Journal of Trends in Computer Science and Smart Technology (ISSN: 2582-4104)
www.irojournals.com/tcsst/

Comparative Benchmarking Study of Deep Learning Models for Geomagnetic Storms Prediction

Prajwal Bhandari[1], Subarna Shakya[2]

[1]School of Mathematical Science, Tribhuvan University, Kathmandu, Nepal.
[2]Institute of Engineering, Tribhuvan University, Kathmandu, Nepal.

E-mail: [1]prajwalbhandari416@gmail.com, [2]drss@ioe.edu.np

Orcid ID: [1]0009-0006-7073-4079, [2]0000-0003-2268-0097

Abstract

Geomagnetic storms have been known to have a substantial impact on the performance of satellites, communication systems, navigation systems, and other space weather-based systems. This research is concerned with developing a data-driven forecast of geomagnetic activity using deep learning and traditional machine learning models. This study focuses on comparing deep learning and machine learning models for predicting the Disturbance Storm Time (Dst) index through solar wind and geomagnetic parameters. For the analysis, an hourly dataset for 2000-2024 has been utilized resulting 219,168 observations. From these, eight parameters including magnetic field components, solar wind speed, geomagnetic indices, and past Dst values have been selected. Additionally, Random Forest, XGBoost, LightGBM, and persistence were employed as traditional baseline models. The performance measures considered included RMSE, MAE, R^2, and Pearson correlation, alongside training time, inference time, number of parameters, and a feature ablation study. Among the evaluated deep-learning architectures, LSTM achieved the best average performance across the five independent seeds. The RMSE of LightGBM and persistence baselines was 3.77459 nT and 4.63338 nT, respectively. The ablation analysis shows that AE, Kp10, and solar-wind variables substantially influence forecasting performance, whereas historical Dst and a longer 48-hour window do not necessarily improve the reported error. Overall, the results have shown that it is possible to model nonlinear temporal dependencies in geomagnetic observations using recurrent architectures. However, the use of traditional baselines along with repeated-seeds and ablation tests makes the evaluation more thorough.

Keywords: Geomagnetic Storm Prediction, Disturbance Storm Time (Dst) Index, Space Weather Forecasting, Time-Series Forecasting, Deep Learning, Long Short-Term Memory (LSTM), Bidirectional LSTM (BiLSTM).

1. Introduction

Geomagnetic storms are considered large-scale disturbances of Earth's magnetosphere that result from the enhancement of energy exchange from the solar wind to the space environment around Earth. These storms are usually caused by coronal mass ejections (CMEs) and fast streams of the solar wind from coronal holes on the Sun. Upon interaction between

DOI: https://doi.org/10.36548/jtcsst.2026.3.022
Received: 24.06.2026, received in revised form: 27.07.2026, accepted: 18.08.2026, published: 29.08.2026

such solar wind phenomena and Earth's magnetic field, changes in geomagnetic indices may arise. The study of geomagnetic storms became relevant because of their effects on both nature and technology. From a natural perspective, the storms increase the intensity of the auroras, which appear quite beautiful but are also very energetic. However, from a technological perspective, these are quite dangerous events. They have been found to cause power outages, disturbances in the functioning of satellites, navigational problems, and radiation risks for astronauts. Geomagnetic storms are large-scale disruptions in Earth's space weather due to disturbances in solar wind and coronal mass ejections. They cause significant disturbances to satellites, communication systems, navigation systems, and power grids on Earth. Therefore, it has become imperative to predict the occurrence of geomagnetic storms and ionosphere disturbances [1][2][3]. Geomagnetic storms are disturbances in Earth's magnetosphere caused mainly by variations in solar wind conditions and the Sun's eruptive activities, such as CMEs. Predicting geomagnetic storms is essential to minimize their detrimental impact on modern-day technological systems [4].

The Disturbance Storm Time (Dst) index refers to the average change in the horizontal (H) component of Earth's magnetic field and is measured by five geomagnetic observatories located at low latitudes worldwide. The Dst index is the most significant geomagnetic index for measuring the strength geomagnetic storms [5].

Recent advances in machine learning and deep learning have demonstrated strong potential in modeling complex nonlinear relationships between solar wind parameters and geomagnetic or ionospheric responses. Unlike conventional physics models, deep learning techniques have an ability to learn sophisticated patterns from huge amounts of space weather data, thus increasing the accuracy of their predictions. With the further development of tailor-made models, they could be used as a base for building early-warning systems for forecasting geomagnetic storms [6]. LSTMs have become an extremely efficient method for modeling complicated non-linear temporal dependence in space weather time-series data, which makes LSTMs appropriate for forecasting geomagnetic and ionospheric disturbances. LSTM is a recurrent neural network with memory cells that enable retention of information over long sequences. Therefore, LSTM has the ability to learn not only short-term variations but also long-term trends in input variables like solar wind variables and geomagnetic indices.

1.1 Objectives

The general objective of this study is to comparatively evaluate deep learning architectures for predicting geomagnetic disturbances using Dst time-series data, with an emphasis on their temporal forecasting performance and applicability to space weather prediction.

1. To compare the performance of LSTM, BiLSTM, CNN-LSTM, and TCN models for predicting geomagnetic disturbances using Dst time-series data.
2. To implement and evaluate the selected deep learning models using standard forecasting metrics, including RMSE, MAE, R^2, correlation coefficient, and prediction efficiency.
3. To examine model prediction behavior and residual errors during geomagnetic disturbances, with an emphasis on prediction deviations during rare and strong storm-time conditions.

Unlike studies that focus primarily on developing or optimizing a single forecasting architecture, this work establishes a reproducible comparative benchmarking framework for Dst forecasting. The framework evaluates heterogeneous deep-learning architectures under a common chronological data split, identical input representation, controlled hyperparameter selection, five independent random seeds, conventional machine-learning baselines, feature and temporal ablation, and computational benchmarking. The contribution is therefore methodological and evaluative rather than the proposal of a new neural-network architecture. This design enables a controlled assessment of the relative strengths, variability, and computational characteristics of alternative forecasting approaches under the same experimental conditions

2. Literature Review

Early efforts to model geomagnetic storm activity focused on establishing quantitative relationships between solar wind conditions and geomagnetic indices. [7] introduced one of the most influential empirical models linking solar wind velocity and the southward component of the interplanetary magnetic field (IMF Bz) to the disturbance storm time (Dst) index through a differential equation framework. The theoretical expression they developed enabled a methodical way of determining ring current amplification and decay processes, which formed the basis of future geomagnetic storm forecast models. The method they developed revealed that the main external forces affecting the geomagnetism process come from the solar wind and are dependent on energy injection.

