A Deep Hybrid Learning Model for Photovoltaic Solar Tracking Systems

Musa Phiri
Department of Computer Science and Information Technology
Mulungushi University
Kabwe, Zambia
phirimusa@live.com

Mwenge Mulenga
Business Studies Division
National Institute of Public Administration
Lusaka Zambia
mwenge.research@gmail.com

Douglas Kunda
ZCAS University
Lusaka Zambia
douglas.kunda@zcasu.edu.zm

Fadele Ayotunde Alaba
Department of Computer Science
Federal College of Education
Zaria, Nigeria
ayotundefadele@yahoo.com

***Abstract*— The demand for clean and sustainable energy has increased the popularity of photovoltaic (PV) solar panels. PV solar panels are the most effective technology for transforming the sun's rays into a valuable energy source. To maximize the amount of energy captured, the rays of light from the sun must be at a 90-degree angle to the surface of the PV panel. To accomplish this, PV panels are modified with solar trackers that can effectively track the sun's position as it shifts during the day. Due to the randomness and nonlinearity of metrological data, using deep learning (DL) algorithms to enhance solar trackers has gained much popularity among researchers. However, given the varying nature of metrological data, single DL models sometimes fail to perform satisfactorily. For this purpose, this study applies sine and cosine transformations (SCT), a convolutional neural network (CNN), and long short-term memory (LSTM) to forecast the sun's trajectory. The SCT captures the cyclic components of the metrological data. The transformed data is then fed into a CNN-LSTM framework, where features of the sun's movement are learned. The SCT-CNN-LSTM framework's performance is evaluated concerning three other models based on the MAE, MAPE, and RMSE performance metrics. Based on the research findings, the CNN-LSTM framework has a superior performance, achieving an MAE, MAPE, and RMSE score of 0.0010, 73.5%, and 0.0012, respectively.**

Keywords—Clean Energy, Solar tracker, Deep hybrid learning, Photovoltaic panels, Metrological Data.

I. INTRODUCTION

In recent years, as the world aims to minimize its dependency on fossil fuels and combat climate change, integrating sustainable energy sources, such as solar energy, into the electrical power infrastructure has become increasingly essential. Various researchers have emphasized the importance of solar energy because of its abundance and environmental friendliness [1]. Further, solar energy is regarded as clean energy evenly distributed worldwide [2]. Currently, utilizing photovoltaic (PV) panels to convert sunlight into a functional form of energy has proven highly effective [3]. The optimal amount of sunlight captured by a PV panel occurs when the sun's beams are at a 90-degree angle to the surface of the PV panel. As a result, the output of the PV panel varies with the change in the sun's position. Therefore, to increase energy output, PV panels are modified with a solar tracker to identify the position of the sun and orient the PV panels with the sun's angle [4].

Generally, single-axis and double-axis solar trackers are the most commonly used solar trackers [5]. The rotation of single-axis solar tracker is based on one degree of freedom and is typically oriented toward the north meridian. There are four typical implementations of single-axis solar trackers: vertical, horizontal, polar-aligned, and horizontal-tilted single-axis solar trackers [5]. Conversely, double-axis solar trackers have two degrees of freedom determining a PV panel's movement when tracking the sun. The tip-tilt and azimuth-altitude double-axis solar trackers are two popular double-axis solar tracker designs.

Lately, there has been a growing focus on enhancing the efficiency of solar trackers through the application of conventional machine learning (ML) and DL approaches [6]. However, conventional ML models have limited capability in handling large datasets and require considerable expertise in understanding data representations [7]. Further, solar energy-based datasets have experienced significant growth, giving rise to the big data era. Due to the randomness and nonlinearity of metrological data, researchers have focused on using DL techniques for solar tracking. Given the varying nature of metrological data, single DL models sometimes fail to perform satisfactorily [8]. Thus, various works [9] have shown that one alternative is to develop a hybrid approach to enhance the predictive ability of an algorithm. Many researchers utilize the benefits of DL models for non-linear feature extraction. In particular, various DL models have been used to perform automatic feature extraction on metrological data [10], [11]. Similarly, other techniques, such as cyclical transformations, have been used to improve predictive performance [12], [13]. However, studies combining cyclical transformations with feature extraction to improve solar tracking accuracy are lacking in the literature.

In this paper, we implement a deep hybrid learning (DHL) algorithm that combines a CNN and LSTM to forecast the sun's location for the next hour. The algorithm comprises three components: SCT for recovering cyclical patterns, CNN for automatic spatial feature extraction, and LSTM for capturing temporal dependencies in data sequences. By combining SCT-CNN-LSTM, we aim to enhance ST by reconstructing data sequences for short-term sun trajectory predictions. Our main contribution lies in the integration of SCT-CNN-LSTM to improve ST.

The rest of the paper is organized in the following manner. The second section provides an overview of relevant literature. The third section outlines the proposed method. The fourth section covers the study's findings and accompanying discussions. Finally, the fifth section presents the conclusion and suggestions for future research.

