https://doi.org/10.33271/nvngu/2025-5/122

Trong Gia Nguyen[1,2],
orcid.org/0009-0006-0765-245X,
Tinh Duc Le[1],
orcid.org/0000-0002-0022-3453,
Long Quoc Nguyen[*1,3],
orcid.org/0000-0002-4792-3684

1 – Hanoi University of Mining and Geology, Faculty of Geomatics and Land Administration, Hanoi, Socialist Republic of Vietnam
2 – Hanoi University of Mining and Geology, Geodesy and Environment Research Group, Hanoi, Socialist Republic of Vietnam
3 – Hanoi University of Mining and Geology, Innovations for Sustainable and Responsible Mining (ISRM) Research Group, Hanoi, Socialist Republic of Vietnam
* Corresponding author e-mail: nguyenquoclong@humg.edu.vn

EMPIRICAL COMPARISON OF FIVE DEEP LEARNING ARCHITECTURES FOR GNSS TIME-SERIES DATA FORECASTING

Purpose. This study aims to propose an effective deep learning model for analyzing time series of GNSS data.

Methodology. Five deep learning models were investigated using the Adam optimization algorithm and the MSE loss function: CNN (Conv1D), GRU, Transformer, Seq2seq, and SimpleRNN. The models' predictive performance was evaluated with data from three CORS stations in Vietnam: HYEN, BTRI, and CTHO. The models were constructed using the Adam optimization function, MSE loss function, and a batch size of 16.

Findings. The prediction results across all models were generally low. Specifically, the Seq2seq model achieved the lowest RMSE value for the Up-component data from the HYEN station. The Transformer model was unsuitable for analyzing GNSS data over time, and the SimpleRNN model showed poor performance with the HYEN station data. However, the proposed CNN(Conv1D)-RF model, which integrates the RF model with the CNN(Conv1D) model, achieved high predictive performance, with RMSE = 0.4 mm, MAE = 0.07 mm, and R-squared = 99.8 % when the input data consisted solely of the Up-component. When the input data included all three components (N, E, and Up), the maximum RMSE value was 0.92 mm.

Originality. This study introduces the CNN(Conv1D)-RF model, a novel integration of the RF model with the CNN(Conv1D) model, for GNSS data analysis.

Practical value. The proposed model demonstrates promising predictive performance, indicating its potential application in accurately analyzing GNSS data for various practical purposes.

Keywords: *deep learning, GNSS time series, machine learning, algorithm*

Introduction. Data from Continuously Operating Reference Stations (CORS) play a crucial role in determining vertical displacements and land subsidence [1]. Results obtained from CORS data allow for the monitoring of geological changes over time. Studies indicate that utilizing CORS data enhances the accuracy of assessing vertical displacements due to factors such as groundwater extraction, construction activities, and climate impacts [2, 3]. Furthermore, analyzing data from CORS stations aids in predicting and managing risks associated with surface subsidence, particularly in urban areas [4]. Therefore, most countries worldwide have implemented CORS networks to protect resources and mitigate damage [5].

Convolutional Neural Networks (CNN) have revolutionized computer vision by employing convolutional layers, nonlinear units, and sampling layers, proving highly effective in tasks such as facial recognition and autonomous driving systems. Variants like 1D-CNN are critical for time series prediction and signal recognition, contributing to advancements in various fields [6, 7].

Gated Recurrent Units (GRU), a variant of Recurrent Neural Networks (RNN), are widely used for processing irregular time series data. GRUs effectively learn and model complex temporal dependencies, facilitating efficient predictions and handling missing data across numerous fields such as healthcare, activity recognition, and environmental monitoring [8].

The Transformer architecture is extensively applied in natural language processing and computer vision [9]. Recently, it has been adapted for time series analysis, leveraging self-attention mechanisms, positional encoding, and encoder-decoder architectures [10].

Seq2Seq is a deep learning model developed to forecast long time series, improving long-term time series forecasting (LSTF) through encoder-decoder structures to handle complex sequence data and long-range dependencies [11].

