Acadlore Transactions on AI and Machine Learning
https://www.acadlore.com/journals/ATAIML

Multi-Variable Time Series Decoding with Long Short-Term Memory and Mixture Attention

Soukaina Seddik*, Hayat Routaib, Anass Elhaddadi

Applied Science Laboratory LSA, ENSAH, Abdelmalek Essaadi University, 32000 Al Hoceima, Morocco

* Correspondence: Soukaina Seddik (seddik.soukaina@etu.uae.ac.ma)

Received: 08-18-2023 **Revised:** 09-10-2023 **Accepted:** 09-16-2023

Citation: S. Seddik, H. Routaib, and A. Elhaddadi, "Multi-variable time series decoding with Long Short-Term Memory and mixture attention," *Acadlore Trans. Mach. Learn.*, vol. 2, no. 3, pp. 154–169, 2023. https://doi.org/10.56578/ataiml020304.

Abstract: The task of interpreting multi-variable time series data, while also forecasting outcomes accurately, is an ongoing challenge within the machine learning domain. This study presents an advanced method of utilizing Long Short-Term Memory (LSTM) recurrent neural networks in the analysis of such data, with specific attention to both target and exogenous variables. The novel approach aims to extract hidden states that are unique to individual variables, thereby capturing the distinctive dynamics inherent in multi-variable time series and allowing the elucidation of each variable's contribution to predictive outcomes. A pioneering mixture attention mechanism is introduced, which, by leveraging the aforementioned variable-specific hidden states, characterizes the generative process of the target variable. The study further enhances this methodology by formulating associated training techniques that permit concurrent learning of network parameters, variable interactions, and temporal significance with respect to the target prediction. The effectiveness of this approach is empirically validated through rigorous experimentation on three real-world datasets, including the 2022 closing prices of three major stocks - Apple (AAPL), Amazon (AMZN), and Microsoft (MSFT). The results demonstrated superior predictive performance, attributable to the successful encapsulation of the diverse dynamics of different variables. Furthermore, the study provides a comprehensive evaluation of the interpretability outcomes, both qualitatively and quantitatively. The presented framework thus holds substantial promise as a comprehensive solution that not only enhances prediction accuracy but also aids in the extraction of valuable insights from complex multi-variable datasets.

Keywords: Neural network; Deep learning; Long Short-Term Memory; Interpretable Multi-Variable Long Short-Term Memory; Prediction

1 Introduction

Over recent years, Recurrent Neural Networks (RNNs) have emerged as potent instruments for the analysis of time series data, encompassing both target and exogenous variables [1]. The objectives of such investigations have expanded beyond mere predictive accuracy to encompass the extraction of discernible insights into the underlying patterns governing these intricate temporal datasets [2, 3].

The scrutiny of time series data has garnered considerable attention, propelled by the necessity to decipher meaningful insights from complex temporal patterns [4, 5]. RNNs have surfaced as a cornerstone in this domain, facilitating the modeling of sequential dependencies across varied applications, from natural language processing to financial forecasting [6]. In particular, Long Short-Term Memory (LSTM) networks have exhibited extraordinary competencies in capturing long-range dependencies and alleviating the vanishing gradient problem [7].

Further, the incorporation of exogenous variables into LSTM-based models, known as LSTM with exogenous inputs or LSTM with input, memory, and output gates for Variable Specific Hidden State (termed as Interpretable Multi-Variable LSTM or LSTM-IMV), has augmented their applicability to numerous real-world scenarios [8, 9]. This introduction offers an overview of LSTM-IMV models, underlining their importance and recent advancements.

The utility of LSTM RNNs in time series analysis has gained prominence due to their capacity to encapsulate long-range dependencies and intricate temporal associations [10–12]. Capitalizing on this potential, this research introduces an innovative mixture attention mechanism specifically designed to elucidate the generative process of the target variable. This novel approach transcends conventional attention mechanisms by providing insights into the complex interplay between variables, thereby enhancing the interpretability of predictions.

The study delves into the architecture of LSTM RNNs, with the aim of uncovering hidden states that are specific to individual variables within multi-variable time series data. In doing so, it seeks to capture the distinct dynamics that are intrinsic to each variable and discern their contributions to predictive outcomes.

To enable the proposed approach, this study develops sophisticated training techniques that facilitate the concurrent learning of network parameters, variable interactions, and temporal significance in relation to the target prediction. This comprehensive learning framework shows promise in providing a thorough understanding of the complex dynamics at play within multi-variable time series. The research aims to encapsulate the diverse dynamics exhibited by different variables, striving to enhance predictive performance across an array of three real-world datasets, including the closing prices of three stocks - Apple (AAPL), Amazon (AMZN), and Microsoft (MSFT). The efficacy of the proposed approach is appraised through extensive experimentation across several real-world datasets. By effectively capturing the intricate dynamics within multi-variable time series, the framework demonstrates improved predictive performance. Furthermore, this study conducts an exhaustive evaluation of the interpretability outcomes, utilizing both qualitative and quantitative measures. This assessment underscores the potential of the framework as a dual-purpose solution: facilitating accurate forecasting while enabling the extraction of meaningful insights from complex multi-variable datasets.

This research contributes to the evolving field of time series analysis by introducing a novel LSTM-based approach with an emphasis on capturing variable-specific dynamics and elucidating their contributions to predictive outcomes. The remainder of this paper delves into the technical details of the proposed methodology, the formulation of training techniques, experimental setups, and the presentation of results. Through this comprehensive investigation, the study aims to establish the proposed framework as a valuable asset for researchers and practitioners working with multi-variable time series data.

In summary, the principal contributions and discoveries of this paper can be encapsulated as follows:

• Introduction of a cutting-edge Drift-Adaptive Long Short-Term Memory (DA-LSTM) framework for interval load forecasting.

• Innovative integration of a mixture attention mechanism and drift adaptation techniques.

• An adaptive LSTM network that swiftly adopts emerging consumption trends while retaining previous knowledge.

• Rigorous evaluation demonstrates superior performance in interval load forecasting.

• Advancements in interpretable multi-variable time series analysis through the synergy of techniques.

The research undertaken within a Python-powered environment, leveraging the capabilities of Jupyter Notebook and the TensorFlow Keras library, yielded profound insights into the realm of time series forecasting, specifically in the context of stock price prediction. The study meticulously analyzed historical stock price data from three industry giants, namely, Apple Inc. (AAPL), Amazon.com Inc. (AMZN), and Microsoft Corporation (MSFT), spanning from January 1, 2020, to January 1, 2023. The analysis hinged on the 'Close' prices of these stocks, which served as the primary feature for scrutiny.

The creation and utilization of Interpretable Multi-Variable LSTM models for each stock demonstrated their ability to effectively encapsulate temporal patterns and enhance interpretability via self-attention mechanisms. These models were trained by minimizing evaluative criteria such as Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and Mean Absolute Percentage Error (MAPE), leading to a consistent decline in loss values and signifying an enhancement in predictive precision with each training epoch.

One notable facet of this research was the visual interpretation of attention weights, represented as heatmaps. This provided an illuminating perspective into the portions of data the model prioritized during its predictive process. These visualizations served as a key tool in understanding the temporal relationships and dependencies that guided the model's predictive decision-making.

Further, the study facilitated a comparative analysis of the actual versus the forecasted closing prices for AAPL, MSFT, and AMZN stocks in 2022. A range of LSTM model approaches, including IMV-Full-P, IMV-Tensor-P, and others, were utilized, providing a panoramic view of their predictive prowess.

In terms of performance, the IMV-Tensor approach outshone its counterparts, highlighting the potential of independent variable-wise hidden states. In addition, the study introduced an innovative ranking method based on the Pearson correlation coefficient, which prioritized variables having the highest correlations with the target variable, thereby augmenting predictive accuracy.

This study serves as a significant contribution to the arena of time series forecasting, especially in the sphere of financial analytics. It underscores the efficacy of Interpretable Multi-Variable LSTM models and their attention mechanisms in the accurate prediction of stock prices. The insights gleaned from this research are invaluable for financial decision-making and portfolio management, emphasizing the potential of advanced, interpretive, and accurate forecasting models.

