Curvature-Informed Attention Mechanism for Long Short-Term Memory Networks

Lynda Ayachi
Orange Innovation Tunisia, Sofrecom, Tunis, Tunisia

Keywords: Time Series Forecasting, LSTM, Attention Mechanism, Encoder-Decoder Model, Interpretability, Explainble AI.

Abstract: Time series forecasting is a crucial task across diverse domains, and recent research focuses on refining model architectures for enhanced predictive capabilities. In this paper, we introduce a novel approach by integrating curvature measures into an attention mechanism alongside Long Short-Term Memory (LSTM) networks. The objective is to improve the interpretability and overall performance of time series forecasting models. The proposed Curvature-Informed Attention Mechanism (CIAM) enhances learning by personalizing the weight attribution within the attention mechanism. Through comprehensive experimental evaluations on real-world datasets, we demonstrate the efficacy of our approach, showcasing competitive forecasting accuracy compared to traditional LSTM models.

1 INTRODUCTION

In the era of rapid technological advancements, marked by the proliferation of Big Data and the Internet of Things (IoT), diverse industries such as industrial engineering, financial technology, and natural sciences have witnessed an unprecedented accumulation of extensive datasets. Among these, time series data stands out as a critical component, serving as a fundamental basis for forecasting future events based on their temporal sequence. Whether in its univariate or multivariate form, time series data presents inherent challenges. The presence of noise and missing values within these temporal datasets introduces complexities, undermining the fidelity of information and, consequently, the precision of predictive models. Additionally, the effectiveness of forecasting models is contingent upon the availability of robust data, highlighting the need for robust methodologies to address the intricacies posed by noisy, incomplete, or insufficient time series data. Various methodologies exist for the prediction of time series data (Nakkach et al., 2023) (Nakkach et al., 2022), categorized broadly into two classes: traditional forecasting methods and machine learning-based forecasting approaches. Traditional forecasting methods, including autoregressive (AR) (McLeod and Li, 1983), moving average (MA) (Torres et al., 2005), autoregressive moving average (ARMA) (ByoungSeon, 2012), and differential autoregressive moving average (ARIMA) (Box, 2015) models, exhibit certain limitations in their predictive capabilities. These traditional approaches are often constrained by strict data requirements and are more adept at handling smooth data, whereas real-life time series data tend to be inherently unstable. Consequently, the accuracy of traditional forecasting methods is frequently suboptimal for forecasting dynamic real-world data.

In recent times, the ascent of artificial intelligence has propelled neural networks to the forefront of problem-solving. These networks, consisting of nodes adept at mastering non-linear functions, prove instrumental in converting input data into desired outputs. The iterative learning process refines the weights of these nodes, minimizing errors at the output and showcasing the adaptability and efficiency of neural networks.

Within this neural network landscape, the Recurrent Neural Network (RNN) (Yu et al., 2019) emerges as a distinctive architecture. Its nodes conduct computations and predictions at each instance, shaping the output by drawing from both historical data represented by a hidden variable and the current input at time t. The unique structure of each RNN unit encapsulates this intricate process.

The Gated Recurrent Unit (GRU) (Cho et al., 2014) represents a popular variant of the Recurrent Neural Network Unit. GRU introduces two key gates:

Ayachi, L.
Curvature-Informed Attention Mechanism for Long Short-Term Memory Networks.
DOI: 10.5220/0012463500003636
Paper published under CC license (CC BY-NC-ND 4.0)
In *Proceedings of the 16th International Conference on Agents and Artificial Intelligence (ICAART 2024) - Volume 3*, pages 1263-1269
ISBN: 978-989-758-680-4; ISSN: 2184-433X
Proceedings Copyright © 2024 by SCITEPRESS – Science and Technology Publications, Lda.