Based on this approach, [8] suggested the energy coupling function (ECF), known as the ϵ parameter, describing the energy flow rate from the solar wind to Earth's magnetosphere. The ECF includes parameters like solar wind velocity, IMF strength, and IMF clock angle and represents the efficiency of magnetic reconnection processes. Akasofu's theory increased understanding of the physical processes responsible for solar wind – magnetosphere interaction and laid theoretical foundations for storm strength estimation using solar wind characteristics observed in the upstream region [8]. After that, [9] introduced the universal function describing solar wind-magnetosphere coupling, which improved predictions due to the combination of several solar wind parameters into one. This model has shown better correlations with geomagnetic indices like Dst and Kp, and has been successfully used in numerous empirical models of space weather forecasting [8].

The ionosphere constitutes another important region affected by space weather phenomena, despite disruptions in the magnetosphere indicated by geomagnetic indices. Geomagnetic storms affect the electric field, plasma drift, and electron density distribution in the ionosphere, according to an exhaustive discussion of the dynamics of ionospheric plasma [3] [10]. Such disruptions usually manifest themselves as TEC disturbances, causing navigation and communication failures.

In 2025, [11], proposed a prediction approach for the Dst index based on LSTM with multi-decade solar wind and geomagnetic data. The researchers formulated different LSTM models for normal and stormy periods and thereafter joined them to form an integrated prediction framework. The findings revealed high correlation coefficients (more than ~0.94) and minimum root-mean-square errors for the short-term prediction of the intensity of geomagnetic storms. This proved that LSTM networks can predict changes in Dst for both normal and stormy.[12] proposed a hybrid model architecture that facilitates the model's ability to learn temporal dependencies of historical TEC data and spatial dependencies of geographic

regions. The experimental results show that the LSTM-STT framework demonstrates high prediction accuracy for different geomagnetic environments, where it maintains low average error rates that amount to several TEC units, irrespective of the solar activity level. The highly pronounced relative performance indicators in these papers show that using spatial attention mechanisms increases the ability of LSTM neural networks to adapt to the complicated forecasting of ionospheric activity. This demonstrates that the use of LSTM neural network models, along with a spatiotemporal learning framework, is a good data-driven solution for forecasting the Total Electron Content (TEC) of the ionosphere. This approach is especially useful when considering the effects of solar wind changes and geomagnetic disturbances on the ionosphere.

Ionosphere forecast studies further reinforce the efficiency of LSTM models compared to conventional techniques. Studies comparing ARIMA, Seq2Seq, and LSTM models for the prediction of storm-time TEC showed that LSTM consistently outperforms the other two, especially in modeling the trends of intense geomagnetic storms. This again emphasizes the use of LSTM for nonlinear and storm-induced time-series analysis [13]. Recent environmental forecast studies have also utilized hybrid and machine learning models to predict time-varying environmental parameters [15] [16].

3. Methodology

3.1 Research Design

In this study, we develop a comparative time-series forecasting model to forecast the Disturbance Storm Time (Dst) index, which t will be our target variable to describe the activity of geomagnetic storms. In this work, we assess four deep learning models with different temporal modeling techniques: Long Short-Term Memory (LSTM), Bidirectional Long Short-Term Memory (BiLSTM), Convolutional Neural Network–Long Short-Term Memory (CNN-LSTM), and Temporal Convolutional Network (TCN).

In addition to the deep-learning models, conventional machine-learning models and a persistence baseline were incorporated to establish meaningful performance references. The complete experimental workflow consists of data acquisition, quality assessment, preprocessing, chronological data partitioning, sequence generation, model construction, hyperparameter optimization, repeated five-seed evaluation, conventional baseline comparison, computational analysis, and ablation experiments. The overall methodology is designed to ensure that all competing models are evaluated using the same temporal dataset and target definition.

3.2 Dataset Description

3.2.1 Data Source

The dataset consists of hourly solar-wind and geomagnetic measurements obtained from the NASA OMNIWeb database. The original dataset covers the period from January 1, 2000 to December 31, 2024, providing a long-term temporal record for geomagnetic forecasting.

From Table 1, the dataset consists of solar wind magnetic field values, solar wind speed values, geomagnetic indices, and the Dst target/history variable.

Table 1. Variables and their roles in the Dst forecasting dataset derived from the NASA OMNI hourly database

Variable	Description	Role
B	Total magnetic-field magnitude	Input
Bx	X-component of the interplanetary magnetic field	Input
By	Y-component of the interplanetary magnetic field	Input
Bz	Z-component of the interplanetary magnetic field	Input
Vt	Solar-wind velocity	Input
Kp10	Planetary geomagnetic index	Input
AE	Auroral electrojet index	Input
Dst	Disturbance Storm Time index	Target/Input history
ProtonFlux	Proton flux	Excluded after missing-value assessment

The original dataset contained ProtonFlux as an additional variable. However, the missing-value audit identified 193,156 missing ProtonFlux observations, corresponding to 88.1315% of the dataset. In contrast, B, Bx, By, and Bz each contained only 304 missing observations (0.1387%), while Vt, Kp10, Dst, and AE contained no missing values. Because of the extremely high proportion of missing ProtonFlux values, ProtonFlux was excluded from the final modeling input. The final modeling dataset therefore contains eight input/target variables: B, Bx, By, Bz, Vt, Kp10, AE, and Dst. After preprocessing, no missing values remained in these variables.

3.2.2 Data Cleaning and Missing-Value Handling

A data-quality audit was first performed to identify missing observations and invalid measurements. The IMF-related variables B, Bx, By, and Bz contained 304 missing observations each, whereas Vt, Kp10, Dst, and AE contained no missing observations. ProtonFlux was removed because its missing rate exceeded 88%. Linear interpolation was applied to isolated missing values in the retained input variables B, Bx, By, and Bz before chronological partitioning. Each missing value was estimated from the nearest available observations surrounding the missing timestamp. Thus, only isolated gaps bounded by valid observations were interpolated; no extrapolation was performed for missing values at the beginning or end of the time series. This preprocessing was restricted to input variables and did not interpolate or otherwise reconstruct the Dst target. Since interpolation was applied only to input measurements to maintain temporal continuity, it was not used to generate target values or alter the forecasting labels.

3.3 Feature Scaling

Because the variables have substantially different numerical ranges, Min–Max normalization was applied before model training. To avoid information leakage, the minimum and maximum values necessary for scaling were estimated only using the training subset. This training subset was then used without alteration to scale the validation and testing subsets. Scaling can be represented by

$$X_{\text{scaled}} = \frac{X - X_{\text{min}}}{X_{\text{max}} - X_{\text{min}}}$$

where X represents an observed feature value, while X_{min} and X_{max} denote the minimum and maximum values calculated from the training subset. This procedure transforms

the variables into the [0,1] range while ensuring that information from the validation and testing subsets is not used during model fitting.