II. Literature review

This section reviews cutting-edge techniques for predicting the sun's path in PV solar trackers. ST systems are essential in maximizing PV panels' energy efficiency. Given the recent development of intelligent ST systems, predicting the sun's position has become a research hotspot in renewable energy. Astronomical-based models, which use mathematical equations and astronomical data, have been applied to PV solar trackers. For instance, Sidek et al. [14] proposed a two-axis PV solar tracker based on a global positioning system (GPS) and astronomical equations. The equations were applied to calculate the sun's position regarding its orientation and elevation angles. The performance of the proposed two-axis solar tracker was evaluated by comparing its energy output to that of a stationary PV panel. The recorded results showed that the proposed PV solar tracker captured 27% more energy than the stationary PV panel. However, one drawback of the proposed equations was their inability to predict the sun's position during days of cloud cover.

Similarly, Fonseca-Campos et al. [15] proposed a mathematical model for tracking the sun's position using GPS and astronomical data. The proposed model showed high effectiveness in predicting the position of the sun. However, since the model was only tested on a small dataset of simulated data, its performance may have limited accuracy in a real-world application. Recently, the performance of ST systems has been enhanced using ML models.

Moreover, ML offers low cost, fast computation, and high accuracy [6]. For example, Kim et al. [16] implemented an ML-based algorithm for ST. The algorithm was used to forecast the best tilt orientations for maximizing a PV panel's energy production. In their study, Kim et al. [16] evaluated the effectiveness of five distinct ML algorithms, including random forest (RF), linear regression (LR), support vector machine (SVM), least minor absolute shrinkage and selection operator (LASSO), and gradient boosting (GB) algorithms. The authors found that the GB algorithm outperformed the other ML algorithms included in the research. Even though the research included a broad range of ML models, no feature engineering techniques were used. Similarly, Mondal and Mondol [17] investigated the efficiency of different ML models to improve the functionality of a PV ST system. While the study showed the superior performance of ML-based ST systems, the study also neglected to include any feature engineering methods.

Given the large amount of data required to implement accurate ST models, researchers have focused on using DL models to forecast the sun's position. Pierce et al. [18] recently proposed a multi-input CNN that uses sky image data to forecast the sun's path. Although the model could effectively extract relevant information for short-term forecasting, it did not account for geographical or seasonal effects. Similarly, Carballo et al. [19] successfully used a deep CNN (dCNN) model for computer vision-based ST. However, the study did not investigate how well the proposed dCNN model performed on other CNN-based architectures. In another study, AL-Rousan et al. [20] investigated and evaluated the most valuable features in predicting the sun's tilt and elevation angles. The optimal features were identified using correlation results between the different features and target variables. The authors used the identified variables in a DL model, which resulted in higher predictive performance. However, their model did not account for seasonal variations in the data.

In response to the shortcomings of single-based DL models, DHL models have recently gained significant attention among researchers. Deep hybrid models can automatically capture and integrate different hidden features to improve performance [8]. For instance, Frizzo Stefenon et al. [21] presented a combination of wavelet energy coefficient (WEC) and LSTM. The WEC was applied to extract signal characteristics and reduce noise. The LSTM was then used to perform time series-based forecasting. Although the authors were able to reduce the forecasting error using this method, they did not consider the cyclical effect of angular data on the model. In another study, Al-Muswe et al. [2] used an RNN with LSTM to forecast the sun's trajectory to maximize the ST system's performance. The authors were able to achieve high accuracy. However, the dataset was relatively small, which may have hindered the model's performance.

Based on the gaps highlighted in the reviewed literature, we apply an algorithm that combines cyclical transformations with feature extraction to improve the model's forecasting performance. The cyclical transformations used in this study convert the original features into their sine and cosine components. This allows for a more precise representation of the fundamental trends in the data [13]. The feature extraction technique is based on a CNN, which is widely used in renewable energy forecasting. In terms of metrological data, a CNN provides improved accuracy through its ability to classify various weather patterns in raw data. Therefore, the algorithm used in this study attempts to recover the cyclical patterns and extract hidden features in the data to improve solar position forecasting.

III. Method

This section presents the method used to develop a hybrid hour-ahead ST DL framework. The section begins by outlining the dataset used in the study, followed by an overview of the basic concepts of the SCT, LSTM, and CNN, and finally discussing how they were integrated. Further, the experiments conducted were implemented using the Python 3.9-based Keras framework with Tensorflow. The program was run using a Graphical Processing Unit (GPU) on Google Collaboratory.

A. Dataset Description

The proposed model was trained using real-world data from the Girasol sky imaging and global solar irradiance repository [22]. The dataset included features such as solar radiation, UNIX time, solar elevation angle, temperature, dew point, atmospheric pressure, wind direction, wind velocity, and relative humidity. The model aimed to predict the sun's elevation angle. The dataset covered 242 days of the solar cycle (2017-2019), with sun position data sampled multiple times per second and meteorological data sampled at 10-minute intervals. To match the time resolution, the meteorological data was interpolated. Additionally, minimum-maximum normalization was applied to scale the data.

