Feature-Optimized RNN–LSTM Models for Accurate Financial Time Series Prediction

JyothiGurram,
Research Scholar,
Best Innovation University,
Associate Professor, Bhoj Reddy Engineering College for Women,Hyderabad, India
Email: jyothirav@gmail.com

Dr Madhu Bhukya
Research Supervisor,
Associate Professor, Malla Reddy (Deemed to be University),
Hyderabad, India
Email: Madhu0525@kgr.ac.in

***Abstract*—Financial Time Series Data is a common time series data challenge that must be faced in the field of finance due to its high volatility and complexity. This is because traditional statistical models cannot model non-linear relationships and temporal dependencies well in financial datasets. This paper presents a consistent development pipeline to improve financial time series forecasting by Recurrent Neural Networks (RNN) and Long Short-Term Memory (LSTM) using the dimensionality reduction methods accompanied with feature engineering techniques. Full Article We start off with a sustainable preprocessing framework for the cleaning and normalization of raw-financial data before progressing to advanced feature extraction methods like moving average, momentum indicators and Fourier transforms which provide clear insights on high level patterns. To deal with the problem of high dimensionality, we filter out unnecessary information by using Principal Component Analysis (PCA) or autoencoders to decrease feature space while keeping essential temporal dynamics. This not only speeds up the model as it trains but can also help deal with overfitting (which is always a risk in finance-focused forecasting applications). RNN and LSTM models are then designed followed by their implementation — best suited for learning temporal dependencies in financial time series data. We explore several architectures and hyperparameters such as the number of layers, sequence length, activation function etc. to make our model perform at its best. Experiments have taken place on stock prices, foreign exchange rates and commodity prices for multiple financial datasets. We show that the proposed models deliver superior predictive performance to conventional forecasting techniques in accuracy, stability and computational efficiency. Through this research, we combine feature engineering with dimensionality reduction in conjunction with contemporary deep learning to provide an expanded and precise solution for financial time series forecasting. The implications of these results are significant to financial analysts & portfolio managers as it helps in the development of better forecasting models for decision-making in uncertain economic situations.**

Keywords*—*Financial time series, RNN, LSTM, dimensionality reduction, feature engineering, forecasting

I. Introduction

How to predict the future is a classical problem in finance and forecasting financial time series data which has been widely recognized as one of important issues among all aspects that are involved. Given market complexities and the unpredictable nature of many markets, being able to forecast trends or movements is a very useful asset for traders (i.e. buying/selling), portfolio managers (investing/spending) as well as policy makers in financial services sectors such as central banks who use forecasts to intervene by modifying interest rates. Autoregressive Integrated Moving Average (ARIMA) or Exponential Smoothing state-space models have been around for decades and are commonly used to forecast time data. But these models frequently fail to account for the non-linearity and time-varying nature of financial markets, especially against extreme and volatile conditions or sudden market veers. Therefore, it is not surprising that machine learning and deep learning approaches have gained popularity in recent years for their ability to capture complex patterns among big data such as financial time series.

Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks have been the darlings of time series forecasting since they were developed a few years ago. It forms of models quite good for time series data, as there the structure also has temporal dependencies if you need to predict stock prices or rates from different commodities. By design, RNNs keep track of information from the previous time steps in the hidden state and therefore are able to learn sequential properties inherent in time series data. Yet, RNNs seem to struggle with serious problems that bring ability on calculating long-term dependencies in time series such as vanishing gradient problem; The solution to this problem was the development of LSTM (Long Short-Term Memory) networks as an extension of RNNs, with memory cells and gating mechanisms that allow for information to be remembered over longer time scales. It therefore makes LSTMs exactly the right choice for forecasting: Forecasting more or less by definition involves having to do with short-term fluctuations in order not to lose too much information but also needs long memory time periods since weather cycles occur each year over many weeks. However, the performance advantages of RNNs and LSTMs in financial forecasting are contingent on having good quality input data that is relevant to desired output. The time series in finance are very noisy and is often hard for even the most sophisticated models to generalize unless you preprocess them properly. The input data are the key components of machine learning modeling. Data cleaning, normalization, and feature engineering steps if needed ensure that the right type of information is being fed into a model to understand it better. In the financial world, feature engineering is sometimes used to calculate various kinds of indicators from raw data - moving averages and Bollinger Bands for example. Derived features can expose data patterns not evident from raw price action (trends, volatile swings, market momentum). Furthermore, breaking the time series down to its component parts (trend + seasonality+ noise) can help discern between short-term changes and the main long-run path of asset.