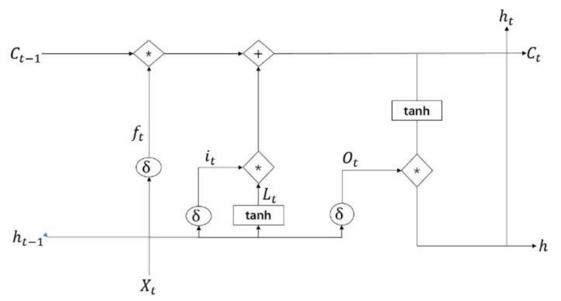

Figure 1: Long Short-Term Memory Network Architecture.

the Update Gate, responsible for time step Zt updates at time t, and the Reset Gate, determining the extent to which past information should be discarded in the current time step. This nuanced architecture provides an alternative perspective for optimizing time series forecasting models.

For non-stationary multivariate time series forecasting. (Liu and Chen, 2019) introduced a linear hybrid gate unit, MIXGU, by combining GRU and MGU models, leading to a synergistic enhancement of prediction outcomes. Another noteworthy example is the work of Dai (Dai et al., 2020), who enhanced PM2.5 concentration prediction in Beijing by integrating the empirical modal decomposition algorithm (EMD) with LSTM. This hybrid approach surpassed the predictive accuracy achieved by individual algorithms, emphasizing the importance of tailored adaptations for improved forecasting performance.

2 CURVATURE-INFORMED ATTENTION MECHANISM FOR LSTM

Time series data typically consists of pairs of values associated with specific time periods or points in time. These pairs comprise two distinct elements: the temporal component and the numerical value. The temporal element signifies a particular time period or point in time, while the numerical element may correspond to a single variable or multiple variables. When the numerical element pertains to one variable, it is termed a univariate time series. Conversely, when it involves multiple variables, it is referred to as a multivariate time series.

2.1 Long Short-Term Memory Network

The LSTM (Gers, 2002) unit, illustrated in Figure1 involves several components, each serving a distinct purpose in processing sequential data. Let X_t be the input sequence at time t, h_t be the hidden state (memory), and C_t be the cell state. The LSTM components are defined as follows: The forget gate, denoted as f_t decides what information from the cell state should be discarded. It takes the concatenation of the input X_t and the previous hidden state h_{t-1} as input and outputs values between 0 and 1 for each element in the cell state. The forget gate is defined as follows:

$$f_t = \sigma(W_f \cdot [h_{t-1}, X_t] + b_f) \tag{1}$$

where W_f is the weight matrix, b_f is the bias, and σ is the sigmoid activation function.

The input gate, denoted as i_t, updates the cell state with new information. Similar to the forget gate, it takes the concatenation of $[h_{t-1}, X_t]$ and outputs values between 0 and 1. The input gate is defined as follows:

$$i_t = \sigma(W_i \cdot [h_{t-1}, X_t] + b_i) \tag{2}$$

The cell state is updated using the forget gate, input gate, and a new candidate value $\tilde{C}_t$. The candidate value $\tilde{C}_t$ is calculated using the tanh activation function:

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, X_t] + b_C) \tag{3}$$

The updated cell state C_t is given by:

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t \tag{4}$$

In these equations, $W_f, b_f, W_i, b_i, W_C, b_C, W_o, b_o$ represent the weight matrices and biases. The sigmoid function σ and the hyperbolic tangent function tanh are used for activation.

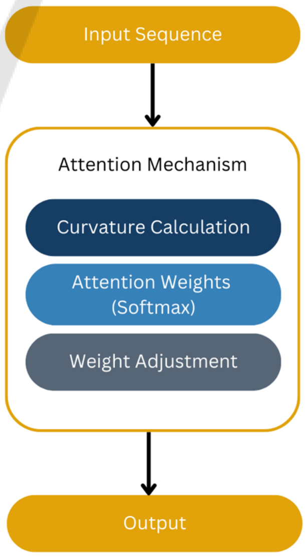

Figure 2: The Curvature-Informed Attention Mechanism.

2.2 Curvature-Informed Attention Mechanism CIAM

Attention mechanisms (Ashish et al., 2017) enhance the capabilities of deep learning models in tasks involving sequential data, such as Natural Language Processing (NLP) and time series forecasting. The core idea behind attention is to enable the model to focus selectively on specific segments of the input sequence, assigning varying levels of significance to different elements.

