

Dynamic Temporal Selectivity in Non-linear Time Series Forecasting: An End-to-End LSTM-Multiplicative Attention Architecture

B. Dilip Kumar Reddy^{1,2}, Rajavarman V.N¹

¹Department of Computer Science and Engineering, Dr M.G.R. Educational and Research Institute (Deemed to be university), Chennai-600095, Tamil Nadu, India. rajavarman.vn@drmgrdu.ac.in

²Department of Computer Science and Engineering, G. Pulla Reddy Engineering College (Autonomous), Kurnool, Andhra Pradesh, India. dilipkumarreddy.cse@gprec.ac.in

Abstract: – Complex non-linear time series data make it one of the most urgent problems in any type of industry, where models that can capture long-range, changing correlations are needed when conventional statistical techniques, such as ARIMA, are insufficient. Therefore, in order to overcome these constraints and attain a higher temporal selectivity and predictive capability, we present and justify the improved, integrated Long Short-Term Memory (LSTM) network enhanced with a Multiplicative Attention mechanism in place of the original. The scientific contribution of the work itself is two-fold: First, we show that this one, and the same deep learning architecture, offers better efficiency and accuracy rates than the multi-component hybrid models that are hard to use. Second, the Multiplicative Attention weighting is created as one of the most important scientific contributions to model interpretability, which gives us dynamically explainable information on which historical time-steps are the most important to each future prediction. This is addressed by dismantling traditional LSTMs' "black-box" characteristics. The model is strong and most effective as validated through experimental testing on a real-life benchmark market, S&P 500 volatility statistics. The LSTM-Multiplicative Attention model proposed provided the low error metrics of a Root Mean Squared Error (RMSE) of 5.75, Mean Absolute Error (MAE) of 4.62 and a Symmetric Mean Absolute Percentage Error (SMAPE) of 8.93. This result is a substantial improvement over the conventional ARIMA model (RMSE of 12.45%) and even the conventional LSTM (RMSE of 7.30%), making the model a highly sophisticated, flexible, and interpretable forecasting tool for nonlinear time series.

Keywords: – Autoregressive Integrated Moving Average, Machine Learning, Long Short-Term Memory, Recurrent Neural Network, Multiplicative Attention Mechanism

1. INTRODUCTION

Forecasting is a vital ingredient to be used in many sectors, though in such categories as finance, retail, energy, or healthcare, the skill of predicting future trends and behaviour matters greatly in decision-making and approach planning. Conventionally, the ARIMA model has been part of the cornerstones of forecasting due to its simplicity and usefulness in managing stationary, linear dependencies in time series data. The limitations of ARIMA are, however, obvious when it comes to such non-linear data, missing values and long-term forecasting of sequences, hence the need to consider more sophisticated methodologies [1].

The recent breakthroughs in deep learning, especially the creation of the LSTM network, have shown great potential in overcoming the limitations. Being a form of RNN, LSTMs have the ability to process sequences and successfully acknowledge long-term connections [2]. Irrespective of their merits, conventional LSTMs have a severe limitation in that the processing of very long sequences: they reduce the entire input history to a fixed-size context vector (the final hidden state). This causes a bottleneck, making the model to water down the significance of past occurrences that are crucial and end up deconstructing it to maintain attention in longer time periods [3]. Besides regular LSTMs do not consider the fact that time steps can be treated differently because the model does not contain the mechanism to prioritize and consider critical and non-critical observations.



To address this bottleneck and improve both the performance and interpretability of LSTM models, the addition of the Multiplicative Attention mechanism has been extensively suggested [4]. The Multiplicative Attention mechanism enables the model to dynamically give each time interval in the input sequence special weighting of importance. This dynamic selectivity is a critical benefit: it gives the model the capability to pay attention mainly to data in the past that is most relevant to the example, a sudden financial crash or a critical environmental condition, hence enhancing the feature of complex data distribution and long time series [5]. Causing the focus of the model to be explicit means that, also through the Multiplicative Attention mechanism, the model is much easier to interpret, and strong trust and utility are established in the serious forecasting areas. In this study, we introduce an integrated deep learning architecture, the integration of an LSTM network and a Multiplicative Attention mechanism. It is novel since it shows how this single and lean architecture can be more efficient at computing and forecasting better than those based on multi-component hybrids (e.g., ARIMA-LSTM) and, at the same time, offer more interpretability, which is a fundamental attribute of current time series analysis.

2. RELATED WORK

Time series forecasting has experienced quite an evolutionary change in the sense that it used to be based on traditional statistical models, but today it has been transformed to use advanced deep learning systems that are capable of simulating non-linear relations, among others. This chapter is a literature review, which is divided by methodology, and it brings out the developments, advantages and limitations embedded within each methodology

2.1. *Traditional and Statistical Methods*

Time series analysis has traditionally been based on traditional models (ARIMA and SARIMA) due to their simplicity and effectiveness. ARIMA is successful in predicting stationary processes, and it identifies linear relationships thus, it is well adapted to predictable low-volatility data [6]. SARIMA expands this ability by adding seasonal elements, which would be beneficial in applications such as energy demand prediction [7]. But the inherent flaw of these approaches is that they are based on the assumption of linearity. This limitation does not allow them to model complex, non-linear dynamics, volatility clustering, and other complicated relationships of time, which are common to modern, high-frequency data [1]. As a result, their performance seriously deteriorates in situations where they are used with chaotic or non-stationary data, where they are typically poor at sequence forecasting in the long term, and when there are missing values.