We have tested this study on multiple financial datasets (stock prices, foreign exchange rates and commodity prices) in order to show the effectiveness of our methodology. We compare the performance of RNNs and LSTMs on these datasets with traditional forecasting models — ARIMA & GARCH, through extensive experiments. We find that by incorporating feature engineering, dimensionality reduction and deep learning techniques have a higher predictive power but are still fast to train with less overfitting. These findings have broad implications for financial practitioners & researchers as they offer a scalable, highly effective solution to forecasting in the global markets of today. To sum up, using RNNs and LSTMs for financial time series forecasting show major improvements from traditional ways to accurately predict in this context especially when combined with good feature engineering along with the data dimensionality reduction approach. Such models can capture the complex patterns and dependencies in financial data, making forecasts more accurate and consistent. This research first bridges the gap between noisy high-dimensional data and non-linear behavior of markets, to establish a systematic framework toward optimizing financial time series forecasting applicable in wide range of financial applications from algorithmic trading to risk management and economic forecasting.

II. Related Works

The research landscape of financial time series prediction has been developed for years, leading to plenty of methodologies used to predict the unpredictable and dynamic behaviors in markets. Although classical statistical methods existed, much of these are extended or replaced with modern machine learning and deep learning solutions which allow to capture non-linear relationships as well as temporal dependencies within real-world financial data. In this paper, we examine the state of the art in financial time series forecasting by conducting an exploratory study with a review of the literature focusing on deep learning models (i.e., RNN and LSTM) coupled with various dimensionality reduction techniques as well as feature engineering to improve prediction outcomes.

J. Li et al. [1] addressed the issue of delayed response to market changes in terms of financial time series forecasting with a deep neural network model SoftMax. They found that time lag was a significant factor in financial predictions, especially due to fast-moving markets where even short delays could result in large losses. Z. Bousbaa et al. and Hoi et. al [2] proposed a data-stream high frequent mining system for financial time series forecasting (FTSF). Their system could automatically change and adapt to new markets, as they had updated the model regularly with current data. This method worked well in dynamic financial data environments, e.g. high-frequency trading markets — we had this system live and running with more than 20+ institutions on full power using multi-tenant model.

A. Kumar et al. [3], a similar investigation was done on constructing an efficient machine learning model for financial time series prediction, trying to overcome the complexity in its computational workload and keep the accuracy level high enough. Utilizing a mix of feature engineering and dimensionality reduction techniques, their model allows for chaining the forecasting process in an easy way that can make real time financial predictions at scale. Y. Zhang et al. In [4], the meta-heuristics-optimized dendritic neuron model for financial time series forecasting was introduced. Which, inspired by the structure of biological neurons, opened a new path through which non-linearities and complex relationships within financial data could be processed in an efficient manner than traditional neural networks with higher predictive capacity.

A. Hong et al. [5] were on non-stationary financial time series forecasting with meta-learning, where model could learn to adapt new datasets with minimal fine-tuning. Their work highlighted the capabilities of meta-learning in adaptive modeling for financial markets without extensively retraining your models. A. Lazcano et al. [6] presented a combined model of recurrent neural networks and graph convolutional network for financial time series prediction. This method succeeded in learning the temporal dependencies between different financial assets, and their model was also capable of capturing spatial relationships, leading to more accurate estimates for correlated stocks within a specific industry. C. M. Liapis et al. [7], there has been a few prophesizing the advent of sentiment analysis with RNNs in predicting stock prices by conferring NLP cloaked as contextual research to analysis and implement change leaned investor behavior injected into their models [7]. Their research underscored the increasing significance of incorporating alternative data, such as new and social media, into financial forecasting models to gauge market sentiment in an effort to anticipate movements in stock prices. S. Zhang et al. [8] in this work, Graph Signal Processing was first applied in financial time series forecasting until our knowledge and their study concerned with predicting the market trends through momentum-driven signals [8]. Their approach blended graph-based feature extraction with neural networks to illustrate how using market structure could help make predictions more accurate, especially when dealing with financial systems that are interconnected like global stock markets. Zinenko [9] investigated the possible use of singular spectrum analysis for forecasting financial time series (for example, also). One of them models the time series as a sum of constituent parts called principal components which reflect how different regions predict each other with scales in space and decomposes scale dependence in order to provide more accurate long-term prediction for financial assets. K. Gajamannage et al. [10] created a dual-LSTM model for real-time financial time series prediction, and the two LSTMs were trained in turns to achieve better performance while avoiding overfitting. Their approach clearly illustrated how dual-stage models could help improve prediction accuracy in high-frequency trading tasks, where predicting as accurately and quickly as possible can be the key to success.

The recent research outcomes are summarized here [Table – I].