3.4 Chronological Data Partitioning

Since the objective is time-series forecasting, the dataset was divided chronologically rather than randomly. This prevents future observations from being introduced into the training process and provides a more realistic evaluation of forecasting performance. The final chronological partition is summarized in Table 2.

Table 2. Chronological partitioning of the dataset into training, validation, and testing subsets

Dataset	Observations	Percentage
Training	153,417	70%
Validation	32,875	15%
Testing	32,876	15%
Total	219,168	100%

The chronological partition was performed before sequence construction. Because the forecasting model requires historical context, observations immediately preceding the validation and testing boundaries were retained solely to construct the initial 48-hour input sequences; their corresponding target observations remain within the held-out validation and testing partitions.

The training subset was used for parameter optimization, the validation subset was used for hyperparameter selection and early stopping, and the held-out testing subset was reserved for final performance assessment.

3.5 Time-Series Sequence Generation

Geomagnetic activity exhibits temporal dependence; therefore, the observations were converted into fixed-length sequences using a sliding-window strategy.

For a lookback length T, the input sequence is defined as

$$X_t = \{x_{t-T}, x_{t-T+1}, \dots, x_{t-1}\}$$

and the forecasting target is

$$y_t = Dst_t.$$

A 48-hour lookback window was used in the primary experiment. Thus, each model receives the preceding 48 hourly observations to predict the Dst value at the next time step.

The resulting sequence dimensions were:

- Xtrain=(153369,48,8)
- Xvalidation=(32875,48,8)
- Xtest=(32876,48,8)

The training sequences were generated entirely within the training partition; therefore, applying the 48-hour lookback reduces the training sequence count from 153,417 observations to 153,369 sequences. For validation and testing, the first forecast targets retain access to the immediately preceding 48 hours of historical observations at the partition boundaries. These preceding

observations are used only as historical input context and are not included as validation or test targets. Consequently, the validation and testing sequence counts remain 32,875 and 32,876, respectively. This boundary-aware sequence construction reflects the operational forecasting setting in which observations immediately preceding a forecast time are available as historical context.

3.6 Deep-Learning Models

LSTM: The LSTM model receives a 48 × 8 input sequence and uses a 64-unit LSTM layer followed by dropout with a rate of 0.20. The recurrent representation is passed to a fully connected layer with 32 ReLU units and subsequently to a single linear output neuron for Dst prediction. The architecture contains 20,801 trainable parameters.

BiLSTM: The BiLSTM model uses a bidirectional LSTM with 64 units in each direction, followed by dropout (0.20), a 32-unit ReLU dense layer, and a single linear output neuron. The bidirectional representation has 128 dimensions before the dense layer, resulting in 41,537 trainable parameters.

CNN-LSTM: The CNN-LSTM architecture first applies a 1-D convolution with 64 filters and kernel size 5 using same padding and ReLU activation. The resulting sequence is processed by an LSTM with 128 units, followed by dropout (0.30), a 32-unit ReLU dense layer, and a single linear output neuron. The model contains 105,601 trainable parameters.

TCN: The TCN consists of four causal 1-D convolutional layers with 64 filters and kernel size 5, using dilation rates of 1, 2, 4, and 8, respectively. Each convolutional layer is followed by dropout (0.30). Global average pooling is then applied, followed by a 32-unit ReLU dense layer and a single linear output neuron. The resulting model contains 66,369 trainable parameters.

3.6.1 Temporal Convolutional Network

The TCN architecture models temporal dependencies using causal temporal convolutions. The selected configuration uses 64 filters, kernel size 5, dropout 0.30, and learning rate 0.0005. The resulting model contains 66,369 trainable parameters.

Table 3. Deep-learning model configurations

Model	Units	Filters	Kernel Size	Dropout	Learning Rate	Parameters
LSTM	64	—	—	0.20	0.001	20,801
BiLSTM	64	—	—	0.20	0.001	41,537
CNN-LSTM	128	64	5	0.30	0.0005	105,601
TCN	—	64	5	0.30	0.0005	66,369

3.7 Training Configuration

All deep-learning models were trained using the Adam optimizer with mean squared error (MSE) as the training loss. A batch size of 32 and a maximum of 100 epochs were specified, with early stopping using a patience of 5 epochs to reduce overfitting.

Table 4. Common training configuration

Parameter	Setting
Input window	48 hours
Batch size	32
Optimizer	Adam
Loss function	MSE
Maximum epochs	100
Early stopping patience	5
Primary target	Dst
Evaluation strategy	Chronological test set
Repeated evaluation	5 random seeds

The implementation was executed using TensorFlow 2.20.0 with GPU acceleration.

3.8 Hyperparameter Optimization

To reduce the influence of training configuration on the comparison, a limited validation-based hyperparameter screening procedure was performed before the final repeated evaluation. Two candidate configurations were evaluated for each architecture. The search was intentionally restricted to a small candidate set because the primary objective of the study was controlled architectural benchmarking rather than exhaustive hyperparameter optimization. The validation subset was used for configuration selection, while the held-out test set was not used during this process.

For LSTM and BiLSTM, the candidate configurations included:

- units: 64 or 128;
- dropout: 0.20 or 0.30;
- learning rate: 0.001 or 0.0005.

For CNN-LSTM, the search additionally included:

- convolutional filters: 32 or 64;
- kernel size: 3 or 5.

For TCN, the search included:

- filters: 32 or 64;
- kernel size: 3 or 5;
- dropout: 0.20 or 0.30;
- learning rate: 0.001 or 0.0005.

The final configurations were selected based on validation MSE. The optimization experiments showed, for example, that the selected CNN-LSTM configuration reduced the best validation MSE from approximately 0.000562 for the first trial to approximately 0.000245 for the second trial. Similarly, the selected TCN configuration was obtained by comparing the two candidate configurations. The test set was not used for hyperparameter selection

3.9 Five-Seed Evaluation and Uncertainty Estimation

To reduce dependence on a single random initialization, the optimized configuration of each deep-learning model was retrained using five independent random seeds: 11, 22, 33, 44, and 55. For every seed, RMSE, MAE, R^2, Pearson correlation, training time, inference time, and the number of trainable parameters were recorded.

For each model, the final result was reported as M ±SD, where M is the arithmetic mean over the five runs and SD is the corresponding standard deviation. This repeated-run protocol provides an empirical measure of model stability and allows the variability caused by random initialization to be explicitly reported.

3.10 Conventional Machine-Learning Baselines

To determine whether the deep-learning architectures provide an advantage over conventional forecasting approaches, tree-based machine-learning models were incorporated into the benchmark.