B. Sine and Cosine Transformations

Renewable energy-based datasets, particularly those based on solar and wind energy, often have non-linear fluctuations due to weather patterns [8]. These patterns often have daily, seasonal, and yearly variations, making it difficult to predict future trends based on past patterns alone. To address this challenge, signal processing and engineering fields commonly use SCT for time series and circular analysis [23]. Therefore,

using SCT plays a significant role in analyzing and modeling complex renewable energy-related data. Conceptually, the sine and cosine transformation framework is expressed by the (1).

$$Y_m = \rho + \delta cos(2\pi\varphi m + \vartheta) + \varepsilon_m \quad (1)$$

Where, Y_m is a time series representing the outcome of interest at time m, $m = 1, 2, \ldots, N$, and N denoting an effective duration represented by the number of samples. The fundamental basis of Y_m is reflected by the constant ρ, while the cyclic component has an amplitude of δ, a frequency of φ, and a phase angle of ϑ. The error terms, represented by ε_m, follow a normal distribution and are independently and identically distributed with an expected value of zero ($E[\varepsilon_m] = 0$) and variance of σ^2 ($Var[\varepsilon_m] = \sigma^2$). The theoretical framework utilized to analyze cyclic behaviour involves a cosine function that increases and decreases proportionally throughout a year. Additionally, the timing of the peak concerning the origin is described by a phase angle or shift parameter, which can be used to determine the angular coordinates of two points [24]. In this case, the phase angle is expressed in time units in a time series and can be used to compare seasonal changes.

Assuming that a cycle or period of oscillation is known, then the frequency of a period in *m* units is a constant number [24]. Therefore, using three parameters which include the constant, amplitude, and phase, the conceptual framework can be reformulated as follows;

$$\begin{aligned} Y_m &= \rho + \delta cos(2\pi\varphi m + \vartheta) + \varepsilon_m \\ &= \rho + \beta_c cos(2\pi\varphi m) \\ &\quad + \beta_s sin(2\pi\varphi m) + \varepsilon_m \end{aligned} \quad (2)$$

Where, $\beta_c = -\delta\ cos\ \vartheta$ and $\beta_s = \delta\ sin\ \vartheta$ represent the model parameters. The temporal resolution of the data is described by $\varphi = 1/T$, where T represents the unit of analysis. For instance, if the unit of analysis is monthly, weekly, or daily data, then T is 12, 52.25, and 365.25, respectively. Therefore, based on (2), slight shifts in peaking timing can be recovered using individual sine and cosine components as shown by (3) and (4);

$$sin(2\pi\varphi m_i) = sin(2\pi m_i/T) \quad (3)$$
$$cos(2\pi\varphi m_i) = cos(2\pi m_i/T) \quad (4)$$

This study used the SCT to extract daily and monthly periodicity from the UNIX time (seconds) feature. Similarly, elevation angle and wind direction features were transformed into a 2-dimensional feature space by replacing each cyclic feature X with two features, cosine (X) and sine (X).

C. CNN Module

CNNs have gained significant attention in various fields, including fault diagnosis, computer vision, and speech recognition [25]. These models differ from conventional DL models by utilizing convolutional layers, reducing network parameters through weight sharing and local connectivity. Recent studies have demonstrated the effectiveness of CNNs in univariate time series forecasting [25]. In this approach, the CNN model leverages past observations to predict the next value. The CNN architecture consists of convolution, pooling, and fully connected layers. Therefore, this study employed a simple CNN module with a kernel size of 3 and 32 filters.

D. LSTM Module

LSTM, a type of RNN, is utilized for analyzing sequential data like time series due to its ability to mitigate the vanishing o exploding gradient problems faced by traditional RNNs [26]. With its internal memory unit and gate mechanism, LSTM can retain information across multiple time intervals. The LSTM unit comprises a cell with input, output, and forget gates, regulating the flow of information within the cell state. In this study, LSTM was employed to predict the sun's elevation angle one hour ahead using meteorological features. The model featured one LSTM layer and one dense layer, with 32 and 2 units respectively, predicting the sine and cosine values of the angle.

E. SCT-CNN-LSTM Framework

This section presents the workflow of the SCT-CNN-LSTM framework for optimized ST. The framework starts with data preprocessing, involving time-series resolution adjustment and data exception handling. The SCT technique is applied to selected features, transforming cyclic features into cosine and sine representations, creating a 2-D feature space. The transformed dataset is then fed into a convolution layer to extract internal patterns and uncover hidden dependencies in the meteorological data. The output of the convolutional layer is passed through an LSTM layer, capturing short and long-term temporal relationships. Finally, the LSTM layer's output is processed by a dense layer, generating the final forecasts. Fig. 1 depicts the framework's processing flow.