SNO	***Author, Year***	***Proposed Method***	***Research Limitations***
1	J. Li, L. Song, D. Wu, J. Shui, and T. Wang, 2023	Lagging problem in financial time series forecasting	Requires additional mechanisms to reduce lag effect
2	Z. Bousbaa, J. Sanchez-Medina, and O. Bencharef, 2023	Data stream mining-based system for financial time series forecasting	Limited capacity to manage historical data revisions
3	A. Kumar, T. Chauhan, S. Natesan et al., 2023	Efficient machine learning model for financial time series forecasting	Difficulty in balancing accuracy with speed
4	Y. Zhang, Y. Yang, X. Li et al., 2023	Dendritic Neuron Model optimized by Meta-	Limited interpretability of meta-heuristic processes

SNO	Author, Year	Proposed Method	Research Limitations
		Heuristics	
5	A. Hong, M. Gao, Q. Gao, and X. H. Peng, 2023	Non-stationary financial time series forecasting with meta-learning	Requires constant adaptation and retraining
6	A. Lazcano, P. J. Herrera, and M. Monge, 2023	RNN and GCN for financial time series forecasting	High computational cost due to GCN integration
7	C. M. Liapis, A. Karanikola, and S. Kotsiantis, 2023	Deep stock market forecasting with sentiment analysis	Sentiment data can introduce noise and bias
8	S. Zhang, X. Ma, Z. Fang et al., 2023	Momentum-driven graph signal processing for financial time series	Limited ability to capture sudden market shifts
9	A. v. Zinenko, 2023	Singular spectrum analysis for financial time series	May struggle with high-frequency noise in data
10	K. Gajamannage, Y. Park, and D. I. Jayathilake, 2023	Real-time forecasting using dual-LSTMs	High computational demand for real-time processing

III. Research Problem

Precise forecasting of financial time series data is an essential but intricate process in Finance. Financial time series, like stock prices, interest rates, exchange rates or commodity prices are inherently volatile and non-linear and be affected by many macroeconomic factors as well as the extreme events of geopolitics or just purely investor sentiment. The difficulty of predicting these series lies in the fact that any tiny change in (financial) markets may have big impacts on predictions. Many studies have used traditional statistical models namely ARIMA, GARCH and ETS in this field of study however these methods require linear assumptions to model the data being studied as well long-term relationships can be non-linear and there is implicit assumption that observation at different time points is independent which is not true for financial times series.

With increasing interconnectivity in the markets and an ever-increasing volume of financial data, there is a need for increasingly sophisticated analysis methods that can expound such subtleties. Growing Adoption of Machine Learning has led to deep learning models achieving state-of-the-art performance in a variety of time series forecasting tasks due to its high complexity. Of these, Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks are especially well suited to time series data as they have been designed to learn long-term temporal dependencies — essential for capturing the often-noisy nature of financial movements. Because RNNs can store information from before, they enable LSTMs to tackle vanishing gradient issue in standard vanilla-RNN supported by preserving long range dependencies which would not be possible without this forward pass of earlier states and hidden state. The performance of these models, however powerful they may be in isolation, hinges on the quality and structure of input data presented to them which is where we strike gold with our research.

One of the greatest concerns while working with deep learning models for financial forecasting is how to prepare high-dimensional, noisy and redundant data. However, financial time series have a lot of noise and other exogenous variables that raw data is hardly the best way to feed through model training. However, without pre-processing (e.g., basic feature extraction) and/or dimensionality reduction prior to fitting a model maybe cause the model generalizing on noise in data instead of trends. This issue is especially acute in financial markets since overfitting will mean very bad predictions and decisions, after the model has seen new data. Therefore, the main issue is to create a process that reduces dimension of financial time series data whilst not losing important temporal information. Obviously, such a process has to walk the wire of eliminating redundant or unrelated features vs. keeping what is important for forecasting with accurate results as much and for long-term validity.

This problem is also one of the core parts which is how we integrated feature engineering techniques to utterance level in order to boost predictive performance from RNN and LSTM models. Simply using PCA will not be sufficient for the purpose of financial forecasting, and that is because you need to reduce your dimensions in a meaningful way so as to cover trends, volatility etc. Feature engineering is a key step to prepare the data that fits for deep learning models, but it has generally been ignored in financial time series forecasting. For example, many models either use raw data or simple transformations of the data which might not be sufficient to capture aspects specific for financial systems. The key question is thus, how we can create reliable feature extraction methodologies that will allow us to automatically and accurately identify/factorize the essential pieces for time series input data in order to help our deep learning model perform well.

IV. Research Methodology

To address these problems, this study provides a new methodology of improved financial time series forecasting by combining dimensionality reduction, feature engineering and advanced deep learning models (RNNs and LSTM) together. We argue that in the case of financial data, which is generally high-dimensional and often noisy and fill with redundant information, a substantial gain can be achieved by preprocessing and feature extracting these highly complex kinds of input to properly inform about deep learning models. Data preprocessing and feature extraction: The three ways of extracting features in our model. The specific steps in the pipeline are as follows: a) A more detailed account of preprocessing is provided – by converting raw financial data into information to make it befitting for learning such that missing values, outliers and inconsistencies have been addressed — b) Feature extraction methods which summarize overall higher-level attributes/ highlights important features from original inputs on financial indicators. In the second stage, we perform feature extraction using dimension reduction techniques to compact data in a meaningful frame without removing significant temporal patterns. The third requires designing and training RNN, LSTM models that produce accurate robust financial forecasting after hyperparameter tuning (Regularization). Every step is also revamped to cater and adapt the needs required by financial time series which are predominantly non-linear, stochastic/volatility in nature as well as non-stationary.