2.2. *Deep Learning for Time Series*

More research has concentrated on deep learning, especially RNN, in order to overcome the weaknesses of Linear models. LSTM network has been proposed specifically with the aim of alleviating the vanishing and exploding gradient that affect traditional RNNs to ensure that long-term temporal dependencies can be more effectively trained by the network [8]. LSTMs have proven to be significantly better than ARIMA in non-linear applications, and they have been demonstrated to be successful in fields such as hydrology [9] to general sequence learning. However, there is an inherent drawback in the use of standard LSTMs, namely the bottleneck between the input history and the size of the context (the final hidden state). It should be noted that some design choices to resolve this issue exist, such as using a larger context size to retain more information about the past. The result of this structural bottleneck is memory dilution on very long sequences, and it does not have the granularity of dynamically distinguishing and prioritizing strong events in the input, and they are uniformly distributed along with all past observations [3].

2.3. *Ensemble and Hybrid Methods*

In order to enjoy the strength of statistical modelling and forecasting capabilities of deep learners, scientists have suggested Hybrid Models and Ensemble Learning. This ARIMA-LSTM hybrid scheme, as an example, tries to separate the time series data into the linear (modelled by ARIMA) and non-linear (modelled by LSTM) parts, where it usually has a high accuracy as compared to the single models [10]. Some other ensemble techniques, such as the Guided Method of Time Series Ensemble Forecasting (GATE), consider a combination of several deep learning models (e.g., RNN, LSTM, Conv-LSTM) to enhance stability and robustness with diverse datasets [11]. Although these methods enhance the precision of prediction [12], they are very costly in terms of complexity and computation. They require hand-tuning, component-based tuning, and do not have the architectural beauty and performance of a single, end-to-end deep learning architecture.

2.4. *Multiplicative Attention-Based Models and Position of Work*

Such a structural update as the Multiplicative Attention mechanism integrated into sequence models can be seen as the needed structural improvement to overcome the flaws of normal LSTMs. Multiplicative Attention explicitly adds the layer of dynamic time selectivity to the model, in which the model can give unique importance weights to every time step, and this way, the model can pay attention more specifically, and the fixed context bottleneck can be resolved [4]. Table 1 can be used to synthesize the drawbacks of the current methodologies and is an effective way of placing the contribution of the proposed LSTM Multiplicative Attention architecture. This research clearly positions the LSTM with the Multiplicative Attention mechanism as the necessary solution to overcome these collective drawbacks.

2.4.1. **Superior Selectivity**

Covers the Multiplicative Attention mechanism in contrast to typical LSTMs, the Multiplicative Attention mechanism offers a very important dynamic selectivity layer. It enables the model to place special weight of significance on all past observations, as the forecast is based heavily on the most factor or event of interest (e.g., a critical price jump of a financial series) instead of giving equal weight to all the other past data.

2.4.2. **Increased Interpretability**

This adaptive weighting causes the model to be much easier to interpret by visually projecting the model to understand what time steps the model is paying attention to and developing confidence and utility in key areas of forecasting

2.4.3. **Performance Differentiation**

The higher level of performance, with the lowest error rates (i.e., RMSE of 5.75) of the model than both ARIMA and the normal interpretation of LSTM, empirically confirms that such a structural enhancement is more helpful to model more complex time series, which are non-linear.

3. **VARIOUS METHODS IN TIME SERIES FORECASTING**

3.1. *Convolutional Statistical Models*

ARIMA (Autoregressive Integrated Moving Average): A well-liked and dependable technique for time series analysis is ARIMA method [6]. A time series dataset's underlying patterns and dependencies are fundamentally captured by an ARIMA model, including components for moving averages (MA), differencing (I), and autoregression (AR).

3.1.1. **Leveraging the ARIMA**

The foundation of our forecasting approach is the ARIMA model. By taking into account both autoregressive and moving average components, as well as accounting for seasonality, this potent statistical instrument allows us to record time series data complex dynamics.

3.1.2. **SARIMA (Seasonal ARIMA)**

The SARIMA technique is a simple expansion of ARIMA and non-seasonal autoregressive-moving average (ARMA) previously introduced model. ARIMA (p, d, q) (P, D, Q) m is the notation for SARIMA models, where (p, d, q) and (P, D, Q) m stand for the model's seasonal and non-seasonal components, respectively. The number of seasons is represented by the parameter m [13].

3.2. *Deep Learning Models*

LSTM (Long Short-Term Memory): Neural Networks that repeat RNNs are a common neural network in deep learning. However, it turns out that the gradient vanishing problem (gradient exploding can also happen) makes it exceedingly hard to train classical RNNs to manage long-term reliance. The vanishing gradient problem can be solved with LSTM [18]. The most fundamental part of an LSTM architecture is the LSTM unit, which is made up of a number of gates and cells that cooperate to generate the final output. Gated Recurrent Units (GRU): GRUs also perform well in sequence learning tasks. Despite being primarily applicable to financial time series forecasts, they are rarely employed in the field. And also helps to conquer the issues of vanishing and the explosion of gradients in novel recurrent neural networks (RNNs).