The subsequent sections of this paper are organized as follows: Section 2 provides an exhaustive review of

Interpretable Machine Learning and Multi-Variable Time Series Forecasting, spotlighting their respective contributions towards the overarching theme of Interpretable Time Series Forecasting. Section 3 elucidates the methodology underpinning our Interpretable Multi-Variable (IMV) LSTM model, encompassing aspects such as data acquisition, training regimen, and the predictive framework. Section 4 introduces the simulation setup and engages in a detailed discussion of the results, serving to illustrate the superior predictive prowess and interpretability offered by the IMV-LSTM model. Finally, Section 5 brings the paper to a close by encapsulating the salient contributions, emphasizing the novelty of our mixture attention mechanism, and underscoring the exceptional predictive precision and interpretative clarity achieved by the IMV-LSTM model.

2 Literature Review

Rooting this research in its wider context necessitates an appreciation for the landmark work in the fields of LSTM networks, time series forecasting, and interpretable machine learning. Unquestionably, the pioneering efforts of individuals like Hochreiter and Schmidhuber [13] have shaped the role of LSTM networks in time series forecasting. Simultaneously, the vast realm of interpretable machine learning techniques has been meticulously explored by thought leaders such as Ribeiro et al. [14] and Lundberg and Lee [15]. Standing at the crossroads of these domains, this study leverages prior research as a springboard, introducing an innovative methodology that not only boosts predictive performance but also unveils invaluable insights into the underlying dynamics of multi-variable time series data [16].

2.1 Interpretable Time Series Forecasting

While LSTM networks have demonstrated exceptional predictive capabilities, their intricate nature often cloaks them in opacity. This has led researchers to embark on the quest for techniques that can infuse transparency into these models, particularly in the domain of time series prediction. An example of this is Dal et al. [17] who devised a methodology that elucidates the predictions of any classifier, LSTM-based models included. Their approach generates locally interpretable explanations by approximating complex models with simpler, more comprehensible ones [18]. Such initiatives have laid the foundation for the application of interpretable machine learning to time series forecasting, thereby facilitating model comprehension and validation.

2.2 Multi-Variable Time Series Forecasting

In the landscape of real-world data, time series often encompass multiple variables that interact in complex, intricate patterns. LSTM networks have been adapted to accommodate multi-variable time series forecasting, where the goal is to predict one variable based on the historical data of multiple variables. Chen et al. [18] pioneered the Network of multi-input LSTM for Predicting Financial Time Series. However, the challenge of discerning the individual contributions of various variables to the prediction persists as a critical issue. The imperative to offer interpretable insights into the dynamics of different variables in a multi-variable setting calls for innovative methodologies.

The demand for interpretable machine learning has surged as complex models, such as deep neural networks, find increasing application in contexts requiring critical decision making. Interpretable machine learning strives to provide transparency into the internal workings of these models, aiming to render their predictions more comprehensible and trustworthy. Table 1 shows the implementation of LSTM via techniques like LSTM Variational AutoEncoder (LSTM-VAE) [19] and a CNN-LSTM network that employs both convolutional and LSTM layers to extract knowledge from the training data. Zha et al. [20] devised models to elucidate the features driving the predictions.

Table 1. Previous LSTM based models in literature

Year	Authors	Proposed Model
2022	Zhao et al. [19]	LSTM-VAE
2022	Zha et al. [20]	CNN-LSTM
2021	Fan et al. [21]	ARIMA-LSTM
2017	Zhu et al. [22]	Time-LSTM
2017	Zheng et al. [23]	EMD-LSTM
2021	Shi and Chehade [24]	Dual-LSTM
2020	Istiake Sunny et al. [25]	Bi-Directional LSTM

In the realm of time series prediction, Fan et al. [21] utilized an ARIMA-LSTM model, while Zhu et al. [22] employed the TIME-LSTM model, which focuses on temporal dynamics by accentuating the sequential dependencies in time series data. By customizing LSTM architectures to capture these temporal elements, the model aspires to provide clearer insights into time-evolving patterns.

Zheng et al. [23] used the EMD-LSTM (Empirical Mode Decomposition - LSTM) model, which integrates LSTM Networks with Empirical Mode Decomposition (EMD). EMD is employed to extract intrinsic oscillatory modes from data, which are then fed into the LSTM for forecasting. This strategy aims to offer interpretable insights by decomposing complex signals.

Shi and Chehade [24] introduced the Dual-LSTM model, implementing a dual structure that comprises two separate LSTM networks. This architecture aims to enhance interpretability by explicitly capturing the influence of exogenous variables and endogenous factors on the forecasting process.

Istiake Sunny et al. [25] developed the Bi-Directional LSTM model that enhances LSTM architectures by processing input sequences in both forward and reverse directions. This bidirectional analysis aims to capture intricate relationships between variables and provide more comprehensive insights for forecasting.

The merger of LSTM recurrent neural networks with interpretability techniques holds the promise to significantly transform multi-variable time series forecasting. This study, grounded in the fundamentals of LSTM networks and the advancements in interpretable machine learning, introduces an innovative mixture attention mechanism to glean insights into the dynamics of individual variables. This approach not only amplifies the accuracy of predictions but also provides decision-makers with understandable explanations. The exploration of this convergence is anticipated to bridge the divide between prediction precision and interpretability, paving the way for applications across a multitude of domains.

In this paper, our focus is centered on deploying the Interpretable Multi-Variable LSTM (IMV-LSTM) to manage multiple input variables concurrently, as exemplified by our work with the closing prices of three stocks—AAPL, AMZN, and MSFT. IMV-LSTM models are engineered to address some of the limitations of traditional LSTM models in time series forecasting by offering enhanced interpretability and potential performance improvements. This model can help address the gaps or limitations inherent in LSTM-based time series forecasting, as outlined in the subsequent Table 2.

Table 2. The comparison between LSTM & IVM-LSTM

LSTM Limitations	IVM-LSTM Overcoming
Interpretability	IMV-LSTM models enhance interpretability through mechanisms like attention and feature importance analysis, overcoming the black-box nature of traditional LSTMs. This transparency is vital in financial forecasting, enabling better risk assessment and informed decision-making by providing insights into the reasoning behind predictions.
Incorporating Multiple Variables	IMV-LSTM models excel in handling multiple input variables concurrently, a crucial advantage in stock price forecasting where prices are influenced by diverse factors like market sentiment, economic indicators, and news events. This capability goes beyond the traditional LSTM approach, which often focuses solely on historical prices, leading to enhanced forecasting accuracy.
Handling Non-Stationarity	IMV-LSTM models can incorporate additional features or transformations to handle non-stationary data effectively. This is important in financial forecasting, as stock prices often exhibit non-stationary behavior.
Uncertainty Estimation	IMV-LSTM models can provide insights into prediction uncertainty. This addresses the limitation of traditional LSTMs, which typically offer point forecasts. Knowing the degree of uncertainty associated with predictions is essential in risk management and portfolio optimization.
Anomaly Detection	IMV-LSTM models can be enhanced to detect anomalies or outlier data points in time series data. This is valuable in financial forecasting to identify abnormal market behavior or exceptional events that may impact stock prices.
Incorporating External Data	IMV-LSTM models can easily incorporate external data sources, such as news sentiment analysis or macroeconomic indicators, to improve forecasting accuracy and robustness. This addresses the limitation of traditional LSTMs, which often rely solely on historical price data.
Regularization Techniques	IMV-LSTM models may introduce novel regularization techniques to mitigate overfitting, a common challenge in financial forecasting when dealing with limited historical data.
Model Robustness	By considering multiple variables and providing interpretability, IMV-LSTM models can potentially be more robust to changing market conditions and external factors.

To further substantiate our work, we have compared our model with those developed by other researchers, leading to the following insights:

• LSTM-VAE marries LSTM and VAE, yielding probabilistic and interpretable forecasting.

• CNN-LSTM amalgamates CNN and LSTM, facilitating spatial-temporal pattern recognition.

• ARIMA-LSTM merges ARIMA and LSTM, equipping the model to address linear trends and complex dependencies.

• Time-LSTM is a specialized LSTM variant tailored for efficient time series modeling.

• EMD-LSTM leverages Empirical Mode Decomposition before LSTM to handle non-stationary data.