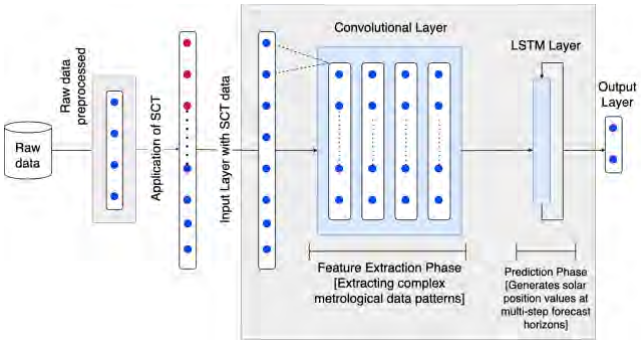

Fig. 1. SCT-CNN-LSTM Framework

F. Performance Metrics

When assessing the variance between a model's predictions and the observed values, common metrics employed include mean absolute error (MAE), mean absolute percentage error (MAPE), and root mean square error (RMSE). In this study, MAE is utilized to measure the average absolute difference between the predicted and actual vectors, as indicated by (5). The actual observations are denoted as m_i, the predicted observations as $\widehat{m}_i$, and p represents the sample size. A lower MAE value indicates a more desirable performance for the model.

$$MAE = \frac{1}{p}\sum_{i=1}^{p}|m_i - \widehat{m}_i| \quad (5)$$

MAPE quantifies the percentage difference between the actual and predicted observations, as described in (6). A low MAPE value is desired for a model's performance.

$$MAPE = \frac{1}{p}\sum_{i=1}^{p}\frac{m_i - \hat{m}_i}{m_i} \quad (6)$$

RMSE computes the square root of the average squared difference between the predicted and actual values, as shown in (7). A smaller RMSE value indicates a higher performance of the model.

$$RMSE = \sqrt{\frac{1}{p}\sum_{i=1}^{p}(m_i - \hat{m}_i)^2} \quad (7)$$

IV. Results and Discussion

The methods used in this research were discussed in the preceding section. Specifically, we outlined the SCTs, CNN, and LSTM. In addition, we discussed how these methods were combined to form the SCT-CNN-LSTM model, which was used to make hour-ahead predictions of the sun's position. To demonstrate the SCT-CNN-LSTM model's superiority, we implemented and evaluated its performance on other DL models, including the CNN, LSTM, and GRU models. The CNN-LSTM framework and other DL models are evaluated using three performance metrics, namely MAE, MAPE, and RMSE. For each DL model, 30 runs of model testing were made, and their performance results were recorded as shown in Table I. Further, significance testing was performed to ensure that the differences observed in the models' performance were statistically significant.

TABLE I. Model Performance Results.

	MAE				MAPE				RMSE			
Run#	LSTM	CNN	CNN+LSTM	GRU	LSTM	CNN	CNN+LSTM	GRU	LSTM	CNN	CNN+LSTM	GRU
1	0.0012	0.0009	0.0010	0.0017	134.7143	68.3040	48.8671	48.9125	0.0013	0.0010	0.0012	0.0018
2	0.0009	0.0010	0.0006	0.0036	231.7931	130.9524	132.8113	92.5745	0.0011	0.0026	0.0008	0.0039
3	0.0022	0.0014	0.0005	0.0031	198.5610	145.3635	83.4768	357.0733	0.0024	0.0018	0.0007	0.0040