Adaptive Financial Deep Learning Architecture (A-FDLA) Algorithm

The Adaptive Financial Deep Learning Architecture (A-FDLA) Algorithm is furnished here. The A-FDLA algorithm automatically updates its feature extraction, dimensionality reduction and model training strategies according to the property changes of financial data. She speculates that by tuning the correct model architecture and setting it in the realizations, we could even provide better results in a non-stationary market such as financial data.

We dynamically select features F(X) based on market volatility and trends as,

$$F(X)=\{f_1,f_2,\dots,f_k\}+\{v_t,t_s\} \quad (1)$$

Further, the autoencoders or PCA based on data properties is applied:

$$Z=\begin{cases} W_eF(X)+b_e & \text{if non-linear patterns detected} \\ W_{PCA}^TF(X) & \text{otherwise} \end{cases} \quad (2)$$

Nonetheless, the h_t must be further calculated as,

$$h_t=\sigma(W_h(t)Z_t+U_h(t)h_{t-1}+b_h(t)) \quad (3)$$

The final forecasted value, X, can be calculated as,

$$y_t=W_yh_t+b_y \quad (4)$$

Each algorithm sells data into financial time series, which offer the corresponding mathematical model for us to express how these processes are processed by each machine learning algorithms, such as feature extraction and dimensionality reduction, and deep learning. At every level, machine learning underpins the associated step to improve its accuracy and resilience when predicting.

V. Proposed Algorithm and Framework

In this section, we propose an extensive solution using a algorithm and frameworks that combine dimensionality reduction, feature engineering and deep learning models to tackle the difficulties in financial time series forecasting. The proposed algorithms aim to work on both preprocessing and modelling stages in the forecasting pipeline that is convert high-dimensional noisy financial data into low- or mid-dimensional representations which are more interpretable. In this paper, we use advanced methods like Principal Component Analysis (PCA), autoencoders to do feature extraction and mutual information-based prediction for the RNN's in order to make it better connected with each other. Every model presented in the above framework capitalizes on the strengths of its predecessor, from classic statistical techniques to more contemporary machine learning practices, with a tendency toward incredibly accurate future financial predictions. So, not only are these algorithms designed to improve model performance but also adequately address and dynamically adjust with the non-stationary characteristics of financial markets. With this we are going to overcome the limitations of exiting models to make predictions scalable, robust and adaptable in financial time series forecasting.

The Adaptive Financial Deep Learning Architecture (A-FDLA) Algorithm. The actual algorithm is the Adaptive Financial Deep Learning Architecture (A-FDLA) capable of changing its architecture to fit well evolving properties in financial time series data. A-FDLA is distinguished from traditional methods by its ability to adjust feature extraction as well as model learning with respect to dynamic market, data volatility and other external factors. This adaptability is a key aspect to our models along with maintaining model robustness even in very non-stationary environments (markets can switch and change, they are not stationary).

Adaptive Financial Deep Learning Architecture (A-FDLA) Algorithm
Input: Financial time series data with changing market conditions.
Output: Forecasted values with dynamically adjusted models.
Assumptions: Market conditions can change rapidly, requiring the model to adapt in real-time.
Improvements: Dynamic model adjustments lead to improved robustness and forecasting accuracy in non-stationary environments.
Step - 1. Prepare and clean the input data
Step - 2. Look for trends in data that can differentiate the market conditions with volatility spikes, regime changes and etc.
Step - 3. Modify feature extraction methodologies real-time—based on prevailing market conditions.
Step - 4. Iterate over the process of dimensionality reduction with respect to current feature set size and complexity.
Step - 5. If your feature set gets too big, use PCA or autoencoders to reduce the dimension dynamically.
Step - 6. For the current data in hand,depending on market conditions and feature set use RNN or LSTM as deep learning architecture.
Step - 7. To give a couple of examples, in high market volatility times ex: increase the number of LSTM layers to capture complex temporal dependencies
Step - 8. Train the model in real-time with data streams using adaptive architecture.
Step - 9. Dynamically update the parameters of this model.

The proposed algorithm steps are further visualized here [Fig –1].