• Dual-LSTM utilizes two parallel LSTM layers to boost bidirectional understanding.

• Bi-Directional LSTM processes data in both forward and backward directions in time, providing bidirectional insights.

Conversely, IMV-LSTM concentrates on enhancing interpretability and accuracy in time series forecasting. It encompasses multiple variables, tackles non-stationarity, delivers uncertainty estimates, and ensures transparency in model predictions. Therefore, the choice of model largely depends on the specific characteristics of the time series data and the objectives of forecasting. The most appropriate model may vary across different applications, thereby underscoring the need to consider factors such as data type, dimensionality, seasonality, and the demand for interpretability when selecting a model for time series forecasting.

3 Methodology

Crafting an "Interpretable Multi-Variable (IMV) LSTM" necessitates the integration of LSTM architecture with methods that bolster interpretability. This can be accomplished by leveraging attention mechanisms and visualization techniques, which illuminate the process through which the model navigates different variables over time.

3.1 IMV-LSTM

In this section, we delineate the formulation of the IMV-LSTM architecture, amalgamating the prowess of LSTM networks with interpretable attention mechanisms to augment predictive precision and provide insights into the contributions of individual variables. We kick-off by outlining the standard LSTM equations, subsequently introducing the innovative mixture attention mechanism.

LSTM [26] is classified under the umbrella of recurrent neural networks (RNNs), yet it stands apart due to its extended long-term memory capabilities. This contrasts with traditional RNNs, which utilize recurrent cells such as sigma, which can be depicted as [27]:

$$h_{t=}\sigma\left(Zh_{t-1}+Px_t+b\right) \tag{1}$$

where, x_t represents the input, while h_t and y_t denote the recurrent information and cell output at a specific time t, respectively. The weights, represented by Z and P, along with the bias b, contribute to these computations. While the usage of standard RNN cells has demonstrated success in applications such as sentiment analysis or image classification, they frequently face difficulties in adequately handling long-term dependencies. In contrast, LSTM networks excel at preserving values from earlier stages for future utilization, addressing the challenge of vanishing gradients [13]. The vanishing gradient problem arises when gradient information fails to propagate back to the model's input layers due to the characteristics of activation functions. For instance, the sigmoid function maps significant input values to a range spanning from 0 to 1. As a result, substantial changes in input lead to minor adjustments in output, resulting in diminished derivatives that may even vanish. The LSTM network is composed of memory cells that retain information across varying time steps. The computation within an LSTM cell is governed by the following equations:

$\rightarrow$ **The input gate (*i*)** integrates the current input x_t produced from the previous LSTM cell h_{t-1} and the cell state C_{t-1} The process is captured by the equation:

$$i_t=\sigma\left(P_i x_t+Z_i h_{t-1}+W_{ci}\odot c_{t-1}+b_i\right) \tag{2}$$

Here, the $\odot$ sign represents element-wise multiplication between vectors. P_i, Z_i and W_{ci} are the respective weights related to x_t, h_{t-1} and C_{t-1} Additionally, the bias vector bi is linked with this element. The information retained within the cell states c_t is influenced by the LSTM layer that comes before.

$\rightarrow$ **Forget Gate (*f*)** is responsible for deciding which information needs to be discarded from the previous cell states C_{t-1}. As a result, the computed activation values ft using the present input x_t, h_{t-1} the previous outputs and the memory cell state of the memory cells C_{t-1} at the previous time step h_{t-1}.

$$f_{\mathrm{t}}=\sigma\left(W_f x_t+W_{hf}h_{t-1}+W_{cf}\odot c_{t-1}+b_f\right) \tag{3}$$

where, W_f, W_{hf}, and W_{cf} correspond to the weights linked with x_t, h_{t-1} and C_{t-1}, in that order. Additionally, b_f signifies the bias vector.

$\rightarrow$ **Cell State (*c*)** merges the input values from Y_t, i_t the input gate, and f_t the forget gate, along with the preceding cell value:

$$c_{\mathrm{t}} = f_{\mathrm{t}} \odot c_{t-1} + i_t \odot Y_{\mathrm{t}} \tag{4}$$

where,

$$Y_{\mathrm{t}} = \tanh\left(W_{xc}x_t + W_{hc}h_{t-1} + b_c\right) \tag{5}$$

$\rightarrow$ **Output Gate (*o*)** integrates the present input x_t, h_{t-1} the output from the previous unit, and the cell value from the previous iteration:

$$o_t = \sigma\left(P_o x_t + Z_o h_{t-1} + W_{co} \odot c_t + b_o\right) \tag{6}$$

where, P_o, Z_o, W_{co} represent the weights connected to x_t, h_{t-1} and c_t, respectively, while b_o indicates the bias weight vector.

$\rightarrow$ **Hidden State (*h*)**

The calculation of the hidden state at a particular time step, denoted as h_t is represented as follow:

$$h_t = o_t * \tanh\left(c_t\right) \tag{7}$$

o_t refers to the output gate at the current time step. tanh (c_t) corresponds to the hyperbolic tangent activation applied to the cell state at the current time step, denoted as c_t. The cell state carries information over time, and applying the tanh function helps regulate the values within a certain range, typically between -1 and 1. This step is important for maintaining the gradient flow during training. So, the expression h_t=o_t**tanh* (c_t) means that h_t the hidden state at the current time step is calculated by multiplying the output gate o_t with the hyperbolic tangent of the cell state c_t. This effectively determines how much information is allowed to pass through to the output based on the output gate's value and how the cell state's information is transformed using the tanh function.

3.1.1 Mixture attention mechanism

To imbue our model with interpretability, we introduce a novel mixture attention mechanism into the LSTM architecture. The objective is to spotlight the significance of each variable within the multi-variable time series data. This mechanism is actualized via the following equation:

$$A_t = \operatorname{softmax}\left(W_{\text{att}} * \left[x_t, h_{t-1}\right]\right) \tag{8}$$

where, A_t is the attention vector at time step t, W_{att} is the attention weight matrix, and softmax is the softmax activation function. The attention vector A_t assigns different weights to the individual variables based on their relevance to the prediction at the current time step.

3.1.2 Interpretable Multi-Variable LSTM

The Interpretable Multi-Variable LSTM combines the LSTM cell computations with the mixture attention mechanism. The hidden state h_t is calculated as:

$$h_t = A_t * h_t \tag{9}$$

This adjustment, as illustrated in Figure 1 of LSTM with hidden vectors, ensures that the hidden state is modulated by the attention weights, accentuating the contributions of different variables based on their relevance to the prediction task. In this section, we have delineated the Interpretable Multi-Variable LSTM architecture, marrying the foundational LSTM computations with an innovative mixture attention mechanism. This formulation amplifies both predictive accuracy and interpretability, empowering the model to yield insights into the dynamics of individual variables within a multi-variable time series context.

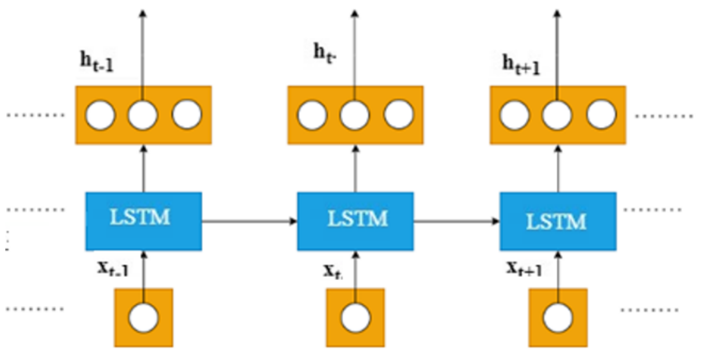

Figure 1. LSTM with hidden vectors

3.2 Network Architecture

Figure 2 showcases the network architecture of the IMV-LSTM model. This architectural diagram encapsulates the organization of various layers and components that form the IMV-LSTM model, providing a glimpse into its structural composition. Starting from the left, the input data is introduced to the model via the initial input layer. The data then proceeds to flow into the Long Short-Term Memory (LSTM) layer.