4	0.0015	0.0007	0.0014	0.0034	60.1762	174.9091	36.2577	100.4340	0.0019	0.0011	0.0019	0.0039
5	0.0016	0.0006	0.0005	0.0016	144.7114	167.3322	104.5754	99.2664	0.0016	0.0008	0.0008	0.0019
6	0.0027	0.0024	0.0007	0.0010	57.1839	61.5591	85.2691	104.3806	0.0029	0.0027	0.0009	0.0013
7	0.0009	0.0020	0.0002	0.0043	146.6215	138.0900	63.5375	104.4056	0.0012	0.0022	0.0004	0.0046
8	0.0014	0.0026	0.0014	0.0013	47.1661	129.9514	47.4434	130.8226	0.0018	0.0034	0.0016	0.0014
9	0.0008	0.0007	0.0012	0.0015	123.9280	114.3099	86.7378	118.8935	0.0010	0.0010	0.0013	0.0017
10	0.0005	0.0010	0.0008	0.0013	54.1388	48.2754	72.0885	106.8177	0.0006	0.0016	0.0010	0.0017
11	0.0013	0.0043	0.0014	0.0020	94.5439	167.2682	98.4000	48.5874	0.0016	0.0044	0.0015	0.0024
12	0.0012	0.0026	0.0008	0.0019	91.8256	238.6070	50.4738	66.6388	0.0013	0.0033	0.0009	0.0022
13	0.0015	0.0047	0.0005	0.0019	121.8543	46.1097	71.6161	125.1795	0.0017	0.0051	0.0006	0.0025
14	0.0018	0.0013	0.0011	0.0016	55.0167	104.1268	77.9744	58.5866	0.0019	0.0015	0.0013	0.0019
15	0.0016	0.0040	0.0009	0.0044	138.4683	50.0205	72.7148	177.9281	0.0017	0.0049	0.0010	0.0046
16	0.0014	0.0011	0.0005	0.0005	113.0741	161.4797	77.7758	83.8191	0.0015	0.0012	0.0006	0.0007
17	0.0035	0.0027	0.0009	0.0014	217.1975	142.1745	95.1167	67.0519	0.0037	0.0030	0.0011	0.0016
18	0.0005	0.0011	0.0008	0.0057	260.7091	97.2478	97.6016	222.0819	0.0007	0.0015	0.0012	0.0059
19	0.0012	0.0020	0.0007	0.0015	99.1513	200.2443	64.2678	164.5121	0.0013	0.0026	0.0009	0.0018
20	0.0022	0.0017	0.0006	0.0036	94.5505	84.9969	34.9129	162.3456	0.0026	0.0022	0.0008	0.0040
21	0.0020	0.0023	0.0019	0.0020	260.7091	97.2478	97.6016	222.0819	0.0023	0.0024	0.0021	0.0022
22	0.0008	0.0012	0.0008	0.0009	43.5129	76.2825	78.2938	159.4505	0.0009	0.0017	0.0010	0.0011
23	0.0011	0.0014	0.0007	0.0012	44.1397	165.5625	62.4806	76.4664	0.0013	0.0016	0.0009	0.0014
24	0.0012	0.0019	0.0020	0.0018	106.2906	66.7375	52.1098	76.3902	0.0015	0.0021	0.0026	0.0020
25	0.0015	0.0022	0.0015	0.0021	99.2502	71.1337	47.7943	82.5341	0.0016	0.0025	0.0018	0.0023
26	0.0011	0.0017	0.0012	0.0043	142.1075	183.9240	96.7991	257.4666	0.0013	0.0035	0.0013	0.0054
27	0.0017	0.0015	0.0013	0.0044	104.8902	134.4932	69.6654	68.8093	0.0018	0.0027	0.0014	0.0056
28	0.0012	0.0010	0.0011	0.0032	54.4842	50.3920	57.2569	127.6479	0.0015	0.0011	0.0013	0.0038
29	0.0038	0.0010	0.0014	0.0025	273.3652	112.5635	62.5290	115.9429	0.0044	0.0014	0.0016	0.0028
30	0.0026	0.0017	0.0004	0.0024	103.0647	306.6463	78.5153	78.4121	0.0028	0.0020	0.0006	0.0029