SimpleRNN is a basic recurrent neural network model, effective in analyzing time series data. Its primary advantage lies in its ability to capture short-term dependencies in sequence data, enabling faster processing with lower computational resource requirements compared to more complex models like LSTM and GRU [12, 13].

Machine learning has enhanced GNSS data processing capabilities, particularly in improving accuracy and reliability in weak signal environments [14]. LSTM-based RNN models have successfully eliminated nearly 50 % of noise from GNSS data, achieving coordinate prediction accuracy with a mean error of 1.1 mm [15].

Eight different machine learning algorithms have been applied to predict changes in land surface elevation based on GNSS position time series. Models such as multilayer neural networks, Bayesian networks, Gaussian Processes (GP), and five other models have been utilized for change prediction. Results indicate that the Gaussian Processes algorithm outperformed others, achieving high accuracy of up to 4 mm and averaging 2 cm across the entire data

series [16]. A radial basis function, with 200 center points and a least squares method in batch mode, was applied for optimal results in time series point position analysis [17]. Generalized regression neural networks have been used to predict GNSS position time series, yielding accuracy up to 6 cm for discrete data and 3 cm for continuous data, outperforming the Theta method [18].

Multiscale sliding windows (MSSW) have been employed in conjunction with LSTM networks, resulting in a reduction of RMSE from 2.1 to 23.7 % and MAE from 1.6 to 22.2 % compared to traditional LSTM models [19].

The Quantum-inspired Long Short-Term Memory model has been applied to develop satellite clock products for the BDS system, achieving an improvement in short-term prediction accuracy of up to 80.4 % compared to traditional models like ARIMA [20]. The RF model has been implemented to detect disruptions in GNSS position time series due to earthquakes, yielding a best performance with an F-score of 0.77 [21].

Temporal Convolutional Networks (TCN) have been utilized to correlate meteorological parameters with GNSS height time series. Results indicate that TCN reduced RMS error by 28.6 %, nearly matching physical models [22].

The DVMD-LSTM model has been applied to predict GNSS time series data, achieving a reduction in RMSE of 9.86 % and an increase in R^2 of 17.97 % compared to the VMD-LSTM model [23].

Models such as GBDT, XGBoost, and RF have been employed to interpolate GNSS coordinate time series, with results demonstrating a 45 % higher interpolation efficiency compared to traditional spline methods, with GBDT and RF outperforming XGBoost [24].

GNSS data used to determine the deformation of irrigation works has been analyzed using a combination of CNN and GRU models, achieving an accuracy improvement of over 45 % [25].

The objective of this paper is to evaluate the performance of five deep learning models- CNN (Conv1D), GRU, Transformer, Seq2Seq, and SimpleRNN – utilizing the Adam optimization algorithm and the MSE loss function to predict GNSS data in time series. Additionally, the performance of the CNN (Conv1D)-MAF model, which combines the CNN (Conv1D) model with a moving average filter, will also be assessed, along with substitutions of the Adam optimization algorithm with the SGD optimization algorithm and the MSE loss function with the Huber loss function.

Data and methodology. ***Data.*** The data used in this study is obtained from three CORS stations (Table 1) from the VNGEONET network established by the Vietnam Department of Surveying, Mapping, and Geographic Information (Fig. 1).

Fig. 1. Locations of the three CORS stations: HYEN, BTRI, CTHO

The data collected at the three CORS stations were obtained over the period from August 10, 2019, to March 18, 2022. Detailed information about the stations is provided in Table 1.

Methodology. The research methodology used in this study is illustrated in Fig. 2.

The data obtained from the CORS stations is converted from raw format to RINEX format for analysis using Gamit/Globk [26], a high-precision GNSS processing software. The processed data is used to work with the following deep learning models to analyze their usage in predicting the data gaps in time series:

a. CNN(Conv1D) model. The CNN (Conv1D) model in convolutional neural networks is widely used to process one-dimensional sequential data, such as time series data. This model performs convolution between the input and filters to extract features from the data sequence. The mathematical formula for the CNN (Conv1D) model is

$$y(t) = \sum_{i=1}^{k} x(t+i) \cdot w(i), \qquad (1)$$

where $x(t)$ is the input data; $w(i)$ is the weight of the filter; k is the kernel size.