3.3. Hybrid Models

ARIMA-LSTM Hybrid: The ensemble algorithm combines the LSTM and ARIMA models according to their specialities, with LSTM applied to non-linear components and ARIMA applied to linear components of the data. Out of all the models analyzed, the hybrid (LSTM-ARIMA) model had the lowest level of metric errors. It outperforms other standalone models with the best MAPE (Mean Absolute Percentage Error) and RMSE (Root Mean Squared Error) [10]. To improve prediction outcomes, a hybrid model was developed, which blends the benefits of LSTM and ARIMA. Like time series data, the autopilot data may be expected to incorporate both nonlinear and linear elements.

3.4. Ensemble Learning Approaches

Guided Methodologies in Time Series Ensemble Forecasting, or GATE: There is a new forecasting approach of ensemble learning that is related to GATE, termed as Guided Methodologies. The distinctness of the time series information may introduce a variation in the performance of the deep learning (DL) models despite their capabilities in the given domain. To avoid this, the GATE model takes advantage of RNN, LSTM, and convolution-LSTM (Conv-LSTM) models by applying an unsupervised learning method to train the ensemble result based on a guided network.

4. PROPOSED METHOD

The proposed model is an implementation of a network that combines both a Long Short-Term Memory (LSTM) network and a multiplicative attention mechanism in a unified and monolithic deep learning-based Time Series forecasting model. The LSTM part performs the task of modelling the sequential dependencies and nonlinear temporal dynamics of complex time-series data. Nonetheless, the conventional LSTM architectures depend on the last hidden state mainly and hence do not enable them to focus on informative historical time steps selectively, especially in long or noisy sequences. To deal with this limitation, a multiplicative attention mechanism is added on top of the LSTM hidden states. This attention module uses a similarity-based interaction to compute the relevance score of every hidden state and thus allows the model to dynamically associate more weight to those time steps. The resultant context vector, which is an expression of a weighted summation of all the hidden states, is a more expressive summary of the sequence than the final hidden state itself. By introducing dynamic temporal selectivity, the unified LSTM-attention architecture improves the model's ability to capture long-range dependencies and nonlinear temporal patterns, which are often poorly represented by conventional LSTM models. Furthermore, the learned attention weights offer interpretability, as they explicitly indicate the relative importance of past time steps in generating predictions.

Technique	Core Drawback	Primary Limitation	Position of this Work
ARIMA / SARIMA	Fundamentally constrained by the linearity assumption and model structure.	Exhibit poor performance on non-linear data and struggle with long-term dependencies.	The LSTM- Multiplicative Attention model overcomes this by harnessing the inherent non-linearity and sequence modelling power of deep learning.
Standard LSTM Networks	Employ uniform time step treatment, forcing memory dilution over long sequences.	Suffer from the fixed-size context vector bottleneck and lack of interpretability in feature selection.	The Multiplicative Attention mechanism is integrated to solve this, enabling the model to dynamically prioritize significant time steps and enhance interpretability.
Hybrid Models (e.g., ARIMA-LSTM)	Inherently involves increased complexity and computational overhead due to multi-component tuning.	Often sacrifices the computational efficiency required for real-time or very large-scale applications.	The LSTM-- Multiplicative Attention model offers a single, unified, end-to-end architecture that achieves superior predictive performance with better efficiency.

Table 1 Comparison of Various Time Series Model

This dual benefit leads to a substantial improvement in both forecasting accuracy and model transparency, making the proposed architecture particularly suitable for complex statements and nonstationary time-series forecasting tasks.

4.1. Architecture Overview

The model has three main stages: The Input Encoding Layer, The Attention Scoring Layer, and the Prediction Output Layer

4.1.1. Encoding Layer (LSTM)

LSTM is the main feature extractor, and it operates on the time sequence of inputs $x = (x_1, x_2, \dots, x_T)$ stepwise. The LSTM at every time produces a hidden state $h = (h_1, h_2, h_3, \dots, h_T)$ that sums up the historical information that it has processed so far.

The complete sequence of all hidden states generated by the LSTM forms the "Source" history, which the attention mechanism can query.

When the input x_t is at every interval t , the LSTM incorporates new gates and states according to the formula below:

Input Gate (i_t)

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i) \quad (1)$$

Forget Gate (f_t)

$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + b_f) \quad (2)$$

Output Gate (o_t)

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + b_o) \quad (3)$$

Candidate Cell State (\tilde{c}_t):

$$\tilde{c}_t = \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \quad (4)$$

Cell State Update (c_t):

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \quad (5)$$

Hidden State Update (h_t):

$$h_t = o_t \odot \tanh(c_t) \quad (6)$$

Where:

x_t input vector at time t .

h_t hidden state at time t .

c_t cell state vector at time t .

W terms depict learnable weights, whereas b terms represent learnable bias vectors.

σ denotes the sigmoid mindfulness.

\odot signifies element-wise (Hadamard) product.