A. *Evaluation of Model Performance and Stability of the Models.*

To illustrate the performance of the models, the average performance results of the different DL models are presented in Table II.

TABLE II. Average Performance of the DL Models.

Model	Performance Metric		
	MAE	MAPE	RMSE
LSTM	0.0016	123.9067	0.0018
CNN	0.0018	124.5435	0.0023
SCT-CNN-LSTM	0.0010	73.4988	0.0012
GRU	0.0024	123.5171	0.0028

According to Table II, the SCT-CNN-LSTM model achieved the best average MAE, MAPE, and RMSE results of 0.0010, 73.4988, and 0.0012 respectively. Additionally, the spread and medians of the MAE, MAPE, and RMSE results obtained are presented in Figs. 2, 3, and 4, respectively. The spread of the performance results is presented in terms of the quartile values, whereas the whiskers show variations outside the first and third quartile values. Furthermore, the points located beyond the whiskers represent extreme values.

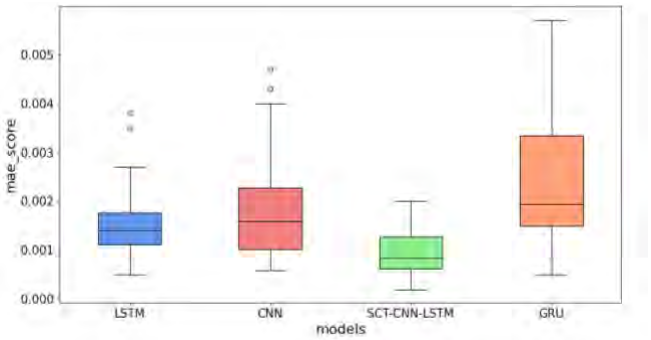

Fig. 2. Box Plot of Model Performance Results Based on MAE

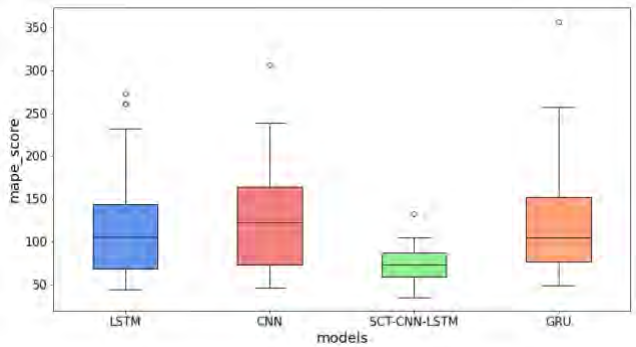

Fig. 3. Box Plot of Model Performance Results Based on MAPE

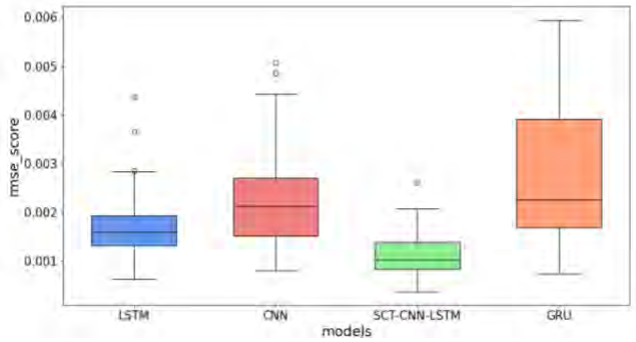

Fig. 4. Box Plot of Model Performance Results Based on RMSE

According to Figs. 2, 3, and 4. The SCT-CNN-LSTM recorded the lowest median score and smallest spread. These results observed further justify the remarkable performance of the SCT-CNN-LSTM framework in terms of minimizing errors and maintaining stability.

B. Significance Testing.

Significance testing was used to determine the statistical significance of the differences in performance of the SCT-CNN-LSTM against the single DL models used in this study. The models' performance results were tested for significance based on their MAE, MAPE, and RMSE scores. To begin, the MAE, MAPE, and RMSE performance results of each model were checked in accordance with the normal distribution. In this study, since we had a small sample size which was based on the performance results recorded from the 30 runs of model testing, the Shapiro-Wilk test was used to test for normality [27]. Based on the 0.05 level of significance, if the p-value is > 0.05, the data distribution is not significantly different from a normal distribution, whereas, if the p-value is < 0.05 there is a significant deviation from the normal distribution. The Shapiro-Wilk normality test results for the DL models based on their performance results are presented in Table III.

TABLE III. SHAPIRO-WILK NORMALITY TEST RESULTS FOR DL MODELS.

Model	MAE		MAPE		RMSE	
	W	*p-value*	*W*	*p-value*	*W*	*p-value*
CNN	0.8639	0.0012	0.9300	0.0493	0.9187	0.0248
LSTM	0.8857	0.0038	0.8851	0.0037	0.8817	0.0031
SCT-CNN-LSTM	0.9548	0.2267	0.9726	0.6114	0.9402	0.0924
GRU	0.9142	0.0190	0.8401	0.0004	0.9091	0.0141

According to Table III, the Shapiro-Wilk test showed that all but the performance results of the SCT-CNN-LSTM departed significantly from normality with p-values < 0.05 and very high statistic (W) values. Thus, due to the sample deviation from the normal distribution, the Kruskal-Wallis (H) non-parametric test was used on the samples of model performance to verify that the results from the different DL models differ significantly [28]. The null hypothesis of the Kruskal-Wallis (H) test is that the data samples do not differ. If the p-value is less than the chosen significance level ($\alpha = 0.05$), we reject the null hypothesis and conclude that there is a significant difference between the group's medians. Table IV presents the Kruskal-Wallis results for the model performance comparisons.

TABLE IV. KRUSKAL-WALLIS TEST RESULTS BASED ON MODEL PERFORMANCE COMPARISON.

Performance Metric	*H*	*p-value*
MAE	33.3544	2.71E-07
MAPE	17.9090	4.59E-04
RMSE	37.1715	4.23E-08

Based on Table IV, all the p-values recorded are considerably less than 0.05, indicating strong evidence against the null hypothesis of equal medians in performance scores among the DL models. Therefore, we reject the null hypothesis, suggesting at least one DL method has a significantly different median performance score compared to the others. To further investigate the pairwise differences between the DL models post-hoc testing was done using Bonferroni-Dunn's test. A two-tailed null hypothesis at the 0.05 level of significance was employed. The results of the significance testing for the CNN-LSTM framework against the other models using testing their MAE, MAPE, and RMSE scores are presented in Tables V, VI, and VII.

TABLE V. RESULTS OF POST-HOC TESTS OF HYBRID AND SINGLE MODELS' MAE PERFORMANCE.

Comparison	*p-value*	*H_0
SCT-CNN-LSTM vs CNN	1.12E-04	***Reject***
SCT-CNN-LSTM vs LSTM	1.85E-03	***Reject***
SCT-CNN-LSTM vs GRU	1.61E-08	***Reject***

TABLE VI. RESULTS OF POST-HOC TESTS OF HYBRID AND SINGLE MODELS' MAPE PERFORMANCE.

Comparison	*p-value*	*H_0
SCT-CNN-LSTM vs CNN	2.84E-04	***Reject***
SCT-CNN-LSTM vs LSTM	9.82E-04	***Reject***
SCT-CNN-LSTM vs GRU	6.57E-04	***Reject***

TABLE VII. RESULTS OF POST-HOC TESTS OF HYBRID AND SINGLE MODELS' RMSE PERFORMANCE.