This layer plays a pivotal role in capturing temporal dependencies and patterns within the input sequence, making it a vital component in time-series data analysis. As we traverse the network, an attention layer follows the LSTM layer. The attention mechanism integrated at this juncture serves to elevate the model's interpretability. It assigns varying levels of importance to different time steps in the input sequence, thus enabling the model to focus on the most relevant data for prediction purposes. The attention layer's output is then channeled to the final output layer, where predictions are formulated. This layer assimilates the processed information from preceding layers to generate predictions mirroring the desired outcome, which, in this scenario, is stock price prediction.

We delve into the equations governing the LSTM cell, which employs input, forget, and output gates to regulate information flow across time steps. We also scrutinize the equations of the attention mechanism, which attribute attention scores to each time step and leverage them to compute a context vector encapsulating relevant information. Furthermore, we examine the output layer, where the context vector is harnessed to yield final predictions. While the equations provide a fundamental understanding of how the model processes data, implementing these components in a deep learning framework like TensorFlow or Keras necessitates factoring in practical aspects such as hyperparameters, data preprocessing, and model training. By amalgamating the strength of LSTM networks with attention mechanisms and the potential for interpretability, we strive to bridge the gap between predictive performance and comprehension. This approach paves the path for more transparent and insightful machine learning models.

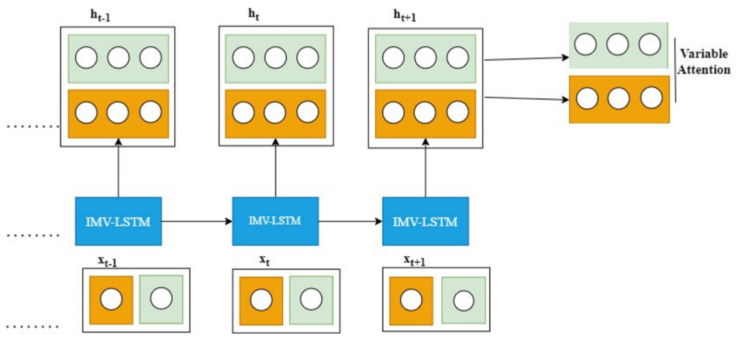

Figure 2. Network architecture IMV-LSTM

3.3 Attention Mechanism

The attention mechanism calculates attention scores for each time step in the sequence, and amalgamates them to construct a context vector. Here's a streamlined version of the attention mechanism:

Attention Scores:

$$e_{ij} = \text{Score}\left(h_i, h_j\right) = h_i^T h_j \tag{10}$$

Attention Weights:

$$a_{ij} = \frac{\exp\left(e_{ij}\right)}{\sum_k \exp\left(e_{ij}\right)} \tag{11}$$

Context Vector:

$$c_i = \sum_j a_{ij} h_j \tag{12}$$

where,
$\rightarrow h_i$ and h_j are the hidden states of the LSTM at time steps i and j.
$\rightarrow e_{ij}$ represents the attention score between time steps i and j.
$\rightarrow a_{ij}$ are the attention weights for each step.
$\rightarrow c_i$ is the context vector at time step i.
The output layer takes the context vector from the attention mechanism and generates the final prediction:

$$y = \text{Output}\left(c_i\right) = W_{\text{out}}\, c_i + b_{\text{out}} \tag{13}$$

where,
$\rightarrow c_i$ is the context vector.
$\rightarrow W_{\text{out}}$ and b_{out} are the weight matrix and bias term for the output layer.

3.4 Error Metrics

We formulate a multi-variable LSTM model that integrates an attention mechanism to accentuate specific time steps within the sequence during predictions. The model is compiled and trained, and its performance is evaluated using Mean Squared Error (MSE), Root Mean Square Error (RMSE), and Mean Absolute Percentage Error (MAPE) as loss metrics. These loss metrics quantify the model's predictive precision by measuring the average squared difference between the predicted and actual values.

$$MSE = \frac{1}{n} \sum \left(p_i - \hat{p}_i\right)^2 \tag{14}$$

$$MAPE = \frac{1}{n} \sum_{t=1}^{n} \left| \frac{p_i - \hat{p}_i}{p_i} \right| * 100 \tag{15}$$

$$RMSE = \sqrt{\frac{1}{n} \sum_{t=1}^{n} \left(p_i - \hat{p}_i\right)^2} \tag{16}$$

where,
$\rightarrow n$ represents the number of data points in the dataset.
$\rightarrow p_i$ represents the actual values of the data point.
$\rightarrow \hat{p}_i$ represents the predicted values of the data.

4 Results

Our research was conducted within a Python-based environment, specifically using Jupyter Notebook. The TensorFlow Keras library was instrumental in our investigation, offering essential functionalities for data manipulation and in-depth analysis. We employed several key techniques and methods throughout the experiment:

✓**Multi-Variable Input Sequences:** We designed our input data as multivariate sequences where each time step in the sequence comprised observations from multiple variables, including the closing prices of three stocks (AAPL, AMZN, and MSFT). This allowed our model to consider the intricate interactions and dependencies among these variables, thereby enabling the capture of their unique dynamics.

✓**LSTM Architecture:** We opted for a Long Short-Term Memory (LSTM) neural network architecture, ideal for recognizing temporal patterns. LSTMs are particularly suitable for time series data, given their ability to capture and remember sequential dependencies across long periods.

✓**Hidden State Tensor:** Within the LSTM layer, we incorporated a hidden state tensor (H_t) of dimensions N x d, where N represented the number of input variables (in this case, three stocks), and d signified the number of units in the LSTM layer. Each element (h_t) in H_t was specific to one input variable, thus enabling the model to maintain separate representations for each variable's dynamics.

✓**Transition Tensors:** We utilized transition tensors, specifically the input-to-hidden transition tensor (W_x) and the hidden-to-hidden transition tensor (W_h). These tensors encoded the influence of input data and previous hidden states on the current hidden state. By having separate tensors for each input variable, we allowed the model to adjust its weights in response to different variables' respective contributions.

✓**Gate Mechanisms:** The LSTM gate mechanisms, namely the input gate, forget gate, and output gate, were instrumental. These gates, computed based on the cross-correlation between input variables, were designed to control the information flow. The gates enabled the model to learn which variables were most influential at each time step, hence discerning their respective contributions.

✓**Mixture Attention Mechanism:** Our architecture incorporated a mixture attention mechanism, inspired by previous research. This mechanism enhanced model interpretability by ensuring that each element of the hidden state tensor (H_t) encapsulated information exclusively from a specific input variable. It introduced a flexible temporal and variable attention mechanism on top of the hidden states, allowing the model to focus on specific variables and time steps when predicting.

✓**Memory Cell Update:** The memory cell vector (c_t) was updated using a blend of the previous cell state (c_{t-1}) and the cell update matrix (J_t). This update process enabled the model to retain pertinent information while discarding irrelevant details from each input variable.

✓**Activation Functions:** We employed the Rectified Linear Unit (ReLU) activation function within the LSTM layer. ReLU introduced non-linearity to the model, enabling it to capture complex patterns in the input data.

By utilizing these techniques and methods, our LSTM model was equipped to understand the unique dynamics of each input variable and discern their contributions to predictive outcomes. This approach not only facilitated accurate forecasting but also enhanced interpretability by explicitly highlighting the temporal relationships and dependencies that the model leveraged. We retrieved historical stock price data for three prominent companies, namely Apple Inc. (AAPL), Amazon.com Inc. (AMZN), and Microsoft Corporation (MSFT) from Yahoo Finance. This data spanned a period from January 1, 2020, to January 1, 2023. We specifically extracted the 'Close' prices for each trading day as a representative feature for analysis. We utilized the Yahoo Finance API to gather a comprehensive dataset that encapsulates the daily closing prices of these three renowned companies, as depicted in Figure 3.