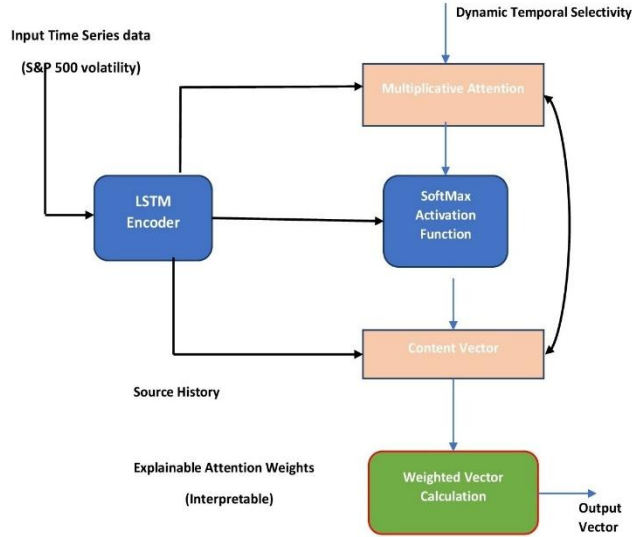


Figure 1 Workflow of LSTM–Multiplicative Attention Model

4.1.2. Attention Scoring Layer

This layer uses the Multiplicative Attention mechanism to calculate a specialized summary of the historical data. The major goal is to identify the most relevant time steps in the past that should be used to predict what is going on at the present. This mechanism matches the output final hidden state of the LSTM (which is the Query) with all past hidden states of the Source history. A comparison gives a score (or energy) of every past time step, which measures its relevance now.

$$e_t = q_t^T W_a h_t \quad (7)$$

Where W_a from Equation 7 is the learnable weight for the multiplicative attention mechanism.

Weight Generation (α_t)

These raw alignment scores are normalised, then by the SoftMax Equation 8 is then applied to the normalised scores to generate attention weights. The weights are a probability distribution that clearly shows the proportional emphasis placed on each observation in the history by the model.

$$\alpha_t = \frac{\exp(e_t)}{\sum_{j=1}^T \exp(e_j)} \quad (8)$$

Where:

e_t attention score at time step t

$\exp(\cdot)$ exponential function

T is the total number of time steps

α_t is attention weight

Context Vector Calculation

A weighted sum of the entire historical sequence of the hidden states is computed, then to yield the final Context Vector using the known Attention Weights. Context vector is an evolutionary, Equation 9 is a very high concentration of the most relevant information on the history of input. This is the most important structural innovation that eliminates the fixed context vector bottleneck that is a part of the traditional LSTMs.

$$ctx_T = \sum_{t=1}^T \alpha_t h_t \quad (9)$$

Where:

h_t is a hidden state

T is the total number of time steps

α_i is attention weight
 ctx_T is context vector at time T

4.1.3. Prediction Output Layer

The Equation 9 (ctx_T) is then mixed with the Query state (h_T) to generate the ultimate forecast \hat{y}_T .

Concatenation:

$$z_T = [h_T; ctx_T]$$

Final Prediction:

$$\hat{y}_T = W_p z_T + b_p$$

Where:

W_p is the matrix of weights
 b_p is the prediction layer's bias vector.

4.2. Enhanced Prediction and Interpretability

The resulting Context Vector is added (concatenated) with the final hidden state of the LSTM to generate an Attention-augmented Vector. This joint vector will have the general trend knowledge and the specific and high-priority information. This refined data is fed into the final fully connected layer to give the forecast output. Enhanced Interpretability is the key to this architecture. Since the Attention Weights are directly computed and normalized, it is possible to visualize them. Such visualizations give an explainable insight into how the given forecast was made, as to what specific past events or time steps were actually paid attention to and assigned a high priority. This openness is essential in instilling faith and usefulness in high-risk forecasting sectors such as finance and healthcare.

5. Experimental Results and Analysis

5.1. Datasets and Pre-processing

To guarantee the validity and the practical usefulness of the findings, the models are trained and tested on two non-linear time series publicly accessible and representing different areas:

5.1.1. Finance – Stock Price Time Series.

Our data is taken on the basis of historical stock prices which are available on the Yahoo Finance platform (S&P 500 and NASDAQ). Each record is made up of daily trading attributes in terms of Open, High, Low, Close, Adjusted Close prices and trading Volume and the associated timestamp. Before modelling, missing values are determined and treated with the forward-fill method of interpolation, and price series are converted to log returns in order to enhance stationarity. The data is divided into chronological 80 training and 20 testing sets. This data is used in both univariate and multivariate time-series prediction and is used as a usual benchmark with which to test financial prediction experiments.

5.1.2. Energy – UCI Electricity Load Diagrams.

The dataset of Electricity Load Diagrams 2011-2014 published by the UCI Machine Learning Repository includes the records of electricity consumption of 370 individual clients in Portugal, measured at 15-minute intervals during 2011-2014. Each observation will have a timestamp and the value of electricity load per meter (in kW) and it will be represented as 370 consumption attributes, the attributes of a single client. So as to minimize variations in the experiments, the data are further summarized by averaging across the 15-minute data to the hourly data. The linear interpolation is used to substitute missing and sparse values. The resultant multivariate time series is divided into training, validation and test based on an 80: 20 chronological split.