The Curvature-Informed Attention Mechanism (CIAM) introduces a novel approach by incorporating the intrinsic curvature measure of the original time series data, providing a quantifiable representation of variations and trends. The curvature has been employed in many approaches to recognize and charectrize curves (Ayachi et al., 2023) (Ayachi et al., 2020) (Abbasi et al., 2000). Unlike traditional attention mechanisms that rely solely on sequential patterns or external features, CIAM leverages the inherent curvature measure. Figure 2 present an overview of the approach.

Step 1: Curvature Calculation

The curvature (κ_i) for each data point (X_i) is computed by taking the second derivative of the data with respect to time:

$$\kappa_i = \frac{d^2 X_i}{dt^2} \tag{5}$$

Step 2: Attention Weights Calculation

Attention weights (α_i) are determined based on the softmax function applied to the curvature measures:

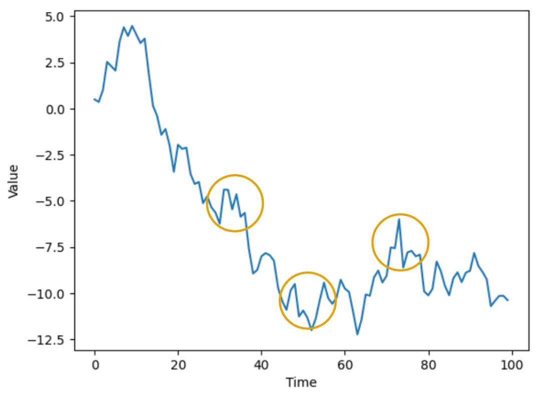

Figure 3: In Blue, a random time serie. In Orange, zones to be focused on by CIAM. Larger weights are expected to be attributed to these areas.

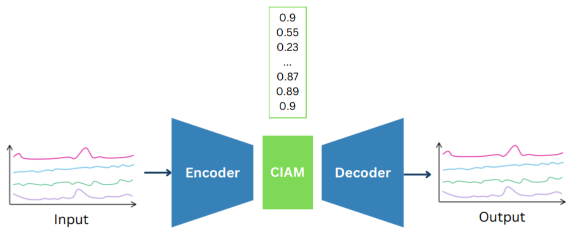

Figure 4: CIAM based encoder-decoder architecture. The input is formed by univariate or multivariate time series. The output represent the reconstructed initial series.

$$\alpha_i = \frac{\exp(\kappa_i)}{\sum_{j=1}^{N} \exp(\kappa_j)} \tag{6}$$

Step 3: Dynamic Weight Adjustment

Attention weights (α_i') are dynamically adjusted to incorporate curvature values and previous attention weights, controlled by a hyperparameter (β):

$$\alpha_i' = \beta \cdot \alpha_i + (1 - \beta) \cdot \kappa_i \tag{7}$$

Step 4: Full CIAM Process

The complete CIAM process is expressed as a set of adjusted attention weights for each data point:

$$\text{CIAM}(X_1, X_2, \ldots, X_N) = \{\alpha_1', \alpha_2', \ldots, \alpha_N'\} \tag{8}$$

2.3 Encoder-Decoder Network

The concept of the code-and-decode model (Sutskever et al., 2014) (Bahdanau et al., 2014) (Liu et al., 2019) (Junczys-Dowmunt et al., 2018) initially emerged as a solution for tackling the seq2seq problem, primarily designed for tasks like text translation or responding to input queries. As its application expanded to include time series prediction, promising results were observed. Despite its effectiveness over standalone models, the encoder-decoder model is not without limitations. Its reliance on a single link between the encoder and decoder, embodied by a fixed-length vector, poses challenges, particularly for lengthy input sequences. The fixed-length vector struggles to retain information from earlier inputs when processing subsequent ones. To address this limitation, the attention mechanism has been integrated into the encoder-decoder model, aiming to enhance its capacity to capture and utilize information from the entire sequence.