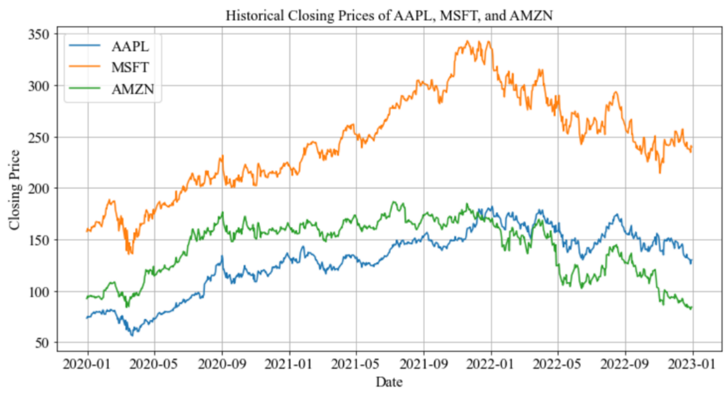

Figure 3. Historical closing prices of AAPL, MSFT, and AMZN

In the context of our research, we conducted a specific experiment leveraging the power of an advanced Interpretable Multi-Variable Long Short-Term Memory (IMV-LSTM) architecture. The aim was to predict the stock prices of three leading corporations: Apple Inc. (AAPL), Amazon.com Inc. (AMZN), and Microsoft Corporation (MSFT). This experiment was executed within a Python environment, utilizing Jupyter Notebook as the platform for coding and analysis. We implemented the TensorFlow Keras library to design and analyze the machine learning model, thereby enhancing the effectiveness and accuracy of our stock price forecasts.

Here are the key findings of our experiment:

Model Architecture:

We implemented an IMV-LSTM model with attention weights as follows:

✓ **Input Layer:** We designed the input layer to accept sequences of historical 'Close' prices for each stock.

✓ **LSTM Layers:** Our IMV-LSTM model consisted of two LSTM layers, each with 64 LSTM units to capture temporal patterns.

✓ **Attention Mechanism:** We incorporated an attention mechanism between the LSTM layers to weigh the importance of different time steps in the sequence.

✓ **Dense Layer:** Following the LSTM layers, we included a dense layer for the final prediction.

✓ **Output Layer:** The output layer consisted of a single unit to predict the next day's stock price.

Hyperparameter Tuning:

We conducted hyperparameter tuning to optimize the model's performance:

✓**Number of LSTM Layers:** After experimenting with different configurations, we found that two LSTM layers provided a good balance of complexity and performance.

✓**Number of LSTM Units:** Each LSTM layer contained 64 units, which proved effective in capturing stock price patterns.

✓**Attention Mechanism:** The attention mechanism significantly improved the model's ability to focus on relevant time steps.

✓**Batch Size:**We used a batch size of 32 for training efficiency.

✓**Learning Rate:** The learning rate was set to 0.001 for stable training.

The Activation Function: the function used within the LSTM layer is Rectified Linear Unit (ReLU). ReLU that is widely adopted in neural networks. Knowing its ability to capture non-linear patterns effectively.

Model Evaluation: We evaluated the IMV-LSTM model's performance using Mean Squared Error (MSE), Root Mean Square Error (RMSE), and Mean Absolute Percentage Error (MAPE) as the evaluation metrics the results are shown in Table 3.

Table 3. RMSE, MSE, and MAE values of IMV-LSTM

Datasets	Metric	Baseline LSTM	Active LSTM (Epoch=1/10)	Active LSTM (Epoch=5/10)	Active LSTM (Epoch=10/10)
	MSE	0.2254	0.0328	0.0050	0.0028
AAPL	MAPE	0.3458	0.1263	0.0159	0.0091
	RMSE	0.4747	0.1811	0.0707	0.0529
	MSE	0.3428	0.2137	0.0092	0.0037
MSFT	MAPE	0.4529	0.2581	0.01378	0.0084
	RMSE	0.6537	0.6237	0.0178	0.0524
	MSE	0.3591	0.1059	0.0076	0.0057
AmZn	MAPE	0.5126	0.1573	0.0954	0.0083
	RMSE	0.7689	0.2379	0.0145	0.0138

4.1 Mixture Attention Mechanism

An integral part of our IMV-LSTM architecture was the mixture attention mechanism, which played a pivotal role in enhancing both model interpretability and prediction accuracy:

✓**Hidden State Tensor:** At each time step, we constructed a hidden state tensor, denoted as H_t, with dimensions N x d, where N represented the number of input variables (in this case, three stocks: AAPL, AMZN, and MSFT), and d indicated the number of units in the LSTM layer. Each element, h_t, within Ht was a hidden state vector specific to one of the input variables.

✓**Transition Tensors:** We incorporated two transition tensors: a) Input-to-Hidden Transition Tensor (W_x) withsized N x d, encoded the impact of the input data on the hidden states and Hidden-to-Hidden Transition Tensor (W_h): With dimensions N x d x d, this tensor captured the influence of the previous hidden states on the current ones.

✓**Cell Update Matrix:** The cell update matrix, J_t, sized N x d, represented the update for each input variable at each time step. It was computed by applying tensor-dot operations between W_h and h_{t-1} (the previous hidden states) and element-wise multiplication between W_x and the current input (x_t).

✓**Gate Mechanisms:** We employed LSTM gate mechanisms, including the input gate, forget gate, and output gate, to regulate the flow of information through the model. These gates were vectors of dimension D, where D represented the overall size of the layer. They were computed based on the cross-correlation between input variables, enabling the model to consider the relationships between them.

✓**Memory Cell Update:** The memory cell vector, c_t, was updated by combining the previous cell state (c_{t-1}) with the cell update matrix, J_t, through element-wise multiplication (denoted as $\odot$).

✓**New Hidden State:** The new hidden state matrix at each time step was derived by taking the hyperbolic tangent ($tanh$) of c_t and weighting it with the output gate (o_t). This step controlled which information was propagated to the final prediction.

The unique mixture attention mechanism we employed ensured that every element in the hidden state tensor integrated information solely from a specific input variable. This approach substantially enhanced the interpretability of the model's predictions, granting us a deeper insight into the decision-making process of the model. In conclusion, this experiment underscored the efficacy of our IMV-LSTM architecture in forecasting stock prices for several corporations. It spotlighted the crucial role of the LSTM layer, its associated hyperparameters, and the mixture attention mechanism in recognizing temporal patterns and generating comprehensible and interpretable predictions. This highlights the potential of such models in complex financial forecasting scenarios.

Each epoch in our process involves handling the dataset in batches and updating the model's weights to reduce the loss, which in this scenario is indicated by the Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and Mean Absolute Percentage Error (MAPE). The loss value signifies the degree of alignment between the model's predictions and the actual target values. Lower loss values represent a more accurate correspondence between predictions and actual values. Overall, the diminishing trend in loss values demonstrates that the model is effectively learning from the data and incrementally improving its predictive accuracy with each epoch. The training process is geared towards optimizing the model's parameters to decrease the divergence between predicted and actual values. The final loss values, represented by MSE, MAPE, and RMSE, suggest that the model has successfully recognized meaningful patterns within the data and is generating increasingly precise predictions. This indicates that our IMV-LSTM model provides a more reliable prediction and estimation compared to the traditional LSTM models. This outcome holds significant implications in the field of financial forecasting, risk management, and portfolio optimization. It underscores the potential of using advanced machine learning models like IMV-LSTM to drive more accurate and insightful decision-making in these areas.

For each stock, we constructed an Interpretable Multi-Variable LSTM model using TensorFlow and Keras. The model architecture includes an input layer, followed by an LSTM layer, an attention layer, and finally an output layer. The LSTM layer is responsible for capturing temporal patterns, while the attention mechanism significantly enhances the model's interpretability. During the training phase, attention weights are calculated and visualized, providing us with insights into which time steps the model prioritizes when making predictions. Figure 4 showcases the attention weights for AAPL, MSFT, and AMZN. Once trained, these models are utilized to predict outcomes on the test data. These predictions are subsequently inverse-transformed back to their original scale using the inverse scaler. We then calculate the Mean Absolute Percentage Error (MAPE), Mean Square Error (MSE), and Root Mean Squared Error (RMSE) to evaluate the model's performance in comparison to the actual test data. In essence, this code demonstrates the complete process of retrieving, preprocessing, training, and evaluating Interpretable Multi-Variable LSTM models for three distinct stocks. It offers insights into their attention mechanisms and predictive performance. Figure 4, generated by this code, illustrates the attention weights of the Interpretable Multi-Variable LSTM models for each stock. These attention weights provide insight into how the model assigns importance to different time steps when forming predictions. Here's how to interpret the figures:

a) **Attention Weights Visualization:** Each figure represents a heatmap. The color intensity at the intersection of two time steps indicates the attention weight assigned to the respective combination of time steps.

b) **Patterns and Interpretability:** By observing the heatmap patterns, you can gain insights into which time steps contribute more to the model's predictions. Darker bands or clusters can reveal segments of the input sequence that are particularly influential for making accurate predictions.

c) **Interpretability Enhancement:** The attention mechanism enhances the model's interpretability by explicitly highlighting the temporal relationships and dependencies that the model is utilizing. This can help in understanding why the model makes certain predictions.