5.1.3. Traffic – METR-LA and PeMS-BAY

The measurements are taken at 10-minute frequency and provide various types of climate information which may include temperature, atmospheric pressure, humidity, wind speed/ direction, and solar radiation Our evaluation of the traffic flow forecasting is based on two existing spatio-temporal datasets such as METR-LA and PeMS-BAY. METR-LA has speed measurements for 207 loop detectors installed on Los Angeles County freeways, taken at 5-

minute intervals from March through June 2012. The Caltrans Performance Measurement System (PeMS) has a set of similar sensor readings on the San Francisco Bay Area, which make up the PeMS-BAY. Time of day and the day of the week are added as temporal characteristic to the raw speed/flow data in both datasets. To avoid a look-ahead bias, data are separated into training and validation sets in a chronological manner, and the test set. Such datasets are very strong measurements of multivariate and spatially correlated forecasting problems.

5.1.4. Weather – Jena Climate Dataset

Jena Climate Jena climate is a set of meteorological measurements made by a single weather station in Jena, Germany, over several years. The data set consists of time stamped multivariate climatic characteristics such as air temperature, atmospheric pressure, relative humidity, wind velocity, wind direction, sun radiation, vapor pressure air density, and precipitation. To deliver the time series with an hourly resolution, similar to other data and to compose the computations faster, the data is resampled to hourly data points via mean aggregation. The resulting multivariate time series is split into training, validation and test subsets sequentially. Because the dynamics of the weather processes are extremely nonlinear, this data is effectively applicable in terms of gauging the performance and interpretability of the developed forecasting model.

In all the datasets, there will be an 80:20 split of the data into training and testing sets. The missing values are filled with forward-fill method and z-scoring, which normalizes all data, using mean and SD values of the training set. Also, the Min-Max scaling is used to scale the values of features to the interval [0, 1], the use of which facilitates the removal of the outliers and stabilizes the training of neural networks. The time-series data is then converted to an instance of a supervised learning format with a fixed look-back of 15-time steps.

5.2. Model Configuration and Hyperparameters

From Table 2 clearly specifies that to achieve reproducibility, the exact settings and hyperparameters of each of the three models are listed directly below [20,21], and each neural network is trained over 50 epochs using early termination in the form of validation loss

5.3. Results and Comparative Analysis.

There are three typical error measures of the models based on the Equation 10,11 and 12 Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and Symmetric Mean Absolute Percentage Error (SMAPE).

These measures are quantitative in nature.

$$RMSE = \sqrt{\frac{1}{N} \sum_{t=1}^N (y_t - \hat{y}_t)^2} \quad (10)$$

Model	Hyperparameter	Value	Notes
LSTM-Multiplicative Attention (Proposed)	LSTM Layers/Units	2 Layers / 64 Units each	Optimized for feature extraction complexity.
	Attention Layer Dimensions	64	Matched to the hidden state size for efficient scoring.
	Sequence Length (T)	15	Time steps used for the look-back window.
	Learning Rate	0.001	Utilized the Adam optimizer.
	Batch Size	32	Standard batch size for efficient GPU memory utilization.
Standard LSTM	LSTM Layers/Units	2 Layers / 64 Units each	Matched to the proposed model for a fair comparison.
	Learning Rate	0.001	Same Adam optimizer setting.

ARIMA	Order (p, d, q)	(5, 1, 0)	Selected using the ACF/PACF plots and the Akaike Information Criterion (AIC) on the training set.
	Seasonal Order (P, D, Q)m	Not Applicable	Applied only if seasonality is dominant (e.g., in the Energy dataset).

Table 2 Hyperparameters for Time Series Models

$$MAE = \frac{1}{N} \sum_{t=1}^N |y_t - \hat{y}_t| \quad (11)$$

$$SMAPE = \frac{1}{N} \sum_{t=1}^N \frac{|y_t - \hat{y}_t|}{(|y_t| + |\hat{y}_t|)/2} \quad (12)$$

Where:

N is the total number of observations

y_t is the actual (true) value

\hat{y}_t Predicted value at time

Table 3 shows the results that the linear ARIMA model does not perform well for two non-linear datasets, which has proven it to be inherently limited. The Standard LSTM network forms a major reduction in error on all the metrics due to its incredible properties that enable it to model the long-term time variations. Importantly, the LSTM, which incorporates the Multiplicative Attention Mechanism, provides the lowest error rates continuously. This confirms the overall hypothesis that the Multiplicative Attention mechanism of flexibly focusing its attention on the time steps, which are best suited to the sequence, is crucial to enhance the accuracy.