Within the Curvature-Informed Attention Mechanism (CIAM), the weights undergo dynamic adjustments contingent on the curvature measure of the original time series data, as illustrated in Figure 3

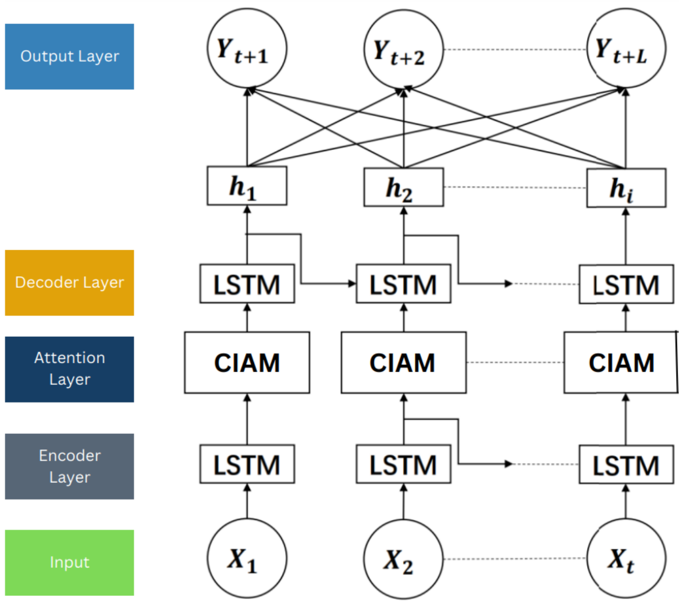

Figure 5: A detailed overview of the CIAM based architecture : The encoding is performed using LSTM, followed by an attention mechanism based on curvature. The decoding is realized based on an inverse operation to obtain the initial series. The invertibility characteristic of the curvature enhances the decoding process.

and Figure 4. This entails calculating curvature measures for each data point, encapsulating variations and trends in the sequence. The application of the softmax function to these curvature values yields attention weights. Consequently, the attention mechanism contributes to refining forecasting outcomes, particularly in scenarios characterized by random trends and seasonality.

The proposed architecture is a robust sequence-to-sequence model comprising an LSTM-based encoder, a Curvature-Informed Attention Mechanism (CIAM), and a decoder LSTM (Figure 5). The LSTM encoder captures temporal dependencies, while CIAM dynamically adjusts attention based on curvature measures, enabling the model to prioritize significant points in the sequence. The decoder LSTM reconstructs the time series using information from the encoder and CIAM, resulting in an effective and adaptive model for accurate time series prediction.

It is noteworthy that the information encoded in the curvature measure holds the potential for reconstructing the initial time series curve. This intrinsic capability adds value to the encoder-decoder architecture, where the decoder plays a crucial role in the reconstruction of data.

3 EXPERIMENTATIONS

This section contains a presentation of the evaluation metrics and used datasets. Subsequently, we delineate the dataset preprocessing methodology and expound on the pertinent parameter configurations for the experiments. The subsequent segment involves an in-depth analysis and assessment of the experimentally obtained results.

3.1 Datasets

Our experimentation involved the utilization of five publicly accessible datasets: the Beijing Air Quality dataset (Air)(27,), the Household Electricity Consumption dataset from Paris, France (Electricity) (28,), the Daily Stock dataset (Stock) (29,), the Daily Gold Price dataset (Gold) (30,), and the Chengdu PM2.5 Concentration dataset (CDPM2.5) (ByoungSeon, 2012).

The Air dataset originates from the US Embassy in Beijing, collecting weather and air pollution indices over five years (2010 to 2014) with hourly granularity. This multivariate time series dataset encompasses variables like time, PM2.5 concentration, dew point, temperature, atmospheric pressure, wind direction, wind speed, cumulative hourly snowfall, and cumulative hourly rainfall.

The Electricity dataset, also multivariate, captures household electricity consumption in Paris from De-

Table 1: Overview of Dataset Characteristics for Experimental Evaluation.

Dataset	Size	Variables
Air	8760	8
Electricity	6120	7
Stock	2426	6
Gold	2227	6
CDPM2.5	8760	9

cember 2006 to November 2010 at one-minute intervals. It includes time, active power per minute, reactive power per minute, average voltage per minute, average current intensity per minute, and specific electricity usage details.