The evaluated conventional models were:

1. Random Forest;
2. XGBoost;
3. LightGBM.

A persistence baseline was also included. For the persistence model, the last known Dst value is used to predict the future Dst value at the next step. The RMSE, MAE, R^2, and correlation values of the persistence baseline are 4.6334, 2.9174, 0.9538, and 0.9769 respectively. The same time-based evaluation process has been adopted in the case of conventional baselines to ensure the consistency of the time-based prediction problem.

3.11 Performance Evaluation Metrics

The predictive performance was evaluated using four primary statistical metrics: RMSE, MAE, R^2, and Pearson correlation.

3.11.1 Root Mean Square Error

$$MSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}$$

where y_i represents the observed Dst value and $\hat{y}_i$ represents the predicted value. RMSE penalizes larger forecasting errors more strongly and was therefore used as the primary error metric.

3.11.2 Mean Absolute Error

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i|$$

MAE provides the average magnitude of prediction error without applying the squared-error penalty.

3.11.3 Coefficient of Determination

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}}$$

where SS_{res} represents the residual sum of squares and SS_{tot} represents the total sum of squares.

3.11.4 Pearson Correlation

Pearson correlation was employed to determine the linear relationship between the observed and forecasted Dst index values. The correlation provides additional information besides the RMSE and MAE, showing the degree of similarity in the time variations of the forecasted series to the observed series. The application of RMSE, MAE, R^2, and Pearson correlation aligns with the methodology of the manuscript.

4. Results and Discussion

4.1 Overall Deep-Learning Results

The current study employs the above four models to forecast the Dst-index using the same chronological data split, preprocessing method, input encoding approach, and testing criteria. This dataset consists of 219,168 observations, which are valid after preprocessing. The split consists of 70% for training, 15% for validation, and 15% for testing in chronological order. Sequence tensors consist of 48 historical time steps and eight input features.

To increase reproducibility and decrease dependency on a single random initialization, each of these four deep learning architectures was analyzed through five independent random seeds. For each seed, RMSE, MAE, R^2, and Pearson correlation were obtained. The mean and standard deviation of these values from the five runs were computed.

The results from the five-seed runs show significant differences between the four models. LSTM achieved the lowest mean RMSE of 6.3698 ± 4.8349, followed by TCN with 7.8052 ± 3.2206, BiLSTM with 8.5720 ± 6.6232, and CNN-LSTM with 11.4037 ± 4.3021. The corresponding mean MAE values were 5.1611, 5.2896, 6.7341, and 9.1620, respectively.

Table 5. Five-seed deep-learning performance

Model	RMSE	MAE	R^2	Correlation	Parameters
LSTM	6.3698 ± 4.8349	5.1611 ± 4.8930	0.8725 ± 0.1972	0.9785 ± 0.0112	20,801
TCN	7.8052 ± 3.2206	5.2896 ± 2.1695	0.8511 ± 0.1376	0.9276 ± 0.0694	66,369
BiLSTM	8.5720 ± 6.6232	6.7341 ± 5.7431	0.7664 ± 0.2826	0.9356 ± 0.0678	41,537
CNN-LSTM	11.4037 ± 4.3021	9.1620 ± 3.7657	0.6883 ± 0.2248	0.9175 ± 0.0538	105,601

From the results obtained, it is clear that LSTM performs better than all the other three deep learning models considered in terms of prediction performance using the experiment setup. It has the minimum average RMSE and MAE while giving the maximum average R^2 and correlation. However, the large standard deviation observed, especially in LSTM and BiLSTM, indicates that the performance varies depending on the random seed. Consequently, the mean values must be understood in conjunction with the standard deviations and not necessarily as deterministic model attributes.

The results obtained from different random seeds further support the varying

performance across different runs. For instance, while LSTM yields RMSE results ranging between 3.74 and 14.93 across the five different seeds used, BiLSTM yields results ranging between 3.70 and 16.83.

4.2 Effect of Hyperparameter Optimization

Hyperparameter optimization was performed using two candidate configurations for each deep-learning architecture. For LSTM and BiLSTM, the search varied the number of recurrent units, dropout rate, and learning rate. CNN-LSTM additionally varied convolutional filters and kernel size, while TCN varied its convolutional filters, kernel size, dropout, and learning rate.

The selected LSTM configuration consisted of 64 units, 0.20 dropout, and a learning rate of 0.001. The corresponding BiLSTM configuration also used 64 units, 0.20 dropout, and a learning rate of 0.001. For CNN-LSTM, the selected configuration used 128 recurrent units, 64 convolutional filters, kernel size 5, dropout 0.30, and learning rate 0.0005. For TCN, the selected configuration used 64 filters, kernel size 5, dropout 0.30, and learning rate 0.0005.

The optimization results demonstrate that the selected configurations were not arbitrarily chosen. For example, the first LSTM trial produced a validation MSE of approximately 0.000033, compared with approximately 0.000684 for its second candidate configuration. Similarly, the first BiLSTM configuration achieved a lower validation MSE than its alternative. CNN-LSTM and TCN also showed substantial differences between candidate configurations.

These findings indicate that hyperparameter selection can substantially influence the observed predictive performance and justify the use of an explicit tuning stage before the final multi-seed evaluation.

4.3 Comparison with Conventional Machine-Learning Baselines

To determine whether recurrent and temporal deep-learning architectures provide an advantage over conventional approaches, Random Forest, XGBoost, and LightGBM were evaluated using the same prepared dataset. A persistence baseline was also included.

The persistence model achieved an RMSE of 4.6334, an MAE of 2.9174, an R^2 of 0.9538, and a correlation of 0.9769.

The conventional machine-learning results obtained in the experiment are summarized in table 6.

Table 6. Comparison with conventional machine-learning baselines

Model	RMSE	MAE	R^2	Correlation	Training Time (s)	Inference (ms/sample)
Persistence	4.6334	2.9174	0.9538	0.9769	—	—
Random Forest	3.8999	2.5298	0.9673	0.9836	2702.09	0.0156
XGBoost	4.0183	2.4555	0.9626	0.9826	10.04	0.0121
LightGBM	3.7746	2.4225	0.9693	0.9846	62.02	0.0134

Among the conventional baselines, LightGBM achieved the lowest RMSE (3.7746), lowest MAE (2.4225), highest R^2 (0.9693), and highest correlation (0.9846). Random Forest also performed strongly, whereas XGBoost produced slightly higher RMSE than both Random Forest and LightGBM.

An important observation is that the persistence baseline itself produced a relatively strong result, with an RMSE of 4.6334 and R^2 of 0.9538. This indicates that Dst exhibits strong temporal persistence and provides an important reference point for interpreting the improvements obtained by the machine-learning models.