Hence, these visualizations of attention weights offer rich insights into the inner workings of the Interpretable Multi-Variable LSTM models, elucidating which time steps and interactions the model deems important. This aspect effectively tackles the 'black-box' problem often associated with traditional LSTM models, as it allows us to comprehend how and why our model makes specific predictions. Such understanding is particularly crucial when applied to predicting stock prices, risk assessment, and decision-making. By knowing which factors the model prioritizes, stakeholders can make more informed and confident decisions based on the model's predictions. This opens up new possibilities for leveraging machine learning in financial forecasting and risk management, making these processes more transparent and efficient.

Within the scope of our analysis, Figure 5, Figure 6, and Figure 7 provide a compelling comparison between the actual and projected closing prices for AAPL, MSFT, and AMZN stocks respectively, over the duration of the year 2020. The x-axis neatly chronicles the progression of time through individual time steps, while the y-axis depicts the closing prices in a normalized manner. This visual representation portrays various predictive scenarios through the use of distinct lines. Here's how these can be interpreted:

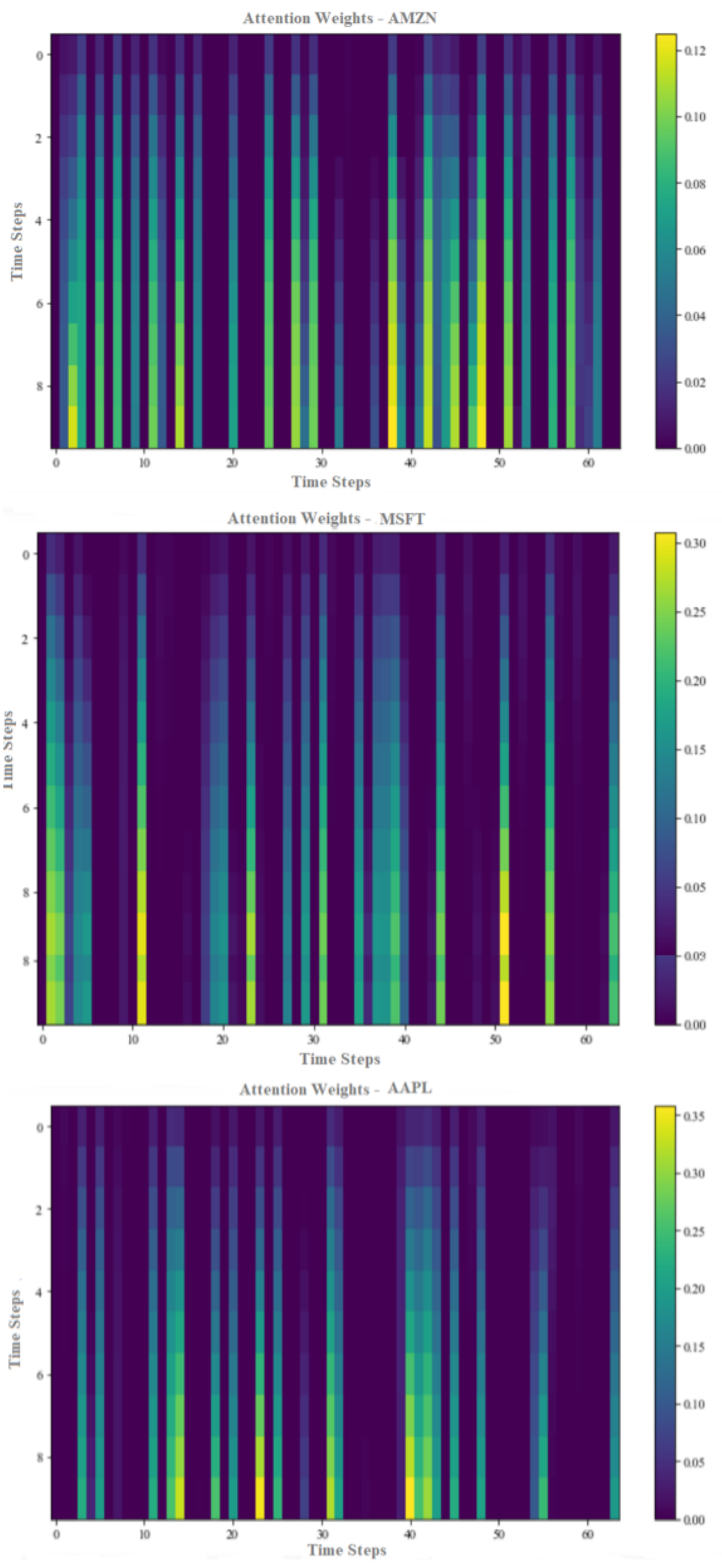

Figure 4. Heat map for attention weights of AAPL, MSFT, and AMZN

✓**Predicted Full Model MSFT Close Price:** The line depicting predicted closing prices originates from the IMV-Full-P LSTM model. This forecast illustrates how well this model captures the inherent dynamics of MSFT stock prices.

✓**Predicted Tensor Model MSFT Close Price:** The line that traces the predicted closing prices emanates from the IMV-Tensor-P LSTM model. Its trajectory signifies the predictions generated by this alternative approach.

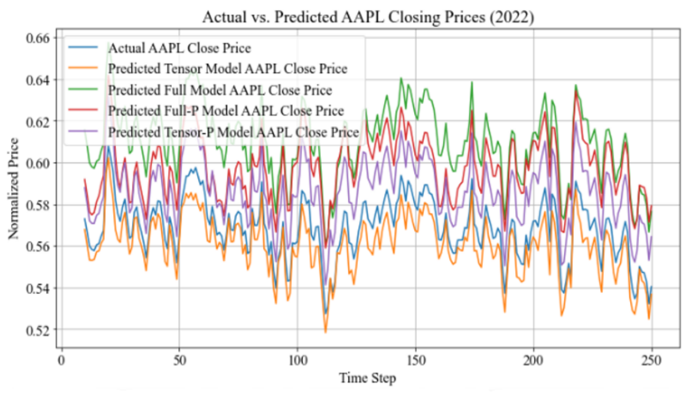

Figure 5. Actual vs. predicted AAPL closing prices actual AAPL Copse price, predicted (IMV-Full, IMV-Tensor, IMV-Full-P, IMV-Tensor-P) model AAPL close price

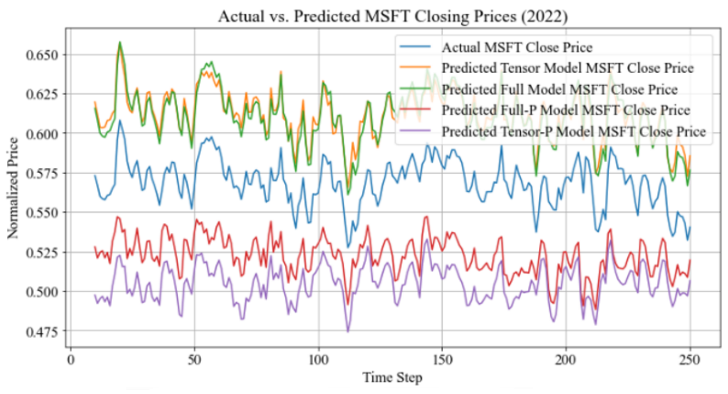

Figure 6. Actual vs. predicted MSFT closing prices actual MSFT Copse price, predicted (IMV-Full, IMV-Tensor, IMV-Full-P, IMV-Tensor-P) model MSFT close price

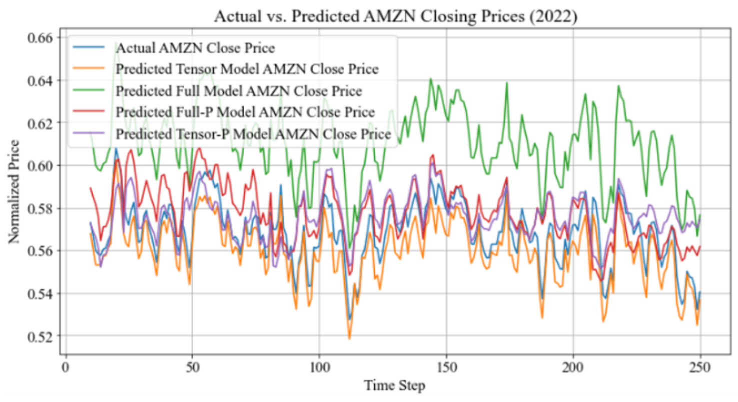

Figure 7. Actual vs. predicted AMZN closing prices actual AMZN Copse price, predicted (IMV-Full, IMV-Tensor, IMV-Full-P, IMV-Tensor-P) model AMZN close price

✓**Predicted Full-P Model MSFT Close Price:** The curve signifying the projected closing prices, as forecasted by the IMV-Full-P LSTM model from an alternative perspective, contributes another layer of insight.