The Stock dataset documents daily stock movements from April 26, 2010, to April 24, 2020, with daily data points. It incorporates information such as stock opening price, closing price, all-day high, all-day low, total, and code.

The Gold dataset records daily gold prices from January 1, 2014, to August 5, 2022, with daily collection intervals. It includes data like the daily closing price of gold, opening price, highest price for the day, lowest price for the day, total number of trades for the day, and daily rise and fall.

The fifth dataset pertains to PM2.5 data for Chengdu from 2010 to 2015, recorded on a daily basis. Variables cover PM2.5 concentration, dew point temperature, humidity, pressure, combined wind direction, cumulative wind speed, precipitation, and cumulative precipitation.

3.2 Data Pre-Processing

In the initial data processing stage, we transformed all five datasets into a 3D tensor of [sample, time step, feature], making their format adequate for subsequent model input. To handle outliers and missing values, mean filling was employed for the latter, while the former were directly removed.

Specifically, in the case of the AIR dataset, the non-predictive variable of wind direction was excluded, and missing values were addressed using mean filling. Likewise, the ELECTRICITY dataset saw the removal of non-compliant variables related to kitchen active energy, electric water heater, and air conditioner, with mean fill applied for any missing values. For the STOCK dataset, non-compliant coded variables were eliminated. The GOLD dataset underwent a similar process, removing variables that didn't align with experimental requirements. In the CDPM2.5 dataset, anomalous variables like PM2.5 and combined wind data were excluded, and irrelevant variables like precipitation and cumulative precipitation were also removed. This resulted in final datasets containing 7, 5, 5, 5, and 5 variables, respectively.

During experimentation, datasets were divided into training and testing sets. For AIR and CDPM2.5, the first 11 months were designated for training, with the remaining month for testing. Predictors included wind speed for AIR and humidity for CDPM2.5. In the ELECTRICITY dataset, the first 6000 entries constituted the training set, and average voltage per minute was chosen as the final predictor. For STOCK, 2376 entries were utilized for training, with the closing price as the predictor variable. In the GOLD dataset, the first 2160 entries served as the training set, and the closing price was selected as the predictor variable. Following dataset slicing, a maximum-minimum normalization process was applied to prevent data discrepancies from affecting results. In the final prediction phase, back-normalization of the data to its original values was carried out.

3.3 Experimental Parameters

Concerning the experimental configuration, the setup for each of the five datasets varied due to differences in their acquisition intervals. This discrepancy primarily manifested in the time step configurations. Given the hourly acquisition period of the AIR dataset, the time step for both AIR and CDPM2.5 was set to 24, aligning with the predicted length. Similarly, for the ELECTRICITY dataset with a collection period of 1 minute, the time step was configured at 30, matching the predicted length. In the cases of STOCK and GOLD, both with a daily collection period, the time step was set to 24, again corresponding to the predicted length.

As for the model parameters, a consistent epoch count of 50, a batch size of 128, and a hidden layer of 100 in the LSTM were employed across all experiments. The activation function utilized was the ReLU function. To mitigate overfitting, a dropout function was incorporated after each LSTM model, with a dropout parameter of 0.3. Batch normalization was applied to the data to enhance the stability of the entire neural network at each layer's intermediate output. Throughout the training process, the mean squared error (mse) loss function was chosen, and optimization utilized the Adam algorithm with a learning rate set to 0.001. Additionally, the learning rate decayed by 1e-5 per round.

Here is a table summarizing the experimental parameters:

Table 2: Multivariate time series forecasting results for five datasets.