Table 1

Information on GNSS CORS measurement data

Station name	Time		Receiver type	Antenna type	Interval (second)
	First epoch	Last epoch			
HYEN	2019/08/10	2022/03/18	LEICA GR50	LEIAR25.R4 LEIT	30
BTRI					
CTHO					

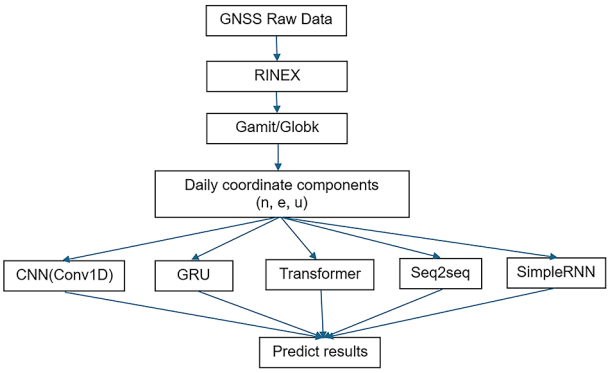

Fig. 2. Prediction Process with Deep Learning Models: CNN(Conv1D), GRU, Transformer, Seq2seq, SimpleRNN

The output of the CNN (Conv1D) is fed into layers such as Flatten to convert it into a flat vector, followed by dense layers to generate the output. Conv1D is often combined with the ReLU activation function to enhance the non-linearity of the model.

The process of using the CNN (Conv1D) model to analyze data in this case is illustrated in Fig. 3.

b. GRU model. GRU is a variant of recurrent neural networks (RNN) designed to address the gradient vanishing problem in processing of the long time series data. This model utilizes an update gate (Eq. 2) and a reset gate (Eq. 3), with the hidden layer defined as follows

$$z_t = \sigma(W_z x_t + U_z h_{t-1}), \quad (2)$$

where σ as the sigmoid function ranging from 0 to 1; W_z is the weight matrix applied to the current input x_t, and U_z is the matrix applied to the previous hidden state h_{t-1}.

$$r_t = \sigma(W_r x_t + U_r h_{t-1}), \quad (3)$$

where W_r and U_r are the weight matrices for the current input x_t and the previous hidden state h_{t-1}.

The hidden layer represents the newly stored information after applying the gates, as given in Eq. (4)

$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot \tanh(W_h x_t + r_t \odot U_h h_{t-1}, \quad (4)$$

where ⊙ represents element-wise multiplication; tanh is the hyperbolic tangent function that limits the output between –1 and 1; W_h and U_h are the weight matrices for the input and the previous hidden state. Fig. 4 is a simplistic illustration of implementing GRU in this case.

c. Transformer model. The Transformer is a deep learning model that efficiently processes sequential data and is widely used in natural language processing tasks. The attention mechanism has been implemented in the

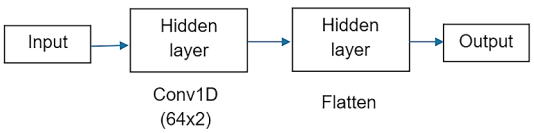

Fig. 3. Flowchart of the GNSS time series data analysis process using the CNN (Conv1D) function

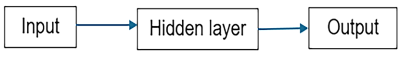

Fig. 4. Diagram of the GNSS time series data analysis process using the GRU

Transformer model to focus on different parts of the input simultaneously. The mathematical formulation of this model is given in Eq. (5)

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V, \quad (5)$$

where Q, K, and V are the query, key, and value matrices. After the attention mechanism, a feed-forward neural network processes the output.