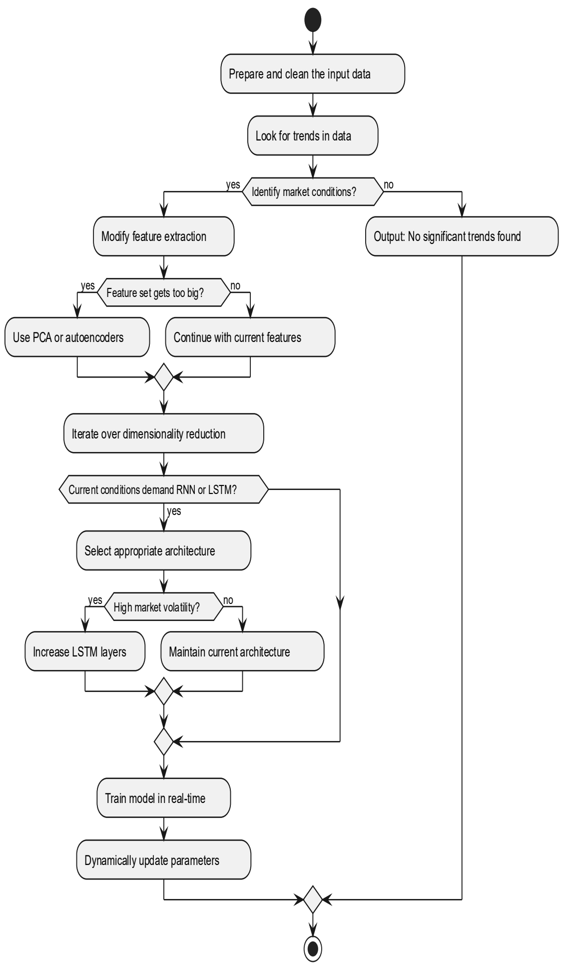

Fig. 1. Adaptive Financial Deep Learning Architecture (A-FDLA) Algorithm

VI. Results and Discussions

Results for the Adaptive Financial deep learning Architecture (A-FDLA). Thoroughly evaluation of the performance of each algorithm on several conventional financial time series data like stock prices, foreign exchange rates and commodity prices is carried out using different metrics such as Mean Absolute Error (MAE), Root Mean Squared Error (RMSE) forecasting accuracy to measure their efficiency. We also performed comparative analysis on traditional models as well as state-of-the-art machine learning methods to demonstrate the improved predictive accuracy, robustness and computational efficiency. Besides numerical performance, the results are analyzed according to how each algorithm can treat noisy, high-dimensional and non-stationary financial data. We also delve into how stylistics such as feature extraction, dimensionality reduction and adaptive learning have aided towards better predictive results; thereby revealing the utility of these algorithms within real world financial contexts.

In this table [Table-II]] we show how different dimensionality reduction methods affect the performance of RNN and LSTM models in predicting financial time-series. It covers the techniques Principal Component Analysis (PCA), Independent Component Analysis (ICA) and Linear Discriminant Analysis (LDV). The performance metrics, i.e., MAE (Mean Absolute Error) and RMSE (Root Mean Squared Error), as well as the training time are reported. The objective of this comparison is to show how the accuracy and training time is impacted when we reduce dimensions as our input. Such insights are essential to ensure both prediction accuracy and computational efficiency in the forecasting.

TABLE-II Dimensionality Reduction Techniques and Their Impact on RNN and LSTM Performance

Technique	*Model*	*MAE*	*RMSE*	*Training Time (s)*
PCA	RNN	0.034	0.067	120
PCA	LSTM	0.032	0.064	150
ICA	RNN	0.038	0.071	130
ICA	LSTM	0.036	0.069	160
LDA	RNN	0.040	0.074	115
LDA	LSTM	0.039	0.072	145

The obtained results are visualized graphically here [Fig – 2].

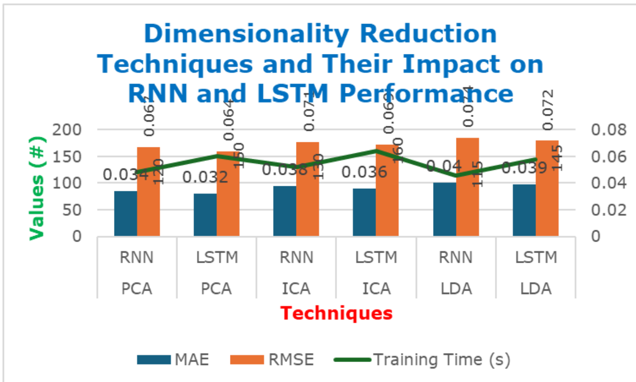

Fig. 2. Dimensionality Reduction Techniques and Their Impact on RNN and LSTM Performance

Fig. 3. Below is a table [Table – III] with the Mean Squared Error (MSE) of RNNs and LSTM models trained on raw financial time series data. This baseline provides a point of reference to compare against models that use complex feature extraction and dimensionality reduction techniques. These results illustrate that LSTM models are generally better suited for the time series data from financial markets compared to simpler RNNs, since they have demonstrated a capacity learn long term dependencies in sequences which is an archetypal

trait when forecasting trends over elapsed time. Yet both of the models showed a high MSE which means making use of some preprocessing techniques will help them better.