Methods	CNN		CNN-LSTM		LSTM		Stacked-LSTM		BiLSTM		Encoder-decoder-LSTM		CIAM LSTM	
	RMSE	MAPE			RMSE	MAPE	RMSE	MAPE	RMSE	MAPE	RMSE	MAPE	RMSE	MAPE
AIR	1.282	49.85%	1.234	50.09%	1.264	52.17%	1.291	52.74%	1.530	61.83%	1.315	58.86%	1.144	48.42%
ELECTRICITY	2.792	9.54%	0.749	2.68%	0.937	3.31%	1.484	5.15%	1.675	5.79%	0.907	3.21%	0.635	1.53%
STOCK	0.305	60.30%	0.030	12.28%	0.128	30.07%	0.334	61.64%	0.139	29.56%	0.470	79.52%	0.034	11.20%
GOLD	4.878	77.47%	0.507	26.93%	1.506	54.17%	3.205	63.90%	1.238	46.19%	0.690	38.65%	0.203	19.56%
CDPM2.5	0.233	11.52%	0.219	11.11%	0.178	9.03%	0.349	17.72%	0.551	27.62%	0.170	8.39%	0.166	8.01%

Table 3: Experimental Parameters.

Parameter	**Value**
Epoch	50
Batch Size	128
Hidden Layer in LSTM	100
Activation Function	ReLU
Dropout Parameter	0.3
Learning Rate	0.001
Learning Rate Decay	1e-5

3.4 Results and Discussion

The comprehensive evaluation of various forecasting methods on five distinct datasets, namely AIR, ELECTRICITY, STOCK, GOLD, and CDPM2.5, is presented in Table 2. The performance metrics, including root mean square error (RMSE) and mean absolute percentage error (MAPE), provide insights into the effectiveness of each method.

For the AIR dataset, the CIAM LSTM outperforms other models, achieving an RMSE of 1.144 and a MAPE of 48.42%. Notably, it surpasses baseline models such as CNN, CNN-LSTM, LSTM, Stacked-LSTM, BiLSTM, and Encoder-decoder-LSTM in both metrics.

In the case of ELECTRICITY, CIAM LSTM demonstrates superior performance with an RMSE of 0.635 and a MAPE of 1.53%, outclassing alternative methods like CNN, CNN-LSTM, LSTM, Stacked-LSTM, BiLSTM, and Encoder-decoder-LSTM.

For the STOCK dataset, CIAM LSTM excels with an RMSE of 0.034 and a MAPE of 11.20%, showcasing its effectiveness compared to competing models.

In the GOLD dataset, CIAM LSTM again exhibits strong performance, achieving an RMSE of 0.203 and a MAPE of 19.56%, surpassing several other methods.

For the CDPM2.5 dataset, CIAM LSTM demonstrates competitive accuracy, with an RMSE of 0.166 and a MAPE of 8.01%, showcasing its efficacy in multivariate time series forecasting.

Overall, the results highlight the robustness of the proposed CIAM LSTM, consistently providing accurate predictions across diverse datasets. These findings emphasize the potential of the Curvature-Informed Attention Mechanism in enhancing LSTM-based forecasting models for various multivariate time series applications.

4 CONCLUSION

In conclusion, the Curvature-Informed Attention Mechanism (CIAM) stands out as an explainable and interpretable enhancement to traditional attention mechanisms in the realm of time series forecasting. By incorporating the intrinsic curvature measures of the original time series data, CIAM introduces a novel approach that surpasses conventional methods relying solely on sequential patterns or external features.

The dynamic adjustment of attention weights, in an encoder-decoder architecture, based on curvature measures empowers CIAM to discern and prioritize the significance of various points within a sequence. This adaptability is particularly valuable in scenarios characterized by random trends, where patterns defy regular trajectories. Unlike rigid attention mechanisms that may struggle to capture unpredictable variations, CIAM focuses on dynamically allocating attention, enabling the model to flexibly respond to important deviations and nuanced fluctuations in the data.

The experimental results, as showcased in the presented multivariate time series forecasting results for five diverse datasets, demonstrate the efficacy of CIAM. Notably, CIAM exhibits superior performance in comparison to established methods like CNN, LSTM, and Encoder-decoder-LSTM across various evaluation metrics such as RMSE and MAPE. This reinforces the notion that incorporating curvature measures in the attention mechanism significantly contributes to the model's predictive accuracy, especially in the presence of random trends and intricate seasonality patterns.

The invertibility characteristic of the curvature makes it a promising foundation for an encoder-decoder model. Mainly in the decoding part where the reconstruction process utilizes the this characteristic to rebuild the original time serie.