Comparison	*p-value*	*H_0
SCT-CNN-LSTM vs CNN	4.00E-06	***Reject***
SCT-CNN-LSTM vs LSTM	3.41E-03	***Reject***
SCT-CNN-LSTM vs GRU	1.16E-08	***Reject***

According to the results in Tables V, VI, and VII, the group-wise p-value observed in the comparison of the hybrid model and the single-based DL models are less than 0.05. Thus, it can be concluded that the results attained by the models differ significantly from one another.

V. CONCLUSION

This study has shown that deep hybrid techniques can significantly improve ST. The method applied in this study combined cyclical transformations with feature extraction to improve solar tracking accuracy. This was performed using SCT and a CNN model for automatic feature extraction. Further, the LSTM model was used to forecast the subsequent time interval by considering the past data on the sun's position. The deep hybrid model was evaluated against three models based on the MAE, MAPE, and RMSE performance metrics. In the experiment, the SCT-CNN-LSTM model attained the most favorable average performance compared to the other models. Further, the SCT-CNN-LSTM model was the most stable, proving that it can be applied to enhance solar tracking technologies in diverse weather conditions. Additionally, significance testing was conducted. The test proved that the performance scored by the SCT-CNN-LSTM was statistically different from the other base models. This provided further justification that the hybrid-based model provides superior results in predicting the sun's position.

REFERENCES

[1] S. Abdallah and O. O. Badran, "Sun tracking system for productivity enhancement of solar still," *Desalination*, vol. 220, no. 1–3, pp. 669–676, Mar. 2008, doi: 10.1016/j.desal.2007.02.047.

[2] A. F. A. Al-Muswe, I. K. A. Al-Amri, M. N. H. Al-Turfi, and A. Alkhayyat, "Solar Tracking System by Utilized Optimized Algorithm Based Deep Learning," in *2022 5th International Conference on Engineering Technology and its Applications (IICETA)*, Al-Najaf, Iraq: IEEE, May 2022, pp. 37–44. doi: 10.1109/IICETA54559.2022.9888545.

[3] M. Asif and T. Muneer, "Energy supply, its demand and security issues for developed and emerging economies," *Renew. Sustain. Energy Rev.*, vol. 11, no. 7, pp. 1388–1413, Sep. 2007, doi: 10.1016/j.rser.2005.12.004.

[4] M. Ali Jallal, S. Chabaa, and A. Zeroual, "A novel deep neural network based on randomly occurring distributed delayed PSO algorithm for monitoring the energy produced by four dual-axis solar trackers," *Renew. Energy*, vol. 149, pp. 1182–1196, Apr. 2020, doi: 10.1016/j.renene.2019.10.117.

[5] A. Z. Hafez, A. M. Yousef, and N. M. Harag, "Solar tracking systems: Technologies and trackers drive types – A review," *Renew. Sustain. Energy Rev.*, vol. 91, pp. 754–782, Aug. 2018, doi: 10.1016/j.rser.2018.03.094.

[6] N. AL-Rousan, N. A. Mat Isa, and M. K. Mat Desa, "Efficient single and dual axis solar tracking system controllers based on adaptive neural fuzzy inference system," *J. King Saud Univ. - Eng. Sci.*, vol. 32, no. 7, pp. 459–469, Nov. 2020, doi: 10.1016/j.jksues.2020.04.004.

[7] G. Alkhayat and R. Mehmood, "A review and taxonomy of wind and solar energy forecasting methods based on deep learning," *Energy AI*, vol. 4, p. 100060, Jun. 2021, doi: 10.1016/j.egyai.2021.100060.

[8] B. Gao, X. Huang, J. Shi, Y. Tai, and J. Zhang, "Hourly forecasting of solar irradiance based on CEEMDAN and multi-strategy CNN-LSTM neural networks," *Renew. Energy*, vol. 162, pp. 1665–1683, Dec. 2020, doi: 10.1016/j.renene.2020.09.141.

[9] G. Li, S. Xie, B. Wang, J. Xin, Y. Li, and S. Du, "Photovoltaic Power Forecasting with a Hybrid Deep Learning Approach," *IEEE Access*, vol. 8, pp. 175871–175880, 2020, doi: 10.1109/ACCESS.2020.3025860.

[10] S. Ghimire, R. C. Deo, N. Raj, and J. Mi, "Deep solar radiation forecasting with convolutional neural network and long short-term memory network algorithms," *Appl. Energy*, vol. 253, p. 113541, Nov. 2019, doi: 10.1016/j.apenergy.2019.113541.

[11] W. Lee, K. Kim, J. Park, J. Kim, and Y. Kim, "Forecasting Solar Power Using Long-Short Term Memory and Convolutional Neural Networks," *IEEE Access*, vol. 6, pp. 73068–73080, 2018, doi: 10.1109/ACCESS.2018.2883330.

[12] Y. Zhang, Y. Zhang, and A. Haghani, "A hybrid short-term traffic flow forecasting method based on spectral analysis and statistical volatility model," *Transp. Res. Part C Emerg. Technol.*, vol. 43, pp. 65–78, Jun. 2014, doi: 10.1016/j.trc.2013.11.011.

