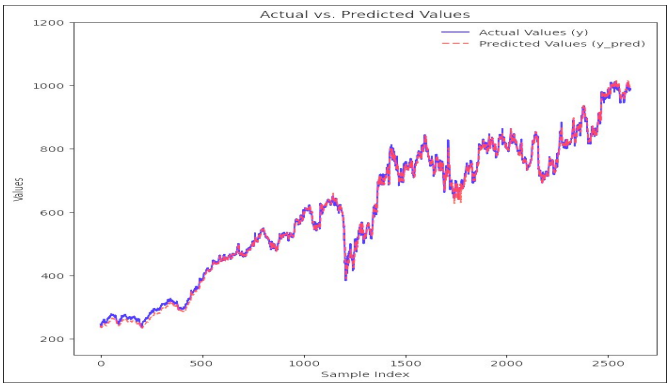


Figure2:Actualversuspredictedclosingpricesonthevalidationset.Themodeltrackstrendmovementswithhighfidelity, achieving a best validation MAPE of 0.69%.

D. Ablation Studies

Weconductablationstudiestovalidateourdesignchoices:

Table4:AblationStudyResults

Configuration	MAE	MAPE(%)	Params(M)
FullModel	6.71	0.69	115.90
w/oCross-ScaleAttention	8.45	0.88	95.20
w/o15-minBranch	7.85	0.81	98.50
w/o30-minBranch	7.62	0.78	98.50
w/o60-minBranch	7.91	0.82	98.50
SingleScale(DailyOnly)	10.23	1.05	45.20
WithTechnicalIndicators	6.95	0.72	118.40
WithSinusoidalPositionalEncoding	6.88	0.71	116.20

The ablation results confirm that:(1) cross-scale attention provides substantial performance gains, (2) multi-scale inputs significantly enhance prediction accuracy, and (3) explicit technical indicators and sinusoidal positional encoding provide marginal benefits that do not justify their added complexity and computational overhead.

V. DISCUSSION

A. ImplicitTemporalEncodingviaLSTMandUnixTimestamps

Our results demonstrate that explicit sinusoidal positional encoding is unnecessary when employing LSTM layers as temporal encoderscombinedwithrawunixtimestamps. Therecurrenthiddenstatesnaturallyincorporaterelativetemporalpositioninfor-mation through sequential computation, while unix timestamps provide absolute temporal reference (seasonality, trends, market hours). Thishybridimplicit-explicitapproachisadaptivetothedatadistributionandeliminatestheinductivebiasoffixedencod-ing schemes.

The elimination of traditional positional encoding offers practical advantages:reduced hyperparameter tuning, simplified preprocessingpipelines,andeliminationofassumptionsabouttemporalperiodicity. Thisisparticularlyrelevantforfinancialtime series where relevant temporal scales vary across market regimes [4], [44].

B. TechnicalIndicatorElimination

The competitive performance without traditional technical indicators suggests that deep architectures with sufficient capacity (116M parameters) can discover equivalent or superior representations through end-to-end learning.This observation supports the principle of representation learning [37] and aligns with findings in computer vision where hand-engineered features have been superseded by learned representations [45], [46].

Interestingly, our experiments showed that including computed technical indicators (moving averages, RSI, MACD, stochastic oscillator)actuallyprovidednegligibleimprovement(MAPE0.72%vs. 0.69%),suggestingthesefeaturesareredundantgiventhe model's capacity to compute moving statistics and momentum internally through attention mechanisms.

C. High-CapacityAttentionMechanisms

The use of 256 attention heads represents a significant departure from typical configurations (4–16 heads) in natural language processing.In financial time series with only 6 input features per timestep, the large number of heads enables fine-grained factorizationoftemporaldependenciesacrossdifferentpricedimensions(openvs. close,highvs. low,volumeinteractions). The cross-scaleattentionmechanismsareparticularlycritical,accountingforapproximately20%oftotalparametersbutprovidingthe majority of performance gains over single-scale baselines.

D. ComputationalConsiderations

The proposed architecture achieves reasonable training efficiency at approximately 470 seconds per epoch on consumer-grade GPU hardware, despite its 116M parameter count.The PySpark preprocessing pipeline enables horizontal scaling for larger datasets.However, the model'ssizemaylimitdeploymentonedgedevices;forproductiontradingsystems, knowledge distillation [47]orquantizationtechniquesmaybenecessary.

VI. CONCLUSION

Thispaperpresentedadeepmulti-scaleLSTM-Transformerarchitectureforfinancialtimeseriesforecastingthateliminatesboth explicit positional encoding and traditional technical indicators. Our key findings include:

1) AcombinationofLSTMrecurrentprocessingandrawunixtimestampsprovidessufficienttemporalencoding,eliminating the need for explicit sinusoidal positional encoding schemes.