In summary, CIAM not only advances the current state-of-the-art in time series forecasting but also provides a conceptual framework for integrating intrinsic data characteristics into attention mechanisms. This novel approach represents a step forward in enhancing the predictive capabilities of machine learning models, laying the foundation for more sophisticated and adaptive forecasting methodologies.

REFERENCES

(apr. 5, 2022). beijing pm2.5 dataset. [online]. available: https://archive.ics.uci.edu/ml/datasets/beijing+pm2.5+data.

(apr. 5, 2022). daily gold price dataset. [online]. available: https://www.kaggle.com/datasets/nisargchodavadiya/daily-gold-price20152021-time-serie.

(apr. 5, 2022). daily stock dataset. [online]. available: https://www.kaggle.com/datasets/dsadads/databases.

(apr. 5, 2022). individual household electric power consumption dataset. [online]. available: https://archive.ics.uci.edu/ml/datasets/ individual+household+electric+power+consumption.

Abbasi, S., Mokhtarian, F., and Kittler, J. (2000). Enhancing css-based shape retrieval for objects with shallow concavities. *Image and vision computing*, 18(3):199–211.

Ashish, V., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Å., and Polosukhin, I. (2017). Attention is all you need. In *Proc. Adv. Neural Inf. Process. Syst.*, volume 30, pages 105–109.

Ayachi, L., Benkhlifa, A., Jribi, M., and Ghorbel, F. (2020). Une nouvelle description du contour de l'espace: Une représentation espace-echelle généralisée basée sur la courbure et la torsion.

Ayachi, L., Jribi, M., and Ghorbel, F. (2023). Generalized torsion-curvature scale space descriptor for 3-dimensional curves.

Bahdanau, D., Cho, K., and Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Box, G. (2015). *Time Series Analysis: Forecasting and Control*. Wiley, San Francisco, CA, USA.

ByoungSeon, C. (2012). *ARMA Model Identification*. Springer, New York, NY, USA.

Cho, K., van Merrienboer, B., Bahdanau, D., and Bengio, Y. (2014). On the properties of neural machine translation: Encoder-decoder approaches.

Dai, S., Chen, Q., Liu, Z., and Dai, H. (2020). Time series prediction based on emd-lstm model. *J. Shenzhen Univ. Sci. Eng.*, 37(3):221–230.

Gers, F. (2002). Applying lstm to time series predictable through time-window approaches. In *Neural Nets WIRN Vietri-01*, pages 193–200, London, U.K. Springer.

Junczys-Dowmunt, M., Grundkiewicz, R., Dwojak, T., Hoang, H., Heafield, K., Neckermann, T., Seide, F., Wiesler, S., Germann, U., and Fikri Aji, A. (2018). Marian: Fast neural machine translation in c++. *arXiv preprint arXiv:1806.09235*.

Liu, J. and Chen, S. (2019). Non-stationary multivariate time series prediction with mix gated unit. *J. Comput. Res. Develop.*, 56(8):1642.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

McLeod, A. I. and Li, W. K. (1983). Diagnostic checking arma time series models using squared-residual autocorrelations. *Journal of Time Series Analysis*, 4(4):269–273.

Nakkach, C., Zrelli, A., and Ezzeddine, T. (2022). Deep learning algorithms enabling event detection: A review. In *2nd International Conference on Industry 4.0 and Artificial Intelligence (ICIAI 2021)*. Atlantis Press.

Nakkach, C., Zrelli, A., and Ezzedine, T. (2023). Long-term energy forecasting system based on lstm and deep extreme machine learning. *Intelligent Automation & Soft Computing*, 37(1).

Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks. *Advances in neural information processing systems*.

Torres, J. L., García, A., Blas, M. D., and De Francisco, A. (2005). Forecast of hourly average wind speed with arma models in navarre (spain). *Solar Energy*, 79(1):65–77.

Yu, Y., Si, X., Hu, C., and Zhang, J. (2019). A review of recurrent neural networks: Lstm cells and network architectures. *Neural Computation*, 31(7):1235–1270.