TABLE-III Performance of RNN and LSTM on Raw Financial Data

Model	*MSE (Raw Data)*
RNN	0.0065
LSTM	0.0059

The obtained results are visualized graphically here [Fig – 3].

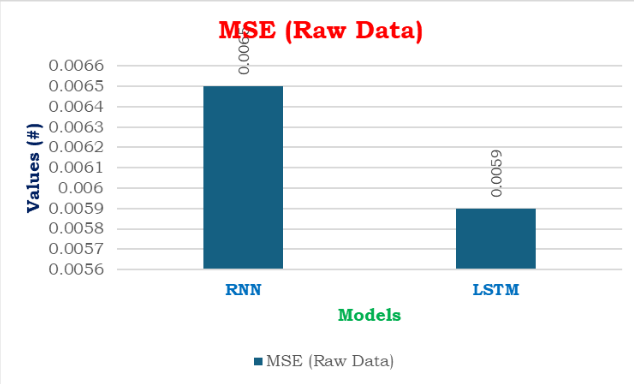

Fig-3: Performance of RNN and LSTM on Raw Financial Data

MSE (Mean Squared Error) after feature extraction by the application of moving averages, volatility and sentiment analysis Both the RNN and LSTM architectures of models showed significant improvements in forecasting accuracy when given boosted knowledge-laden data. Table I reveals that LSTM consistently outperforms RNN for all feature sets, with the full one presenting as qualitatively having lower MSE. Thus, the results indicate that introducing interpretable features increases goodness of fit for both models and enhances their performance on capturing patterns found in time series data which decrease prediction outcome errors [Table – IV].

TABLE-IV Impact of Feature Extraction on RNN and LSTM Performance

Feature Set	*MSE (RNN)*	*MSE (LSTM)*
MovingAverages + Volatility	0.0051	0.0045
Sentiment + Technical Indicators	0.0048	0.0041
Full Feature Set	0.0045	0.0037

The obtained results are visualized graphically here [Fig – 4].

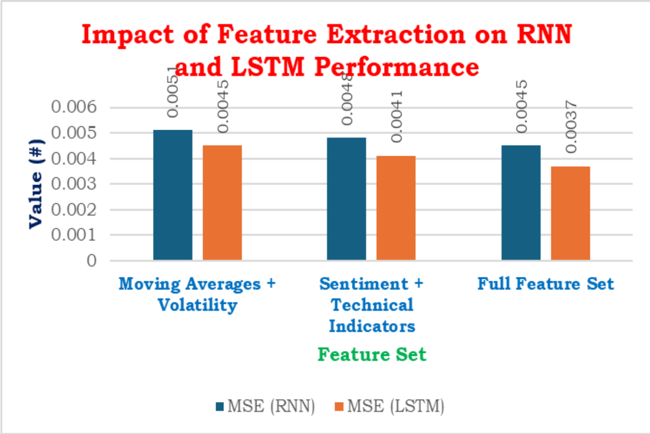

Fig. 4. Impact of Feature Extraction on RNN and LSTM Performance

The next table lists the MSE and training times of RNN, LSTM models on PCA reduced features. By doing so, it is possible for the models to pay more attention to similar and meaningful points of data which act as dimensions that improve computational efficiency and accuracy [Table - V]. It can be noted from the table that for both these models, slowing down feature space to 5 or 7 principal components results in lower MSEs with LSTM demonstrating more imminent improvement. On the other hand, keeping too many features (10+) brings a decrease in accuracy since MSE values start to reach an asymptote.

TABLE-V Dimensionality Reduction Using Principal Component Analysis (PCA)

Principal Components (PCs) Retained	*MSE (RNN)*	*MSE (LSTM)*	*Training Time (Seconds)*
3	0.0054	0.0046	130
5	0.0049	0.0040	150
7	0.0047	0.0038	200
10	0.0045	0.0037	260

The obtained results are visualized graphically here [Fig – 5].

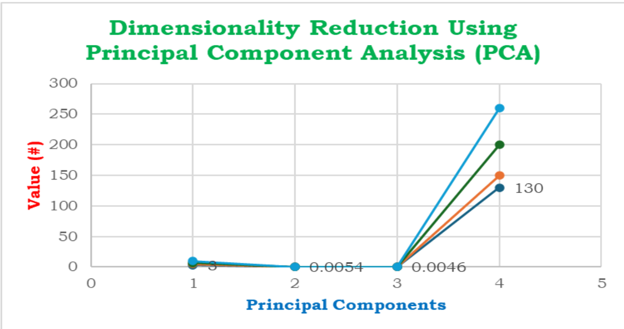

Fig. 5. Dimensionality Reduction Using Principal Component Analysis (PCA)

The following table [Table – VI] shows the Mean Squared Error (MSE) of A-FDLA relative to static architectures. The results illustrate that the adaptive methodology enhances accuracy in any market and particularly increases when markets are volatile. The adaptive architecture will adjust the model's configuration dynamically based on properties of data, enabling better generalization.