Model	RMSE	MAE	SMAPE
ARIMA	12.45%	10.23%	18.65%
LSTM	7.30%	5.90%	11.48%
LSTM with Multiplicative Attention	5.75%	4.62%	8.93%

Table 3 Comparison of metric errors of ARIMA, LSTM, and LSTM with Multiplicative Attention

5.4. Visual Interpretation

Figure 2 provides a comparison of bar charts with the performance of three forecasting models, ARIMA, LSTM, and LSTM with Multiplicative Attention, upon a nonlinear time series dataset. There exist three common metrics of error [24], which are applied to these models Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and Symmetric Mean Absolute Percentage Error (SMAPE). As it is in the Comparison, the ARIMA model has the highest error values in all three measures, with RMSE being 12.45, MAE being 10.23, and SMAPE being 18.65. This reiterates the recognized weakness of ARIMA to reflect the complex, non-linear trends because of its linearity. The LSTM algorithm [25], predicts errors up to a great extent, and RMSE is 7.30, MAE is 5.90, and SMAPE is 11.48 percent. The reason behind this enhancement is that LSTM has the inbuilt ability to present a temporal relationship of sequential information over a long duration. In particular, LSTM with the Multiplicative Attention mechanism achieved the highest rate of the smallest errors concerning all of the parameters with an RMSE of 5.75, MAE of 4.62, and SMAPE of 8.93%. The combination of the multiplicative attention mechanism introduces additional refinements that confirm its effectiveness in reducing error rate and a model that is resilient to mechanisms achieved the smallest errors concerning all of the parameters with an RMSE of 5.75, MAE of 4.62, and SMAPE of 8.93%. The combination of the multiplicative attention mechanism introduces additional refinements that confirm its effectiveness in reducing error rate and a model that is resilient.

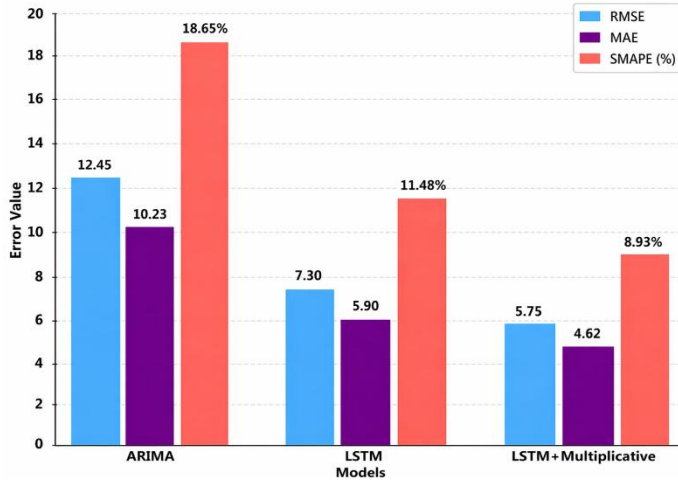


Figure 2 Comparison of performance of various forecasting models

.Figure 3 presents a comparison of ARIMA, LSTM and the proposed LSTM that included the Multiplicative Attention model and the actual time series data regarding forecasting performance. The truth is displayed by the black curve, and ARIMA, LSTM, and LSTM with Multiplicative Attention predictions have the red dashed, blue dashed, and green dashed lines, respectively. One can note that the ARIMA model shows significant deviation, especially at major peaks and troughs, when compared to major dynamics that are nonlinear and therefore complicated to model [26,27]. The truth is displayed by the black curve, and ARIMA, LSTM, and LSTM with Multiplicative Attention predictions have the red dashed, blue dashed, and green dashed lines, respectively. One can note that the ARIMA model shows significant deviation, especially at major peaks and troughs, when compared to major dynamics that are nonlinear and therefore complicated to model [26,27]. Nevertheless, it is also characterized by certain latency in sudden turns. Conversely, the LSTM with Multiplicative Attention has the best predictive ability and effectively meets the actual series and captures both world trends and local changes. These findings agree that the addition of a Multiplicative mechanism of attention helps the LSTM to actively concentrate on the relevant past observations [28,29]. So as to increase its robustness and minimize errors in forecasting. All in all, the results attest to the excellence of deep learning-based solutions that specifically LSTM [30] with Multiplicative Attention is more effective than conventional statistical techniques like ARIMA in forecasting nonlinear time series.

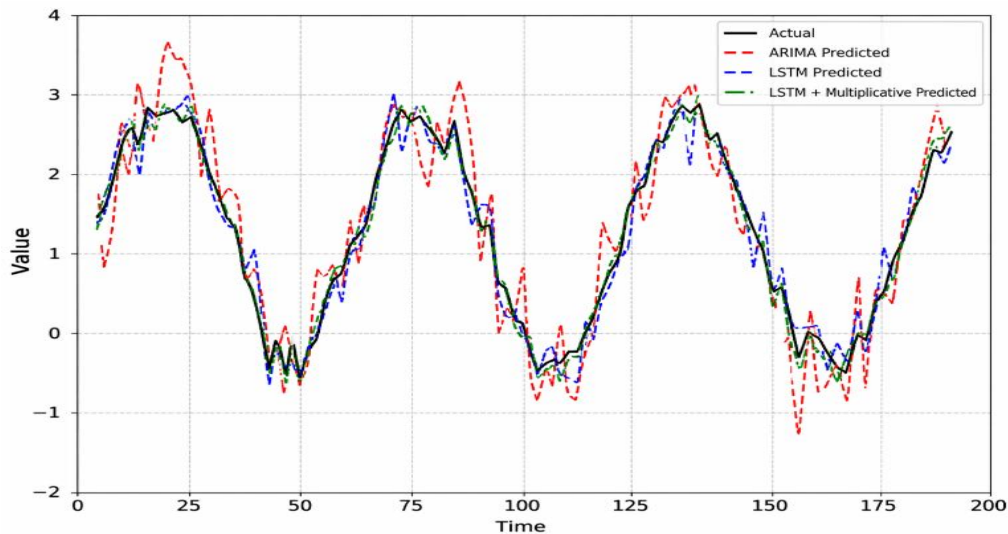


Figure 3 Comparison of predicted with actual values using ARIMA, LSTM, and LSTM-Multiplicative Attention.