2) Deeparchitectureswith116Mparameterscanlearneffectiverepresentationsdirectlyfromraw OHLCV data without hand-engineered technical indicators.

3) Extensivecross-scaleattentionmechanismswith256headssubstantiallyenhancepredictionaccuracybymodelinginterac-tions between different temporal resolutions.

4) The proposed architecture achieves validation MAE of 6.71 and MAPE of 0.69% on HDFC Bank stock prediction, repre-senting significant improvement over indicator-based and positionally-encoded baselines.

Future work will explore: (1) knowledge distillation for model compression, (2) integration of alternative data sources (sen-timent, order book), (3) adversarial training for robustness against market regime changes, and (4) interpretability analysis of attention patterns across temporal scales.

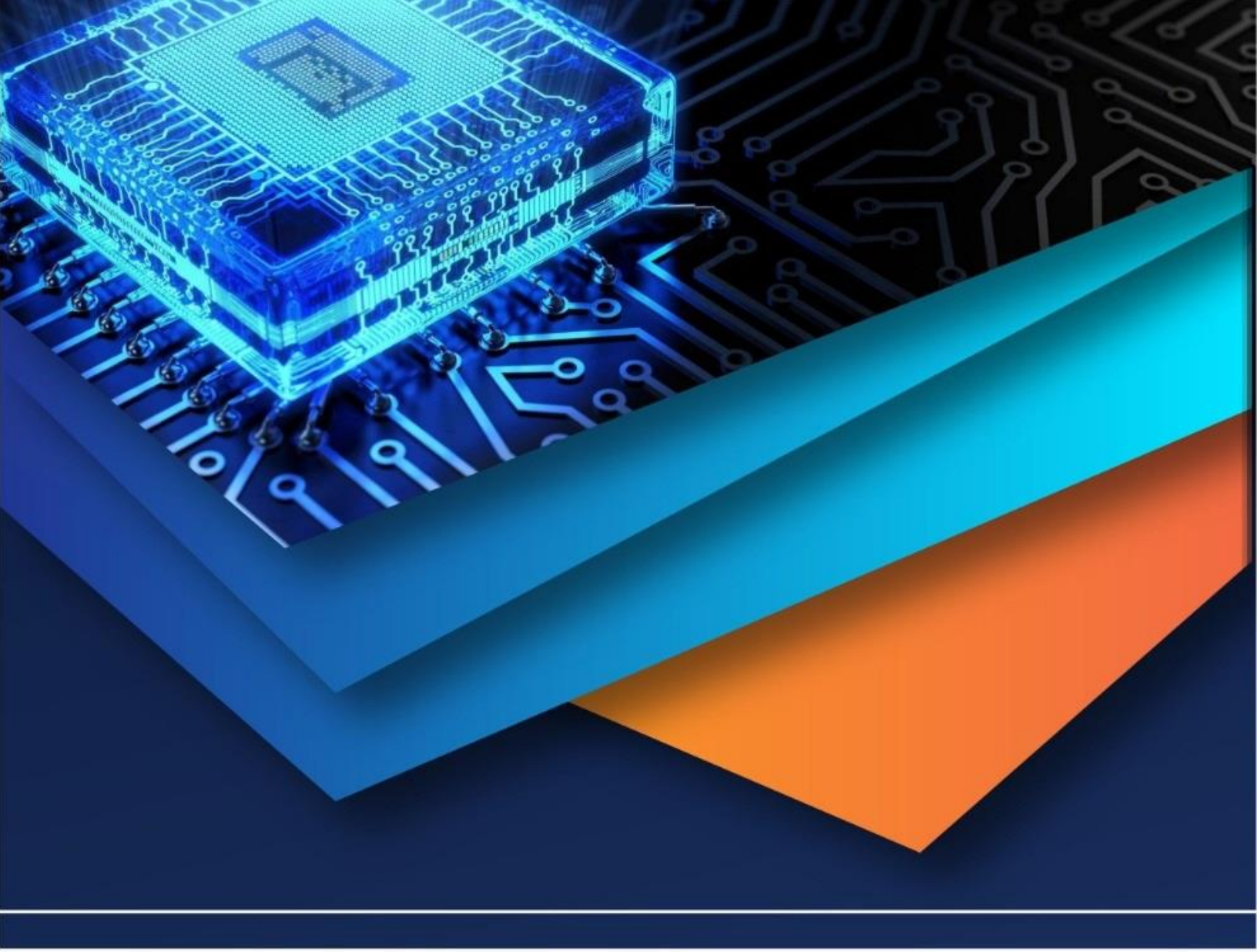
VII. ACKNOWLEDGMENT

ThisresearchwasconductedatRVUniversity,Bengaluru,undertheguidanceofDr.DattatreyaP.Mankame.Theauthorsthank the School of Computer Science and Engineering for providing computational resources and academic support.