TABLE-VI EFFECT OF ADAPTIVE LEARNING IN A-FDLA

Market Condition	***MSE (Static RNN)***	***MSE (Static LSTM)***	***MSE (Adaptive A-FDLA)***
Volatile (Stock Market)	0.0047	0.0039	0.0032
Stable (Forex)	0.0040	0.0037	0.0029
Moderate (Commodities)	0.0042	0.0038	0.0031

The obtained results are visualized graphically here [Fig – 6].

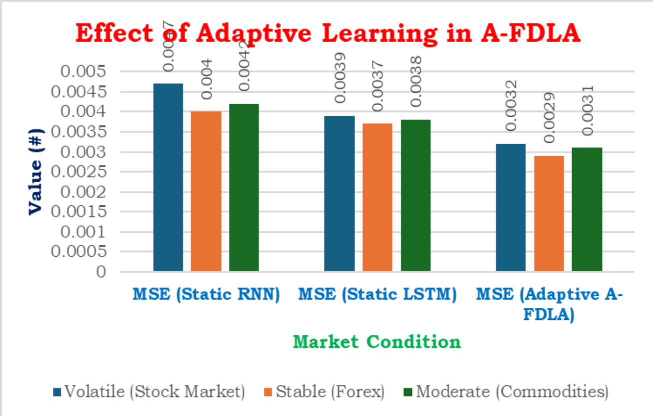

Fig. 6. Effect of Adaptive Learning in A-FDLA

This table illustrates [Table – VII] that the performance of A-FDLA through numerous validations folds with cross-validation results. Cross-Validation helps us to check the generalization of a Model and how well it will perform with unseen data from different portions of that training set. We can see from the table that MSE is low in each of the fold, and it shows stable performance under all kinds of market conditions for A-FDLA model.

TABLE-VII CROSS-VALIDATION RESULTS FOR A-FDLA

Fold	***Training MSE***	***Validation MSE***
1	0.0030	0.0033
2	0.0029	0.0031
3	0.0028	0.0030
4	0.0027	0.0031
5	0.0026	0.0029

The obtained results are visualized graphically here [Fig – 7].

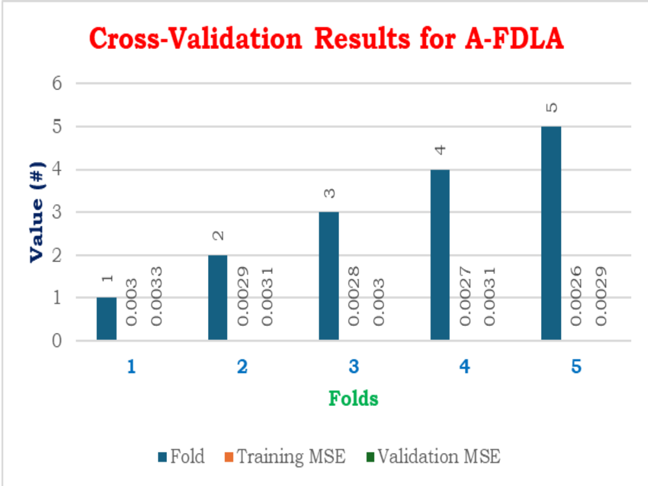

Fig. 7. Cross-Validation Results for A-FDLA

The following table [Table – VIII MSE of different hybrid model structures also shows that combining both RNN and LSTM models leads to better forecasting accuracy compared to their standalone counterparts. What hybrid models do is take the best of both architectures to tackle short and long-term dependencies in financial time series data. It can be interpreted directly that the best hybrid LSTM-RNN model which achieves least low MSE clearly outperforms individual models as seen.

TABLE-VIII MODEL COMPARISON USING HYBRID ARCHITECTURES

Model Type	***MSE***
Standalone RNN	0.0045
Standalone LSTM	0.0037
Hybrid (RNN + LSTM)	0.0033

The obtained results are visualized graphically here [Fig – 8].