6. Conclusion

LSTM networks combined with the Multiplicative Attention mechanism are an interesting and requisite addition to the field of time series forecasting. This method effectively addresses the two underlying flaws of conventional statistical models, such as ARIMA, which are incapable of handling long and nonlinear interactions. The principal scientific input of the work is the empirical validation that the Multiplicative Attention-LSTM architecture can effectively model complex, non-linear time series. The Multiplicative Attention mechanism facilitates the explainability and predictive performance of the model by emphasizing the important time steps. This is well justified by the proposed analysis: the proposed model Root Mean Squared Error (RMSE) of 5.75, Mean Absolute Error (MAE) of 4.62 and Symmetric Mean Absolute Percentage Error (SMAPE) of 8.93 in the simulated non-linear data. This performance improvement is significant and shows a significant improvement in precision compared to the classic LSTM model (RMSE of 7.30%) and also compared to the classic ARIMA model (RMSE of 12.45%). This fact proves that, due to the capacity to adjust the weights of importance dynamically, the Multiplicative Attention-LSTM model is effective at capturing the linear and non-linear trends, and provides a better level of accuracy and flexibility in the field of financial forecasting as well as smart farming. Future studies should concentrate on creating more efficient algorithms to lower the computational complexity and memory requirements of Multiplicative Attention-LSTM models for large-scale, real-time applications. There is also a need for further customization to suit specific domain characteristics and to enhance the explainability of the attention weights for critical decision-making procedures.

Acknowledgment

There are no funding agencies or organizations supporting this work.

Conflicts of interest

No conflicts of interest are disclosed by the writers.

Data availability

In this study, several publicly available datasets are utilized including Kaggle, UCI Repository, Zendo.

These datasets are accessible through the following sources:

<https://www.kaggle.com/datasets/paultimothymooney/stock-market-data>

<https://archive.ics.uci.edu/dataset/321/electricityloaddiagrams20112014>

<https://zenodo.org/records/5146275>

<https://www.kaggle.com/datasets/mnassrib/jena-climate>

References

1. M.Bhuvaneshwari, S. Kumaravel, " Time series based adaptive forecasting model for equity market price prediction," AIP Conference Proceedings, vol. 3193, No. 1, 2024.
2. Y. Touzani, K. Douzi, "An LSTM and GRU based trading strategy adapted to the Moroccan market," Journal of Big Data, vol. 8, no. 1, 2021
3. K. Gajamannage, Y. Park, D. I. Jayathilake, "Real-time forecasting of time series in financial markets using sequentially trained dual-LSTMs," Expert systems with applications, vol. 223, pp. 119879, 2023.
4. M. Waqas , U. W. Humphries, "A critical review of RNN and LSTM variants in hydrological time series predictions," MethodsX, vol. 13, pp. 102946, 2024.
5. Y. Ning, H. Kazemi, P. Tahmasebi, "A comparative machine learning study for time series oil production forecasting: ARIMA, LSTM, and Prophet," Computers & Geosciences, vol. 164, pp. 105126, 2022.
6. M. R. N. Rad, S. S. Gerdefaramarzi, H. A. Akano, A. Cheema, "Seasonal Autoregressive Integrated Moving Average (SARIMA) for Melon (Cucumis melo) Yield from 2011 to 2020 Based on Planting Date Period," Current Agriculture Research Journal, vol. 11, no. 3, pp. 753–760, 2024.
7. R. Khaldi, A. El Afia, R. Chiheb, S. Tabik, "What is the best RNN-cell structure for forecasting each time series behavior," arXiv.org, vol. abs/2203.07844, 2022
8. E. Dave, A. Leonardo, M. Jeanice, N. Hanafiah, "Forecasting Indonesia Exports using a Hybrid Model ARIMA-LSTM," Procedia Computer Science, vol. 179, pp. 480–487, 2021.
9. Tian, J. Shi, H. Zhang, X. Wei, A. Zhao, J. Feng, "Regularized Extreme Learning Machine-Based Temperature Prediction for Edible Fungi Greenhouse," Applied Engineering in Agriculture, vol. 40, no. 4, pp. 483–499, 2024.
10. Y. Wang and Y. Guo, "Forecasting method of stock market volatility in time series data based on mixed model of ARIMA and XGBoost," China Communications, vol. 17, no. 3, pp. 205–221, 2020.