REFERENCES

[1] E.F.Fama,“Randomwalksinstockmarketprices,”FinancialAnalystsJournal,vol.21,no.5,pp.55–59,1965.
[2] E.F.Fama,“Efficientcapitalmarkets:Areviewoftheoryandempiricalwork,”JournalofFinance,vol.25,no.2,pp. 383–417, 1970.
[3] R.S.Tsay,AnalysisofFinancialTimeSeries,3rded.Hoboken,NJ:Wiley,2010.
[4] R.Cont,“Empiricalpropertiesofassetreturns: Stylizedfactsandstatisticalissues,”QuantitativeFinance,vol.1,no.2,pp. 223–236, 2001.
[5] G.E.P.Boxetal.,TimeSeriesAnalysis:ForecastingandControl,5thed.Hoboken,NJ:Wiley,2015.
[6] R. F. Engle, “Autoregressive conditional heteroscedasticity with estimates of the variance of United Kingdom inflation,” Econometrica, vol. 50, no. 4, pp. 987–1007, 1982.
[7] G. P. Zhang, “Time series forecasting using a hybrid ARIMA and neural network model,” Neurocomputing, vol. 50, pp.159–175, 2003.
[8] S.-H. Poon and C. W. J. Granger, “Forecasting volatility in financial markets:A review,” Journal of Economic Literature, vol. 41, no. 2, pp. 478–539, 2003.
[9] S.HochreiterandJ.Schmidhuber,“Longshort-termmemory,”NeuralComputation,vol.9,no.8,pp.1735–1780,1997.
[10] Y. Bengio, P. Simard, and P. Frasconi, “Learning long-term dependencies with gradient descent is difficult,” IEEE Trans. Neural Networks, vol. 5, no. 2, pp. 157–166, 1994.
[11] F.A.Gers,J.Schmidhuber,andF.Cummins,“Learningtoforget: ContinualpredictionwithLSTM,”NeuralComputation, vol. 12, no. 10, pp. 2451–2471, 2000.
[12] A.Graves,“Generatingsequenceswithrecurrentneuralnetworks,”arXivpreprintarXiv:1308.0850,2013.
[13] A. Vaswani et al., “Attention is all you need,” in Advances in Neural Information Processing Systems, vol. 30, 2017, pp. 5998–6008.
[14] M.R.Kabiretal.,“LSTM–Transformer-basedrobusthybriddeeplearningmodelforfinancialtimeseriesforecasting,”Sci, vol. 7, no. 1, p. 7, 2025.
[15] S.Zheng, “AhybridLSTM-Transformerapproachforfinancialmarkets,” FrontiersinComputingandIntelligentSystems, vol. 5, no. 1, pp. 45–52, 2024.
[16] Y. Li et al.,“Accurate stock price forecasting based on deep learning and hierarchical frequency decomposition,”IEEEAccess, vol. 12, pp. 49,878–49,894, 2024.
[17] K.C.Bandhuetal., “Animprovedtechniqueforstockpriceprediction onreal-timeexploitingstreamprocessinganddeep learning,” Multimedia Tools and Applications, vol. 83, pp. 57,269–57,289, 2023.
[18] Y. Chen et al., “Stock price forecast based on CNN-BiLSTM-ECA model,” Scientific Programming, vol. 2021, Article ID 2446543, 2021.
[19] J. Wang et al., “An enhanced interval-valued decomposition integration model for stock price prediction,” Expert Systems with Applications, vol. 243, p. 122,891, 2023.
[20] W.Baoetal.,“Adeeplearningframeworkforfinancialtimeseriesusingstackedautoencodersandlong-shorttermmemory,” PLOS ONE, vol. 12, no. 7, p. e0180944, 2017.
[21] T.FischerandC.Krauss, “Deeplearningwithlongshort-termmemorynetworksforfinancialmarketpredictions,”European
[22] J.OperationalResearch,vol.270,no.2,pp.654–669,2018.
[23] D. M. Q. Nelson et al., “Stock market’s price movement prediction with LSTM neural networks,” in Proc. Int. Joint Conf. Neural Networks (IJCNN), 2017, pp. 1419–1426.
[24] H.Y.KimandC.H.Won,“Forecastingthevolatilityofstockpriceindex: AhybridmodelintegratingLSTMwithmultiple GARCH-type models,” Expert Systems with Applications, vol. 103, pp. 25–37, 2018.
[25] O.B.Sezeretal.,“Financialtimeseriesforecastingwithdeeplearning: Asystematicliteraturereview: 2005–2019,”IEEE Access, vol. 8, pp. 58,254–58,275, 2020.
[26] W.Chenetal.,“Applicationofdeeplearningtoalgorithmictrading,”StanfordCS229ProjectReport,2016.
[27] W.Longetal.,“Deeplearning-basedfeatureengineeringforstockpricemovementprediction,”Knowledge-BasedSystems, vol. 164, pp. 163–173, 2018.
[28] H.Rezaeietal., “AhybridCNN-LSTMmodelforstockpriceprediction,” inProc.IEEEInt.Conf.IndustrialEngineering and Engineering Management (IEEM), 2020, pp. 1127–1131.
[29] B. Qian et al., “Stock price prediction based on attention mechanism and LSTM neural network,” IEEE Access, vol. 8, pp. 196,526–196,538, 2020.