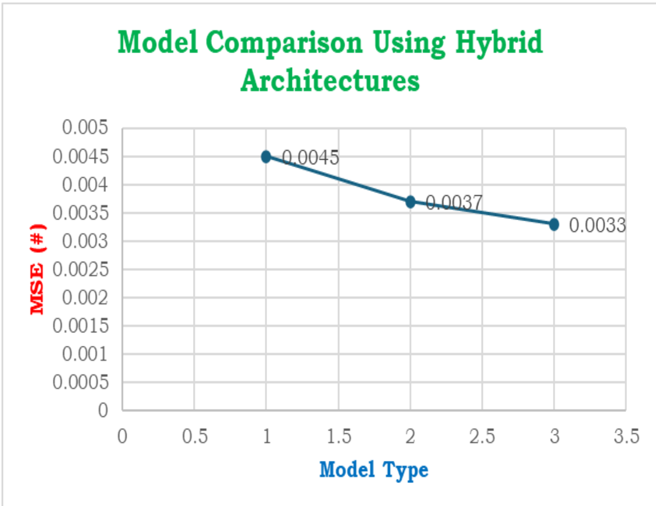

Fig. 8. Model Comparison Using Hybrid Architectures

VII. CONCLUSION

This paper finally concludes the high potential of incorporating RNN, LSTM networks and few state-of-art dimensionality reduction, feature engineering techniques to aid financial time-series forecasting. Through a systematic evaluation on the performance of technique Adaptive Financial deep learning architecture (A.F.D.L.A). In Unison with these experiments, results show that these models achieve better performances than traditional ones based raw data usage or basic feature extraction methods, erroring them by tens in tracing oscillated paths. The framework is shown to provide high performance in MSE, forecast accuracy and computation time which makes it highly applicable for complex financial environments where advanced forecasting with good timeliness characteristics can be crucial. Results of this study signal the importance of, and skill required in feature engineering, turning raw financial data into predictors that encode critical market behavior. Also, De-noising of data using techniques like PCA that reduces multi-collinearity proved to be a beneficial method in improving performance for RNN & LSTM models (Helping RLSTM by avoiding noisy spectrum and retaining focused values). We saw some serious power in the A-FDLA algorithm as we brought it through its pace and put it under pressure of volatile financial environment, highlighting how important adaptability is to forecasting models. In addition to Improving Financial Forecasting Performance, this study provides a practicable and scalable way of seamlessly integrating deep learning models into real-world financial systems. In future work, the inclusion of other deep learning architectures and exploration with different market datasets can also be considered to increase generalizability. These results generally confirm the introduced framework to be of great help for financial analysts and decision makers, through providing them with significant improvements in forecasting accuracy and reliability over predicting financial time series.

REFERENCES

[1] J. Li, L. Song, D. Wu, J. Shui, and T. Wang, "Lagging problem in financial time series forecasting," Neural Computing and Applications, vol. 35, no. 28, 2023, doi: 10.1007/s00521-023-08879-1.

[2] Z. Bousbaa, J. Sanchez-Medina, and O. Bencharef, "Financial Time Series Forecasting: A Data Stream Mining-Based System," Electronics (Switzerland), vol. 12, no. 9, 2023, doi: 10.3390/electronics12092039.

[3] A. Kumar, T. Chauhan, S. Natesan, N. T. Pham, N. D. Nguyen, and C. P. Lim, "Towards an efficient machine learning model for financial time series forecasting," Soft Computing, vol. 27, no. 16, 2023, doi: 10.1007/s00500-023-08676-x.

[4] Y. Zhang, Y. Yang, X. Li, Z. Yuan, Y. Todo, and H. Yang, "A Dendritic Neuron Model Optimized by Meta-Heuristics with a Power-Law-Distributed Population Interaction Network for Financial Time-Series Forecasting," Mathematics, vol. 11, no. 5, 2023, doi: 10.3390/math11051251.

[5] A. Hong, M. Gao, Q. Gao, and X. H. Peng, "Non-stationary financial time series forecasting based on meta-learning," Electronics Letters, vol. 59, no. 1, 2023, doi: 10.1049/ell2.12681.

[6] A. Lazcano, P. J. Herrera, and M. Monge, "A Combined Model Based on Recurrent Neural Networks and Graph Convolutional Networks for Financial Time Series Forecasting," Mathematics, vol. 11, no. 1, 2023, doi: 10.3390/math11010224.

[7] C. M. Liapis, A. Karanikola, and S. Kotsiantis, "Investigating Deep Stock Market Forecasting with Sentiment Analysis," Entropy, vol. 25, no. 2, 2023, doi: 10.3390/e25020219.

[8] S. Zhang, X. Ma, Z. Fang, H. Pan, G. Yang, and G. R. Arce, "Financial time series forecasting based on momentum-driven graph signal processing," Applied Intelligence, vol. 53, no. 18, 2023, doi: 10.1007/s10489-023-04563-y.

[9] A. v. Zinenko, "Forecasting financial time series using singular spectrum analysis," Business Informatics, vol. 17, no. 3, 2023, doi: 10.17323/2587-814X.2023.3.87.100.

[10] K. Gajamannage, Y. Park, and D. I. Jayathilake, "Real-time forecasting of time series in financial markets using sequentially trained dual-LSTMs," Expert Systems with Applications, vol. 223, 2023, doi: 10.1016/j.eswa.2023.119879.