11. S. Khan and H. Alghulaiakh, "ARIMA Model for Accurate Time Series Stocks Forecasting," *International Journal of Advanced Computer Science and Applications*, vol. 11, no. 7, 2020.
12. T. Fang and R. Lahdelma, "Evaluation of a multiple linear regression model and SARIMA model in forecasting heat demand for district heating system," *Applied Energy*, vol. 179, pp. 544–552, 2016.
13. U. M. Sirisha, M. C. Belavagi, and G. Attigeri, "Profit Prediction Using ARIMA, SARIMA and LSTM Models in Time Series Forecasting: A Comparison," *IEEE Access*, vol. 10, pp. 124715–124727, 2022
14. Sravanthi, Chitirala, "DeepLGENet: A Novel Multilayered Deep Learning Model for Accurate Epileptic Seizure Forecasting from EEG Signals via LSTM and a Bi-GRU with Attention." *International Journal of Intelligent Engineering & Systems*, vol. 18, no. 9, pp. 191–210, 2025.
15. Y.C.Jin, Q. Cao, K. -N. Wang, Y. Zhou, Y. P. Cao and X. Y. Wang, "Prediction of COVID-19 Data Using Improved ARIMA-LSTM Hybrid Forecast Models," *IEEE Access*, vol. 11, pp. 67956-67967, 2023.
16. X. Ran, Z. Shan, Y. Fang, and C. Lin, "An LSTM-Based Method with Attention Mechanism for Travel Time Prediction," *Sensors*, vol. 19, no. 4, pp. 861, 2019.
17. Widiyanto, Aditya Januar, Didit Adytia, and Indra A. Aditya. "Feature Modelling for Electricity Load Forecasting Using Hybrid EEMD-BiLSTM with Spatially Correlated Weather Data." *International Journal of Intelligent Engineering & Systems*, vol. 17, no. 6, pp. 1193–1211, 2024.
18. MK. Saravana, MS. Roopa, JS Arunalatha, and KR. Venugopal, "Towards Robust Multivariate Time Series Analysis: A Novel Attention-based Deep Learning Framework." *International Journal of Intelligent Engineering & Systems*, vol. 18, no. 5, pp. 474–492, 2025.
19. E. B. Kim, J. Hoon Park, Y.S. Lee, and C. Lim, "Two-dimensional attention-based multi-input LSTM for time series prediction," *Communications for Statistical Applications and Methods*, vol. 28, no. 1, pp. 39–57, 2021.
20. Y. Akkem, B. S. Kumar, and A. Varanasi, "Streamlit Application for Advanced Ensemble Learning Methods in Crop Recommendation Systems – A Review and Implementation," *Indian Journal Of Science And Technology*, vol. 16, no. 48, pp. 4688–4702, 2023.
21. P. Jia, H. Chen, L. Zhang, and D. Han, "Attention-LSTM based prediction model for aircraft 4-D trajectory," *Scientific Reports*, vol. 12, no. 1, 2022.
22. J. Fattah, L. Ezzine, Z. Aman, H. El Moussami, and A. Lachhab, "Forecasting of demand using ARIMA model," *International Journal of Engineering Business Management*, vol. 10, 2018.
23. S. Bouktif, A. Fiaz, A. Ouni, and M. A. Serhani, "Multi-Sequence LSTM-RNN Deep Learning and Metaheuristics for Electric Load Forecasting," *Energies*, vol. 13, no. 2, pp. 391, 2020.
24. Md. R. Sarkar, "GATE: A guided approach for time series ensemble forecasting," *Expert Systems with Applications*, vol. 235, pp. 121177, 2023.
25. E. Balouji, J. Sjoblom, N. Murgovski, and M. H. Chehrehgani, "Prediction of Time and Distance of Trips Using Explainable Attention-based LSTMs," *arXiv.org*, vol. abs/2303.15087, 2023.
26. S. Suman and U. Verma, "State space modelling and forecasting of sugarcane yield in Haryana, India," *Journal of Applied and Natural Science*, vol. 9, no. 4, pp. 2036–2042, 2017.
27. B. Pandey, A. Shukla, and A. Khamparia, "An Optimized Hybrid ARIMA-LSTM Model for Time Series Forecasting of Agricultural Production in India," *Microbial Data Intelligence and Computational Techniques for Sustainable Computing*, pp. 107–119, 2024.
28. Y. Cheng and Y. Morimoto, "Triple-Stage Attention-Based Multiple Parallel Connection Hybrid Neural Network Model for Conditional Time Series Forecasting," *IEEE Access*, vol. 9, pp. 29165-29179, 2021.
29. B. Yu, J. Li, C. Liu, and B. Sun, "A novel short-term electrical load forecasting framework with intelligent feature engineering," *Applied Energy*, vol. 327, pp. 120089, 2022.
30. Pathania, Nahita, and Balraj Singh. "Design of an Integrated Model Using Hybrid Autoencoder and LSTM for Fault Tolerance and Load Balancing in Cloud Environments." *INTERNATIONAL JOURNAL* 11.6 (2024): 954-971.
31. B. Dilip Kumar Reddy is a research scholar at Dr MGR Educational and Research Institute, Chennai, Tamil Nadu. Received B. Tech Degree From JNTUA in the year of 2009 and Pursued his M. Tech from the Dr MGR University in the year of 2011. And having of 16 years of teaching experience at G.Pulla Reddy Engineering College, Andhra Pradesh.
32. Rajavarman holds a PhD in Computer Science and Engineering from Dr MGR Educational and Research Institute, Chennai, Tamil Nadu. And he is a life member of the Computer Society of India (CSI). He has 25 years of experience in teaching and 11 years in the research field. He published 20 papers in international and national journals.