[13] Z. Song and L. E. Brown, "Multi-dimensional Evaluation of Temporal Neural Networks on Solar Irradiance Forecasting," in *2019 IEEE Innovative Smart Grid Technologies - Asia (ISGT Asia)*, Chengdu, China: IEEE, May 2019, pp. 4192–4197. doi: 10.1109/ISGT-Asia.2019.8881784.

[14] M. H. M. Sidek *et al.*, "GPS based portable dual-axis solar tracking system using astronomical equation," in *2014 IEEE International Conference on Power and Energy (PECon)*, Kuching, Sarawak, Malaysia: IEEE, Dec. 2014, pp. 245–249. doi: 10.1109/PECON.2014.7062450.

[15] J. Fonseca-Campos, L. Fonseca-Ruiz, and P. N. Cortez-Herrera, "Portable system for the calculation of the sun position based on a laptop, a GPS and Python," in *2016 IEEE International Autumn Meeting on Power, Electronics and Computing (ROPEC)*, Ixtapa: IEEE, Nov. 2016, pp. 1–5. doi: 10.1109/ROPEC.2016.7830580.

[16] G. Y. Kim, D. S. Han, and Z. Lee, "Solar Panel Tilt Angle Optimization Using Machine Learning Model: A Case Study of Daegu City, South Korea," *Energies*, vol. 13, no. 3, p. 529, Jan. 2020, doi: 10.3390/en13030529.

[17] J. Mondal and N. Mondol, *Horizontal Axis Solar Tracking System to Enhance Efficiency with Accuracy Test by Machine Learning*. 2022.

[18] B. G. Pierce, J. L. Braid, J. S. Stein, J. Augustyn, and D. Riley, "Solar Transposition Modeling via Deep Neural Networks with Sky Images," *IEEE J. Photovolt.*, vol. 12, no. 1, pp. 145–151, Jan. 2022, doi: 10.1109/JPHOTOV.2021.3120508.

[19] J. A. Carballo, J. Bonilla, M. Berenguel, J. Fernández, and G. García, "Solar tower power mockup for the assessment of advanced control techniques," *Renew. Energy*, vol. 149, pp. 682–690, Apr. 2020, doi: 10.1016/j.renene.2019.12.075.

[20] N. AL-Rousan, N. A. Mat Isa, and M. K. Mat Desa, "Correlation analysis and MLP/CMLP for optimum variables to predict orientation and tilt angles in intelligent solar tracking systems," *Int. J. Energy Res.*, vol. 45, no. 1, pp. 453–477, Jan. 2021, doi: 10.1002/er.5676.

[21] S. Frizzo Stefenon, C. Kasburg, A. Nied, A. C. Rodrigues Klaar, F. C. Silva Ferreira, and N. Waldrigues Branco, "Hybrid deep learning for power generation forecasting in active solar trackers," *IET Gener. Transm. Distrib.*, vol. 14, no. 23, pp. 5667–5674, Dec. 2020, doi: 10.1049/iet-gtd.2020.0814.

[22] G. Terrén-Serrano, A. Bashir, T. Estrada, and M. Martínez-Ramón, "Girasol, a sky imaging and global solar irradiance dataset," *Data Brief*, vol. 35, p. 106914, Apr. 2021, doi: 10.1016/j.dib.2021.106914.

[23] T. Ouyang, A. Kusiak, and Y. He, "Predictive model of yaw error in a wind turbine," *Energy*, vol. 123, pp. 119–130, Mar. 2017, doi: 10.1016/j.energy.2017.01.150.

[24] K. Ramanathan *et al.*, "Assessing Seasonality Variation with Harmonic Regression: Accommodations for Sharp Peaks," *Int. J. Environ. Res. Public. Health*, vol. 17, no. 4, p. 1318, Feb. 2020, doi: 10.3390/ijerph17041318.

[25] O. Abdel-Hamid, A. Mohamed, H. Jiang, L. Deng, G. Penn, and D. Yu, "Convolutional Neural Networks for Speech Recognition," *IEEEACM Trans. Audio Speech Lang. Process.*, vol. 22, no. 10, pp. 1533–1545, Oct. 2014, doi: 10.1109/TASLP.2014.2339736.

[26] V. Gundu and S. P. Simon, "Short Term Solar Power and Temperature Forecast Using Recurrent Neural Networks," *Neural Process. Lett.*, vol. 53, no. 6, pp. 4407–4418, Dec. 2021, doi: 10.1007/s11063-021-10606-7.

[27] P. Royston, "Approximating the Shapiro-Wilk W-test for non-normality," *Stat. Comput.*, vol. 2, no. 3, pp. 117–119, Sep. 1992, doi: 10.1007/BF01891203.

[28] P. E. McKight and J. Najab, "Kruskal-Wallis Test," in *The Corsini Encyclopedia of Psychology*, I. B. Weiner and W. E. Craighead, Eds., 1st ed.Wiley, 2010, pp. 1–1. doi: 10.1002/9780470479216.corpsy0491.