Therefore, the results demonstrate that conventional tree-based models should not be omitted from the benchmarking analysis. In fact, under the present experimental configuration, LightGBM and Random Forest achieved lower errors than the mean five-seed deep-learning results. This supports the interpretation of the study as a comparative benchmark rather than a claim that deep learning is universally superior.

4.4 Residual and Prediction Analysis

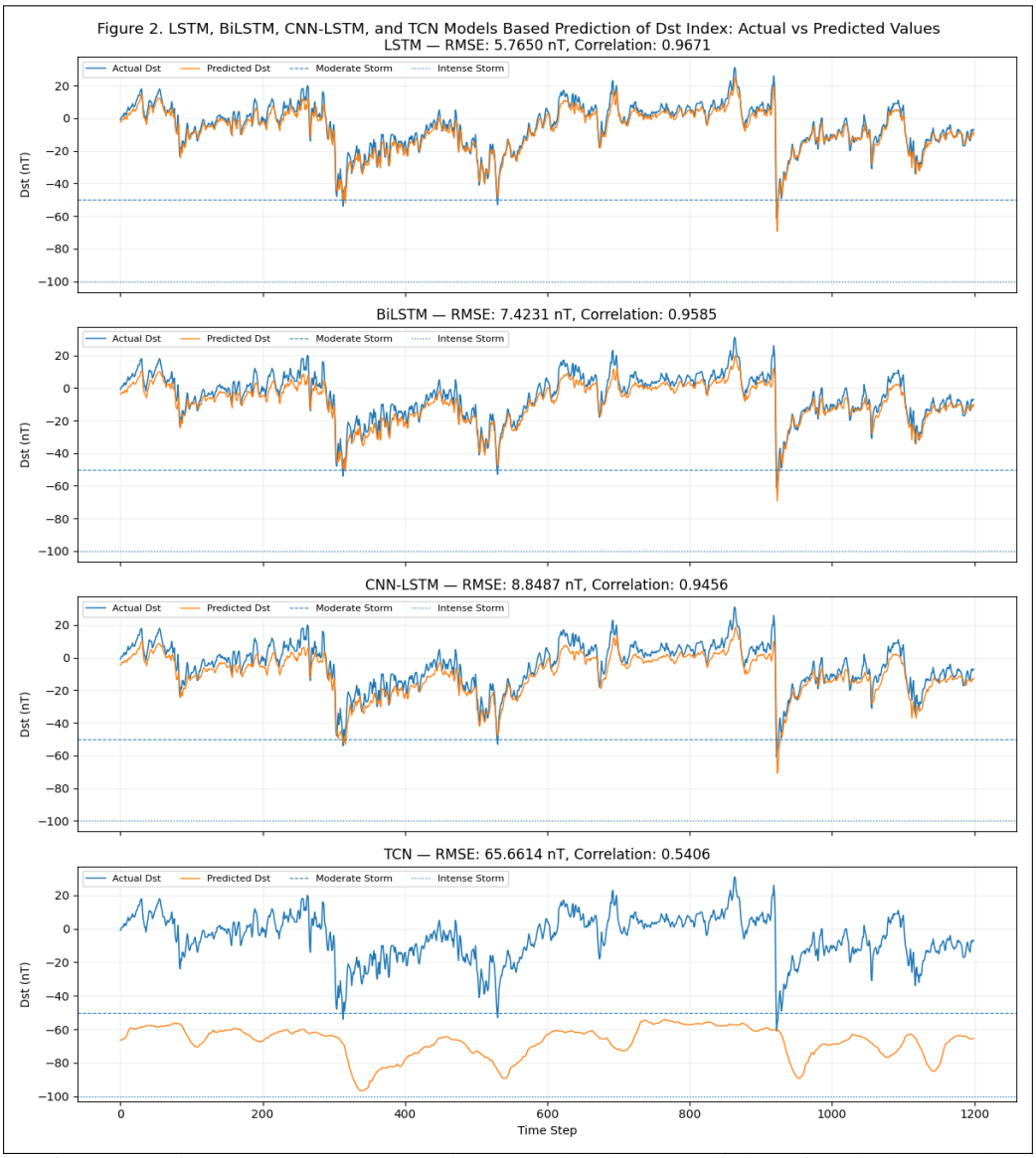

Figure 1. LSTM, BiLSTM, CNN-LSTM, and TCN models-based prediction of Dst index: actual versus predicted values

The prediction and residual analyses provide additional information beyond the aggregate numerical metrics. Figure 1 presents the actual and predicted Dst sequences obtained from the four deep-learning architectures. The LSTM and BiLSTM models generally follow

the temporal evolution of the observed Dst index, including the major negative excursions associated with geomagnetic disturbances. CNN-LSTM also captures the overall temporal pattern but exhibits larger deviations in several regions. The TCN prediction shows substantially larger deviations from the observed Dst series in the evaluated run.

Note: Figure 1 shows the prediction results from one representative evaluation run. The RMSE and correlation values displayed in the individual panels correspond to this single run and are not the five-seed mean ± SD values reported in Table 6. Because the seed identifier for this previously generated figure was not retained, the figure is presented only as a qualitative illustration of model prediction behavior.

The residual behavior of the four deep-learning models is further examined in Figure 2. The residuals of LSTM and BiLSTM are predominantly concentrated around zero, indicating relatively small systematic deviations for a substantial portion of the observations. CNN-LSTM exhibits a comparatively wider residual distribution, whereas TCN shows substantially larger systematic deviations. These differences are consistent with the relative prediction behavior observed in Figure 1.