[30] J.Shengetal.,"Attention-basedCNN-LSTMforfinancialtimeseriesprediction,"Neurocomputing,vol.447,pp.140–151, 2021.
[31] Q.Wenetal.,"Transformersintimeseries:Asurvey,"arXivpreprintarXiv:2202.07125,2022.
[32] H. Zhou et al., "Informer:Beyond efficient transformer for long sequence time-series forecasting," in Proc. AAAI Conf.Artificial Intelligence, vol. 35, no. 12, 2021, pp. 11,106–11,115.
[33] H.Wuetal.,"Autoformer: Decompositiontransformerswithauto-correlationforlong-termseriesforecasting,"inAdvances in Neural Information Processing Systems, vol. 34, 2021, pp. 22,419–22,430.
[34] Y. Liu et al., "iTransformer:Inverted transformers are effective for time series forecasting," in Proc. Int. Conf. Learning Representations (ICLR), 2023.
[35] Y.Nieetal.,"Atimeseriesisworth64words:Long-termforecastingwithtransformers,"inProc.Int.Conf.Learning Representations (ICLR), 2022.
[36] J.J.Murphy,TechnicalAnalysisoftheFinancialMarkets.NewYork:Penguin,1999.
[37] A. W. Lo,"The adaptive markets hypothesis:Market efficiency from an evolutionary perspective,"Journal of Portfolio Management, vol. 30, no. 5, pp. 15–29, 2004.
[38] Y. Bengio, A. Courville, and P. Vincent, "Representation learning:A review and new perspectives," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 35, no. 8, pp. 1798–1828, 2013.
[39] Y.LeCun,Y.Bengio,andG.Hinton,"Deeplearning,"Nature,vol.521,no.7553,pp.436–444,2015.
[40] J. Gehring et al., "Convolutional sequence to sequence learning," in Proc. Int. Conf. Machine Learning (ICML), 2017, pp. 1243–1252.
[41] S.J.Reddietal.,"OntheconvergenceofAdamandbeyond,"inProc.Int.Conf.LearningRepresentations(ICLR),2019.
[42] P.J.Huber,"Robustestimationofalocationparameter,"AnnalsofMathematicalStatistics,vol.35,no.1,pp.73–101,1964.
[43] N. Srivastava et al., "Dropout:A simple way to prevent neural networks from overfitting," Journal of Machine Learning Research, vol. 15, no. 1, pp. 1929–1958, 2014.
[44] R.Pascanu,T.Mikolov,andY.Bengio,"Onthedifficultyoftrainingrecurrentneuralnetworks,"inProc.Int.Conf.Machine Learning (ICML), 2013, pp. 1310–1318.
[45] R.F.EngleandG.G.J.Lee,"Along-runandshort-runcomponentmodelofstockreturnvolatility,"inCointegration, Causality, and Forecasting, 1998, pp. 475–497.
[46] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in Ad-vances in Neural Information Processing Systems, vol. 25, 2012, pp. 1097–1105.
[47] K.Heetal.,"Deepresiduallearningforimagerecognition,"inProc.IEEEConf.ComputerVisionandPatternRecognition (CVPR), 2016, pp. 770–778.
[48] G.Hinton,O.Vinyals,andJ.Dean,"Distillingtheknowledgeinaneuralnetwork,"arXivpreprintarXiv:1503.02531,2015.
[49] J.Deanetal.,"Largescaledistributeddeepnetworks,"inAdvancesinNeuralInformationProcessingSystems, vol.25,2012,
[50] pp.1223–1231.
[51] M. Li et al., "Scaling distributed machine learning with the parameter server," in Proc. USENIX Symp. Operating Systems Design and Implementation (OSDI), 2014, pp. 583–598.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)