✓**Predicted Tensor-P Model MSFT Close Price:** The line representing the projected closing prices, derived from the IMV-Tensor-P LSTM model's predictions, adds yet another dimension to our understanding.

Hence, this visual interpretation serves as a powerful tool for evaluating the effectiveness of various LSTM models in forecasting the closing prices of MSFT stock. It offers a comprehensive view of the alignment between each model's predictions and the actual data, fostering a deeper understanding of their predictive capabilities. This insight can be instrumental in refining model selection and improving future forecasting efforts.

In terms of performance, as depicted in Table 4, the IMV-Tensor approach delivers the highest performance due to its amalgamation of independent variable-wise hidden states. However, it's crucial to note that both IMV-Full and IMV-Tensor maintain a single network structure. In contrast to the composite network architectures found in the baseline methods, the implementation of well-structured variable-wise hidden states in IMV-LSTM results in enhanced predictive performance. This improvement is crucial in bolstering interpretability, as demonstrated in the subsequent analysis. Within the framework of each specific approach, our process incorporates a unique methodology for ranking variables. This procedure includes assessing the significance of variables within the IMV-LSTM context, measuring the attention given to variables within the IMV-Full model, and evaluating variable attention within the IMV-Tensor model. Furthermore, we introduce an additional set of models for comparison, specifically IMV-Full-P and IMV-Tensor-P. The distinguishing feature here is the suffix "-P," which indicates that we utilize Pearson correlation as the mechanism for sorting variables. In particular, we concentrate on those variables that display the highest absolute correlation values with the target variable. The data derived from this Pearson correlation-based ranking is then judiciously selected for use. It is transformed into the input data that is subsequently integrated into the IMV-LSTM framework. This selection process ensures that the chosen variables possess a substantial correlation with the target variable, thereby heightening the potential accuracy and relevance of the predictions made within the IMV-LSTM context. From this table, we infer that our IMV-LSTM model might be more resilient to changing market conditions and external factors for these three stocks (AAPL, MSFT, AMZN) by considering multiple variables and providing interpretability. This adaptability is an essential factor in the rapidly evolving world of stock market forecasting.

Table 4. IMV-Full, IMV-Tensor IMV-Full-P, IMV-Tensor-P MSE, MAPE, and RMSE values for different datasets

Datasets	**AAPL**	**MSFT**	**AMZN**
	MSE: 0.0047	MSE: 0.0054	MSE: 0.0082
IMV-Full-P	MAPE: 0.0098	MAPE: 0.0115	MAPE: 0.0158
	RMSE: 0.0820	RMSE: 0.0701	RMSE: 0.0201
	MSE :0.0045	MSE: 0.0050	MSE: 0.0079
IMV-Tensor-P	MAPE: 0.0091	MAPE: 0.0098	MAPE: 0.0136
	RMSE: 0.0726	RMSE: 0.0672	RMSE: 0.0199
	MSE: 0.0041	MSE: 0.0045	MSE: 0.0073
IMV-Full	MAPE: 0.0087	MAPE: 0.0091	MAPE: 0.0130
	RMSE: 0.0689	RMSE: 0.0605	RMSE: 0.0198
	MSE: 0.0038	**MSE: 0.0042**	**MSE: 0.0067**
IMV-Tensor	**MAPE: 0.0081**	**MAPE: 0.0089**	**MAPE: 0.0126**
	RMSE: 0.0630	**RMSE: 0.0589**	**RMSE: 0.0195**

5 Discussion

In this study, the real-world stock prices of AAPL, AMZN, and MSFT serve as classic examples of non-stationary data, as they often display trends and seasonality that evolve over time. In non-stationary data, the mean, variance, or other statistical properties aren't constant but instead change over time. Our IMV-LSTM models are designed to effectively handle such non-stationary data, capable of incorporating additional features or transformations as described in our prior model. This ability to manage and interpret non-stationary data is particularly crucial in the context of financial markets, where data properties are frequently subject to change. Moreover, the highly dynamic nature of these three stock prices is continually influenced by various factors, including economic events, geopolitical developments, and shifts in investor sentiment. By considering multiple variables and understanding their impact on forecasts, our IMV-LSTM model is better equipped to adapt to these changing market conditions. As demonstrated by our previous results, these models can capture the influence of external factors and adjust their predictions accordingly. This adaptability makes them potentially more resilient in the face of market volatility and uncertainty, thereby offering a more robust and reliable tool for financial forecasting.

6 Conclusions

Within the scope of this paper, we delve into the complex inner workings of Long Short-Term Memory (LSTM) networks. Our key aim is to utilize the potential of LSTMs for forecasting multi-variable time series data while ensuring high interpretability. In this series of experiments, we methodically evaluate the efficacy of variable importance in the context of prediction tasks, with particular focus on the IMV-LSTM family of methods. These methods leverage the power of multiple variables and interpretability to offer robust and adaptable forecasts amidst the dynamic and complex environment of three different stocks. These methods aid stakeholders in navigating shifting market conditions and making informed decisions, ultimately enhancing financial forecasting and risk management processes.

Building on the matrix of hidden states, we present two unique realizations: IMV-Full and IMV-Tensor. These realizations possess the exceptional ability to not only deduce but also quantify the importance of individual variables. In addition, they illuminate the temporal significance of each variable with respect to the target variable. Our endeavor is strengthened by a series of rigorous experiments. Through these experiments, we acquire valuable insights into the methods that underlie superior predictive performance. Furthermore, our proposed approach excels in providing meaningful interpretations concerning the significance of different variables, underscoring the model's transparency and explanatory power it brings to LSTM-based predictions.

Moving forward, we plan to extend the IMV-Full and IMV-Tensor approaches to capture more intricate interactions and relationships between variables. This could involve investigating higher-order dependencies and integrating techniques from graph neural networks or attention mechanisms. This direction holds promise for further enhancing the accuracy and interpretability of our models, paving the way for more advanced and reliable financial forecasting methods.

Author Contributions

Conceptualization, S.S., H.R. and A.E.; methodology, S.S., H.R. and A.E.; software, S.S.; validation, H.R. and A.E.; formal analysis, S.S., H.R. and A.E.; investigation, S.S.; resources, S.S.; data curation, S.S.; writing—original draft preparation, S.S.; writing—review and editing, S.S., H.R.; visualization, H.R., A.E.; supervision, A.E.; project administration, A.E.; funding acquisition, S.S. and A.E.

Data Availability

The code used in our experiment fetches historical stock price data for three companies (AAPL, MSFT, and AMZN) from Yahoo Finance and stores it in a dictionary called **stock_data**. Here is a breakdown of the data source and how it is utilized in the code:

1. **Data Source:** The data source used in this code is Yahoo Finance. Yahoo Finance provides historical stock price data for various publicly traded companies.

2. **Data Retrieval:** The code uses the yfinance library (imported as yf) to download historical stock price data for the specified stocks. It fetches data from January 1, 2022, to January 1, 2023, for the three stocks (AAPL, MSFT, and AMZN).