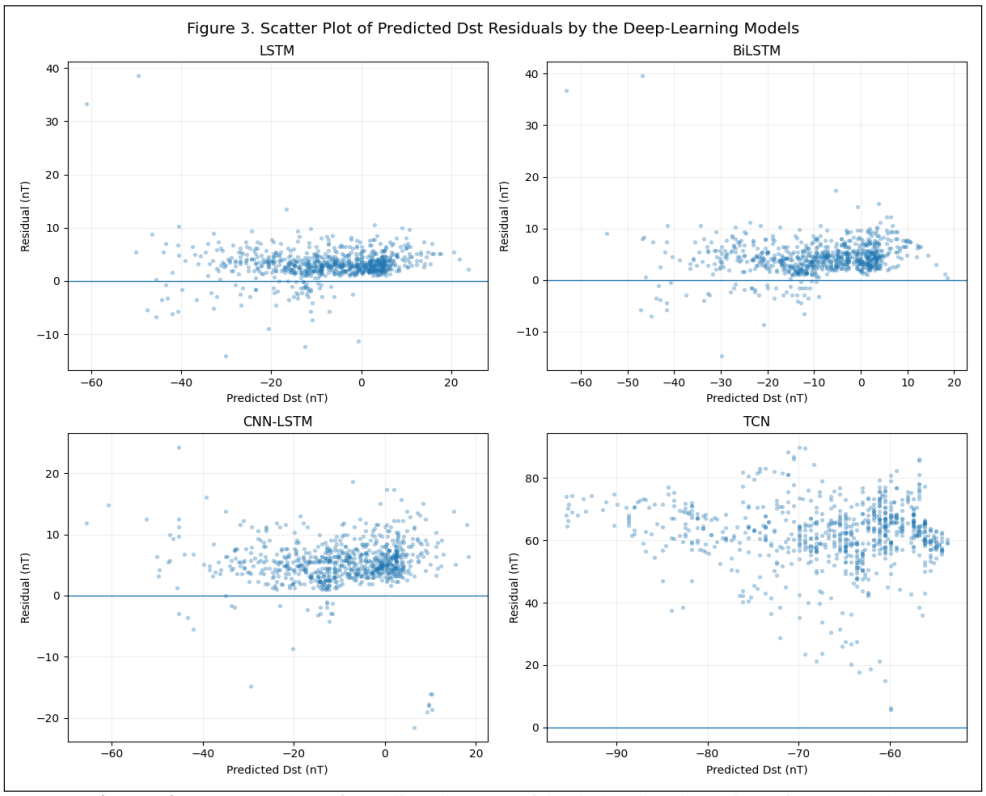

Figure 2. Scatter plot of predicted Dst residuals by the deep-learning models

The prediction and residual plots provide qualitative evidence of increased prediction errors during rare and extreme geomagnetic disturbances. These observations indicate that overall metrics alone may not fully characterize model behavior during storm-time conditions. In particular, a model can obtain a favorable overall RMSE while still exhibiting substantial errors for rare storm-time observations. Therefore, the prediction plots and residual distributions should be considered together with the numerical metrics.

The five-seed results reported in Table 5 provide the primary quantitative comparison,

while Figures 2 and 3 provide complementary visual evidence of prediction behavior and error distribution. Importantly, the results should not be interpreted as demonstrating statistically significant superiority of one deep-learning architecture over all others, since the observed differences among the five-seed results did not reach the conventional significance threshold.

4.5 Training and Validation Analysis

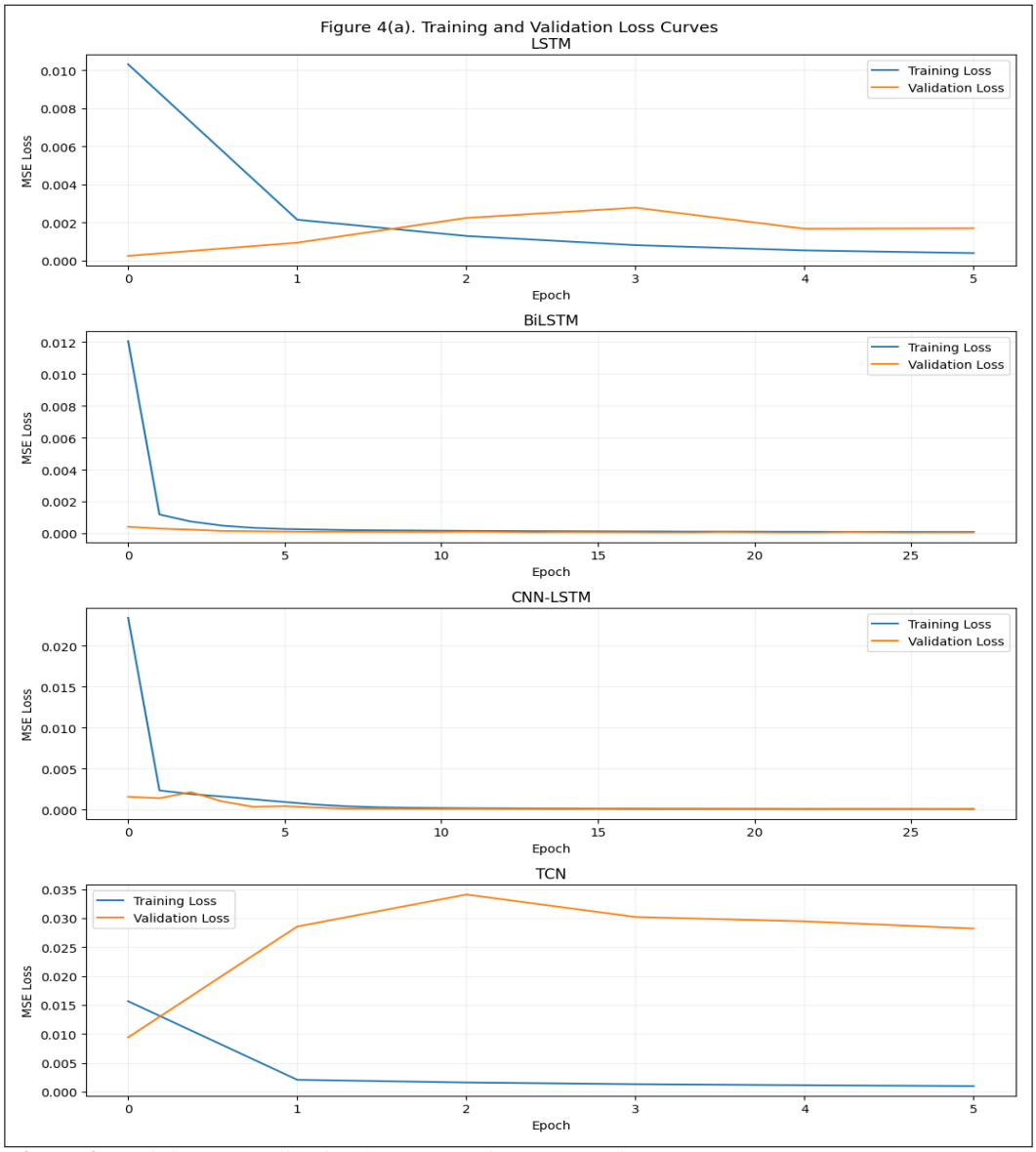

Figure 3. Training and validation loss curves for LSTM, BiLSTM, CNN-LSTM, and TCN models

Figure 3 presents the training and validation loss curves for the LSTM, BiLSTM, CNN-LSTM, and TCN architectures. The LSTM model shows a rapid reduction in training loss during the initial epochs, followed by stabilization, while the validation loss remains comparatively higher after the early epochs. BiLSTM and CNN-LSTM exhibit rapid convergence, with their training and validation losses approaching relatively small values as training progresses. In contrast, the TCN model shows a clear separation between training and validation loss, with the validation loss remaining substantially higher throughout training. The separation between training and validation loss indicates a larger generalization gap for TCN under this training configuration; however, its five-seed test performance remains competitive

with the other evaluated deep-learning architectures. Overall, the learning curves indicate that the recurrent architectures, particularly LSTM and BiLSTM, achieve more favorable convergence behavior under the adopted experimental configuration.

4.6 Computational Characteristics

Computational characteristics were evaluated using the measured number of trainable parameters, training time, and per-sample inference time. The LSTM contains 20,801 trainable parameters, compared with 41,537 for BiLSTM, 66,369 for TCN, and 105,601 for CNN-LSTM. The corresponding mean training times were approximately 88.98 s, 99.51 s, 107.98 s, and 55.62 s, respectively, while the measured mean inference times were approximately 0.0159, 0.0227, 0.0405, and 0.0191 ms/sample. These measurements demonstrate that computational characteristics differ substantially across architectures. LSTM has the smallest parameter count and a low measured inference time, whereas CNN-LSTM has the largest parameter count. TCN also requires more parameters than LSTM and has the highest measured mean inference time among the four deep-learning models in this experiment.

Therefore, the results do not support an unconditional statement that TCN is computationally more efficient or inherently suitable for real-time applications. Instead, computational suitability should be discussed in terms of the experimentally measured parameter count, training cost, and inference latency.

The present computational analysis is limited to trainable parameter count, measured training time, and per-sample inference latency. Memory consumption, GPU/CPU utilization, energy consumption, hardware-specific throughput, and scalability under operational data streams were not directly measured. Consequently, the reported inference measurements should not be interpreted as complete evidence of deployment readiness or energy efficiency. Such measurements require controlled hardware-specific profiling and are therefore identified as an important direction for future operational evaluation.

4.7 Statistical Variability and Significance Analysis

The five-seed evaluation provides an empirical estimate of model variability. The standard deviations demonstrate that the performance of some architectures is sensitive to random initialization. In particular, BiLSTM exhibits an RMSE standard deviation of 6.6232, while LSTM exhibits a standard deviation of 4.8349. Pairwise exploratory paired comparisons of the five seed-level RMSE values were also conducted using LSTM as the reference. The resulting p-values were approximately 0.625 for LSTM versus BiLSTM, 0.238 for LSTM versus CNN-LSTM, and 0.654 for LSTM versus TCN. Thus, none of these comparisons reached the conventional $\alpha = 0.05$ threshold.

Consequently, although LSTM achieved the best average deep-learning performance, the observed differences among the deep-learning architectures should not be interpreted as statistically significant superiority based on only five repeated runs. The results are more appropriately described as comparative evidence under the adopted experimental configuration.

4.8 Ablation Study

A systematic ablation study was performed in table 7. To investigate the contributions of the input variables and temporal configuration. Five configurations were evaluated using five independent seeds: the full eight-feature 48-hour configuration, removal of historical

Dst, removal of AE and Kp10, solar-wind variables alone, and a reduced 24-hour temporal window. Ablation-study error metrics are reported in normalized Dst units because the ablation experiments were evaluated using the scaled target representation. These values should therefore not be directly compared with the nT-scale RMSE values reported in Table 6.

Table 7. Ablation-study results

Configuration	RMSE	MAE	R^2	Correlation	ΔRMSE
Full 8 features + 48 h	0.008568 ± 0.001790	0.005927 ± 0.001252	0.959291 ± 0.018615	0.980767 ± 0.003310	—
Without Dst history	0.007893 ± 0.000171	0.005416 ± 0.000242	0.968610 ± 0.001456	0.984917 ± 0.000987	−7.89%
Without AE + Kp10	0.024917 ± 0.000236	0.018540 ± 0.000256	0.867322 ± 0.006309	0.857761 ± 0.001712	+190.80%
Solar-wind only	0.034423 ± 0.006973	0.024409 ± 0.003111	0.344243 ± 0.281043	0.708473 ± 0.110743	+301.75%
Full features + 24 h	0.007798 ± 0.000217	0.005311 ± 0.000149	0.967398 ± 0.001810	0.984234 ± 0.000929	−8.99%

The ablation results demonstrate that input-variable composition and temporal configuration have a substantial effect on Dst prediction. Removing AE and Kp10 produces a marked deterioration in performance, increasing the normalized RMSE from 0.008568 for the full 48-hour configuration to 0.024917. The solar-wind-only configuration produces the largest error, with an RMSE of 0.034423, highlighting the importance of geomagnetic-index information. In contrast, removing historical Dst improves the normalized RMSE by 7.89%, from 0.008568 to 0.007893. Similarly, reducing the temporal window from 48 h to 24 h produces the lowest RMSE of 0.007798, representing an 8.99% improvement relative to the full 48-hour configuration. These results indicate that longer temporal context does not necessarily improve performance under the adopted normalized ablation setting.

4.9 Overall Discussion

The experimental findings provide several important conclusions regarding Dst forecasting. First, LSTM provides the strongest average performance among the evaluated deep-learning architectures, with the lowest mean RMSE and MAE and the highest mean R^2 and correlation. However, the relatively large seed-to-seed variation indicates that model performance is sensitive to initialization, emphasizing the importance of repeated evaluation. Second, the comparison with conventional machine-learning models demonstrates that deep-learning architectures should not automatically be considered superior. LightGBM achieved an RMSE of 3.7746, compared with 3.8999 for Random Forest and 4.0183 for XGBoost. These results were also better than the persistence baseline of 4.6334. Consequently, the principal contribution of the experimental study is the comparative benchmarking of alternative forecasting approaches, rather than establishing that one model family is universally optimal.

Thirdly, the ablation analysis shows that geomagnetic indices such as AE and Kp10 play an important role in the quality of prediction, as their removal led to a significant growth of the error, whereas the solar-wind-only configuration performed much worse. This suggests that the combination of solar-wind and geomagnetic data is more informative than using only solar-wind variables for the particular forecasting task under consideration. Finally, the computational

metrics clearly distinguish models in terms of architecture size and computational cost. For instance, LSTM represents a relatively small architecture consisting of 20,801 parameters, while CNN-LSTM contains more than five times as many parameters. Thus, these metrics give an objective basis for discussing computational aspects without unfounded claims about universal real-time capability. In conclusion, it can be stated that the experiment results support the idea of employing the comparative benchmarking approach. These results show that model selection, input variable composition, context window, and computational characteristics affect the performance of Dst forecasters. Therefore, the findings give a more thorough assessment than single model accuracy comparison and serve as an experiment-based ground for selecting forecaster architecture on a particular dataset and evaluation protocol.