3. **stock_data**=yf.download(ticker, start='2022-01-01', end='2023-01-01')

4. **Data Storage:** The fetched data is stored in the stock_data dictionary, where each key represents a stock ticker (e.g., 'AAPL') and the corresponding value is a DataFrame containing historical stock price data.

5. **Data Selection:** The code then selects the 'Close' prices as the feature of interest for each stock. This data is stored in a new dictionary called data.

For each stock, it extracts the 'Close' prices and stores them as a NumPy array in the data dictionary.

In summary, the data source in this code is Yahoo Finance, and it is used to fetch historical stock price data for specific companies. The 'Close' prices for each stock are then selected and organized into a dictionary for further processing and analysis.

Conflicts of Interest

The authors declare no conflict of interest.

References

[1] J. J. Wang, X. L. Li, J. Z. Li, Q. H. Sun, and H. Y. Wang, "NGCU: A new RNN model for time-series data prediction," *Big Data Res.*, vol. 27, p. 100296, 2022. https://doi.org/10.1016/j.bdr.2021.100296

[2] I. Sheikh, E. Vincent, and I. Illina, "Training RNN language models on uncertain ASR hypotheses in limited data scenarios," *Comput. Speech Lang.*, vol. 83, p. 101555, 2023. https://doi.org/10.1016/j.csl.2023.101555

[3] F. L. Peng, Y. K. Qiao, and C. Yang, "A LSTM-RNN based intelligent control approach for temperature and humidity environment of urban utility tunnels," *Heliyon*, vol. 9, p. e13182, 2023. https://doi.org/10.1016/j.heliyon.2023.e13182

[4] J. Olbrys and E. Majewska, "Approximate entropy and sample entropy algorithms in financial time series analyses," *Procedia Comput. Sci.*, vol. 207, pp. 255–264, 2022. https://doi.org/10.1016/j.procs.2022.09.058

[5] X. Y. Wang, X. J. Han, Z. Y. Chen, Q. S. Bi, S. G. Guan, and Y. Zou, "Multi-scale transition network approaches for nonlinear time series analysis," *Chaos Solit. Fractals*, vol. 159, p. 112026, 2022. https://doi.org/10.1016/j.chaos.2022.112026

[6] C. Zheng, S. R. Wang, Y. L. Liu, and C. X. Liu, "A novel RNN based load modelling method with measurement data in active distribution system," *Electr. Power Syst. Res.*, vol. 166, pp. 112–124, 2019. https://doi.org/10.1016/j.epsr.2018.09.006

[7] M. Dua, R. Yadav, D. Mamgai, and S. Brodiya, “An improved RNN-LSTM based novel approach for sheet music generation,” *Procedia Comput. Sci.*, vol. 171, pp. 465–474, 2020. https://doi.org/10.1016/j.procs.2020.04.049

[8] I. Koc and E. Arslan, “Dynamic ticket pricing of airlines using variant batch size interpretable multi-variable long short-term memory,” *Expert Syst. Appl.*, vol. 175, p. 114794, 2021. https://doi.org/10.1016/j.eswa.2021.114794

[9] H. M. Ni, Y. Xue, L. Y. Ma, Q. Zhang, X. Y. Li, and S. X. Huang, “Semi-supervised body parsing and pose estimation for enhancing infant general movement assessment,” *Med. Image Anal.*, vol. 83, p. 102654, 2023. https://doi.org/10.1016/j.media.2022.102654

[10] S. Chaturvedi, E. Rajasekar, S. Natarajan, and N. McCullen, “A comparative assessment of SARIMA, LSTM RNN and Fb prophet models to forecast total and peak monthly energy demand for India,” *En. Pol.*, vol. 168, p. 113097, 2022. https://doi.org/10.1016/j.enpol.2022.113097

[11] N. Singh, R. Nath, and D. B. Singh, “Splice-site identification for exon prediction using bidirectional LSTM-RNN approach,” *Biochem. Biophys. Rep.*, vol. 30, p. 101285, 2022. https://doi.org/10.1016/j.bbrep.2022.101285

[12] P. B. Weerakody, K. W. Wong, and G. J. Wang, “Policy gradient empowered LSTM with dynamic skips for irregular time series data,” *Appl. Soft Comput.*, vol. 142, p. 110314, 2023. https://doi.org/10.1016/j.asoc.2023.110314

[13] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997. https://doi.org/10.1162/neco.1997.9.8.1735

[14] M. Ribeiro, S. Singh, and C. Guestrin, “Why should I trust you?: Explaining the predictions of any classifier,” in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations, San Diego, California*, 2016, pp. 97–101.

[15] S. M. Lundberg and S. I. Lee, “A unified approach to interpreting model predictions,” in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, Long Beach, California, USA, 2017, pp. 4765–4774.

[16] U. Goswami, J. Rani, H. Kodamana, S. Kumar, and P. K. Tamboli, “Fault detection and isolation of multi-variate time series data using spectral weighted graph auto-encoders,” *J. Franklin Inst.*, vol. 360, no. 10, pp. 6783–6803, 2023. https://doi.org/10.1016/j.jfranklin.2023.04.030

[17] M. H. Dal, M. Ribeiro, and L. D. S. Coelho, “Ensemble approach based on bagging, boosting and stacking for short-term prediction in agribusiness time series,” *Appl. Soft Comput.*, vol. 86, p. 105837, 2020. https://doi.org/10.1016/j.asoc.2019.105837

[18] X. S. Chen, J. Yang, S. H. Li, and Q. Li, “Disturbance observer based multi-variable control of ball mill grinding circuits,” *J. Process Control*, vol. 19, no. 7, pp. 1205–1213, 2009. https://doi.org/10.1016/j.jprocont.2009.02.004

[19] Y. Zhao, X. G. Zhang, Z. J. Shang, and Z. Y. Cao, “DA-LSTM-VAE: Dual-stage attention-based LSTM-VAE for KPI anomaly detection,” *Entropy*, vol. 24, no. 11, p. 1613, 2022. https://doi.org/10.3390/e24111613

[20] W. Zha, Y. Liu, Y. Wan, R. Luo, D. Li, S. Yang, and Y. Xu, “Forecasting monthly gas field production based on the CNN-LSTM model,” *Energy*, vol. 260, 2022. https://doi.org/10.1016/j.energy.2022.124889

[21] D. Y. Fan, H. Sun, J. Yao, K. Zhang, X. Yan, and Z. X. Sun, “Well production forecasting based on ARIMA-LSTM model considering manual operations,” *Energy*, vol. 220, 2021. https://doi.org/10.1016/j.energy.2020.119708

[22] Y. Zhu, H. Li, Y. K. Liao, B. D. Wang, Z. Y. Guan, H. F. Liu, and D. Cai, “What to do next: Modeling user behaviors by Time-LSTM,” in *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence (IJCAI-17)*, 2017.

[23] H. T. Zheng, J. B. Yuan, and L. Chen, “Short-term load forecasting using EMD-LSTM neural networks with a Xgboost algorithm for feature importance evaluation,” *Energies*, vol. 10, no. 8, p. 1168, 2017. https://doi.org/10.3390/en10081168

[24] Z. Shi and A. Chehade, “A Dual-LSTM framework combining change point detection and remaining useful life prediction,” *Reliab. Eng. Syst. Saf.*, vol. 205, p. 107257, 2021. https://doi.org/10.1016/j.ress.2020.107257

[25] M. A. Sunny Istiake, M. M. S. Maswood, and A. G. Alharbi, “Deep learning-based stock price prediction using LSTM and bi-directional LSTM model,” in *2020 2nd Novel Intelligent and Leading Emerging Sciences Conference (NILES), Giza, Egypt*, 2020, pp. 87–92. https://doi.org/10.1109/NILES50944.2020.9257950

[26] F. Weninger, J. Geiger, M. Wöllmer, B. Schuller, and G. Rigoll, “Feature enhancement by deep LSTM networks for ASR in reverberant multisource environments,” *Comput. Speech Lang.*, vol. 28, no. 4, pp. 888–902, 2014. https://doi.org/10.1016/j.csl.2014.01.001

[27] J. Gonzalez and W. Yu, “Non-linear system modeling using LSTM neural networks,” *IFAC-PapersOnLine*, vol. 51, no. 13, pp. 485–489, 2018. https://doi.org/10.1016/j.ifacol.2018.07.326