5. Limitations and Future Work

Several limitations can be identified in this study as a basis for future research. Firstly, the experiments were performed on a single dataset with hourly resolution from 2000 to 2024; therefore so the evaluation on new independent data from another period is required to increase generalizability. Secondly, in primary experiments, the look-back period was equal to 48 hours, but the number of different configurations of temporal variables was rather small. In future studies, different forecasting horizons and more extensive optimization of temporal windows can be considered. Thirdly, despite validation-based hyperparameter screening and evaluation on five different seeds were applied, the hyperparameter search was limited to only two candidates per model, so this search should not be regarded as an optimization process in its fullest meaning. In the current study, the chronological train/validation/test holdout approach was used, whereas future studies should use rolling-origin or expanding-window cross-validation.

Moreover, ProtonFlux was executed from our analysis due to a very high percentage of missing data, and the ablation analysis did not consider every single combination of features and architecture. Future studies could investigate better techniques for handling missing data, probabilistic and uncertainty-based forecasting, attention- and transformer-based architectures, physics-driven models, and specific assessment of extreme storms. While chronological splits of training, validation, and testing sets were made to avoid temporal leakage, there was no use of rolling-origin or expanding-window cross-validation was not used in our experiment. This is acknowledged as a drawback of the current design, and it will be addressed in the future.

6. Conclusion

This study presented a comparative evaluation of LSTM, BiLSTM, CNN-LSTM, and TCN models for Dst-index forecasting using solar-wind and geomagnetic observations. The methodology incorporated chronological data partitioning, hyperparameter optimization, five-seed evaluation, conventional machine-learning baselines, computational benchmarking, and ablation analysis. Among the evaluated deep-learning architectures, LSTM achieved the best average performance; however, LightGBM and Random Forest provided lower RMSE than the mean deep-learning results. The observed variability across seeds and the statistical comparisons indicate that the differences among the deep-learning models should not be interpreted as statistically significant superiority. Conventional models also performed strongly, with LightGBM achieving RMSE 3.7746, MAE 2.4225, R^2 0.9693, and correlation 0.9846, demonstrating the importance of including non-deep-learning baselines. The ablation analysis showed that feature composition substantially affects forecasting performance.

Removing AE and Kp10 increased RMSE from 0.008568 to 0.024917, while using only solar-wind variables increased it to 0.034423. The 24-hour configuration achieved an RMSE of 0.007798, indicating that longer temporal context does not necessarily improve performance. Overall, the primary contribution of this study is a controlled and reproducible benchmarking framework for Dst forecasting rather than a new neural-network architecture.

References

[1] Bhandari, Prajwal, Kiran Bagale, and Bidur Nepal. ”Seasonal Variation of Solar Wind–Geomagnetic Coupling: A Correlation Heatmap Approach.” SS Multidisciplinary Research Journal 2, no. 1 (2025): 117-126.

[2] Boteler, D. H. ”A New Versatile Method for Modelling Geomagnetic Induction in Pipelines.” Geophysical Journal International 193, no. 1 (2013): 98-109.

[3] Silwal, Ashok, Sujan Prasad Gautam, Prakash Poudel, Monika Karki, Narayan P. Chapagain, and Binod Adhikari. ”Variation of Total Electron Content Over Nepal During Geomagnetic Storms: GPS Observations.” Russian Journal of Earth Sciences 23, no. 3 (2023): 1-19.

[4] Abduallah, Yasser, Jason TL Wang, Prianka Bose, Genwei Zhang, Firas Gerges, and Haimin Wang. ”A Deep Learning Approach to Dst Index Prediction.” arXiv preprint arXiv:2205.02447 (2022).

[5] Echer, E., and W. D. Gonzalez. ”Geoeffectiveness of Interplanetary Shocks, Magnetic Clouds, Sector Boundary Crossings and Their Combined Occurrence.” Geophysical research letters 31, no. 9 (2004).

[6] M. Cristoforetti M Battiston Roberto and Gobbi A and Iuppa Roberto and Piersanti, ”Prominence of the Training Data Preparation in Geomagnetic Storm Prediction Using Deep Neural Networks,” Scientific Reports, vol. 12, 2022, 7631.

[7] Burton, Rande K., R. L. McPherron, and C. T. Russell. ”An Empirical Relationship between Interplanetary Conditions and Dst.” Journal of geophysical research 80, no. 31 (1975): 4204-4214.

[8] Akasofu, S-I. ”Energy Coupling Between the Solar Wind and the Magnetosphere.” Space Science Reviews 28, no. 2 (1981): 121-190.

[9] Newell, P. T., T. Sotirelis, K. Liou, C-I. Meng, and F. J. Rich. ”A Nearly Universal Solar Wind-Magnetosphere Coupling Function Inferred from 10 Magnetospheric State Variables.” Journal of Geophysical Research: Space Physics 112, no. A1 (2007).

[10] Kelley, Michael C. The Earth's Ionosphere: Plasma Physics and Electrodynamics. Vol. 96. Academic press, 2009.

[11] LI, Shaowen, Jun NIU, Bing MEI, Lizhu YAO, and Yanbin LI. ”Dst Index Prediction Method Based on LSTM Neural Network.” Chinese Journal of Space Science 45, no. 3 (2025): 641-652.

[12] Jang, Jun, Lang Xu, Chaoqian Xu, and Liang Zhang. ”Forecasting Single-Station Ionospheric TEC Over China Using a Combined DBO-LSTM Model During Geomagnetic Storms.” Advances in Space Research 75, no. 10 (2025): 7684-7695.

[13] Tang, Rongxin, Fantao Zeng, Zhou Chen, Jing-Song Wang, Chun-Ming Huang, and Zhiping Wu. ”The Comparison of Predicting Storm-Time Ionospheric TEC by Three Methods: ARIMA, LSTM, and Seq2Seq.” Atmosphere 11, no. 4 (2020): 316.

[14] S. J. Hochreiter, ”Long Short-Term Memory,” Neural Computation, vol. 9, 1997., 1735-1780. https://doi.org/10.1162/neco.1997.9.8.1735

[15] Subramanian, Dhanalakshmi, Poongothai Marimuthu, and Nithyalakshmi Ramadoss. 2026. “A Multifaceted Framework for Predicting Ambient Air Pollution in the National Capital Region”. Journal of Trends in Computer Science and Smart Technology 8 (3): 558-78. https://doi.org/10.36548/jtcsst.2026.3.007.

[16] Sarkar, Abhijeet, Ashmita Saha, Bhaskar Bhattacharjee, Bishnupada Sahoo, Rounak Bakshi, and Soumik Podder. 2023. “Nanozymes Induced Air Purification- A State of the Art Review”. Recent Research Reviews Journal 2 (1): 11-26.https://doi.org/10.36548/rrrj.2023.1.02