Fig. 5 is an illustration of the process in implementing the Transformer model.

d. Seq2seq model. Seq2seq is a bi-RNN architecture designed for processing sequential data. The model consists of two RNNs: the encoder and the decoder. The Seq2seq model as applied in this case is as follows (6, 7)

$$Encoder = LSTM(100, return\_sequences = \\ = True)(input\_layer). \quad (6)$$

Then, the output of the encoder is passed to the decoder layer, which is also an LSTM with 100 units. The output result is then determined using

$$Output\_layer = Dense(1)(decoder). \quad (7)$$

Fig. 6 below describes the structure of the Seq2Seq model.

e. SimpleRNN model. The SimpleRNN model is a RNN structure primarily used for time series data processing tasks. This model has the ability to remember information from previous states due to its sequential structure. In SimpleRNN, each input is processed through a hidden unit, which can be described by Eq. (8)

$$h_t = \tanh(W_h h_{t-1} + W_x x_t + b), \quad (8)$$

where h_t is the hidden state at time t; W_h and W_x are the weight matrices; x_t is the input at time t; b is the bias. The SimpleRNN model can be expanded with multiple hidden units to improve its learning capacity.

Six statistical parameters are then used to evaluate the individual and inter-model performances of the five artificial intelligence architectures: Mean Square Error (MSE, Eq. (9)), Root Mean Square Error (RMSE, Eq. (10)), Mean Absolute Error (MAE, Eq. (11)), R-Squared (R^2, Eq. (12)), Cohen's Kappa (Eq. (13)), and F1-Score (Eq. (14)).

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2, \quad (9)$$

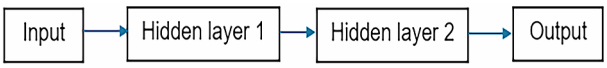

Fig. 5. Diagram of the GNSS data analysis process over time using the Transformer model

Fig. 6. The structure of the Seq2Seq model used

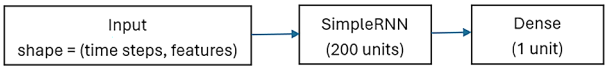

Fig. 7. The structure of the SimpleRNN model used

where y_i is the actual value; $\hat{y}_i$ is the predicted value, and n is the number of observations.

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}; \quad (10)$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|y_i - \hat{y}_i\right|; \quad (11)$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y})^2}{\sum_{i=1}^{n}(y_i - \bar{y}_i)^2}, \quad (12)$$

where $\bar{y}$ is the mean of actual values.

$$\kappa = \frac{P_o - P_e}{1 - P_e}, \quad (13)$$

where *Po* is the observed agreement among raters (or classification accuracy); P_e is the expected agreement by chance, calculated based on the distribution of the classes.

$$F1 - Score = 2 \cdot \frac{Precision.Recall}{Precision + Recall}, \quad (14)$$

where Precision is the ratio of true positive predictions to the total predicted positives, calculated as

$$Precision = \frac{TP}{TP + FP}, \quad (15)$$

and Recall (also known as Sensitivity) is the ratio of true positive predictions to the actual positives, calculated as

$$Recall = \frac{TP}{TP + FN}, \quad (16)$$

where *TP* is True Positive; *FP* is False Positive; *FN* is False Negative.

Results and discussion. ***Results of GNSS data processing using Gamit/Globk software.*** To process GNSS data using GAMIT/GLOBK, first of all RINEX data (observation and navigation files) is converted into the GAMIT-compatible format using *rinex2gamit* command. Then the precise ephemeris data (e.g., IGS orbits) are downloaded in specific directories and required input files are prepared, including the station information file (*.sittbl*). In this file, approximate coordinates of the stations are included as initial estimates. Once the data and directories are prepared, GAMIT is run using the command *gamit* to compute initial station positions/solutions (*.sol* file) and residuals (*.res* file). Once GAMIT is completed, GLOBK is used with the command *globk* to refine the results. It is used to combine or adjust data from multiple stations over time. For this step, precise coordinates of the IGS stations are used in the *.gk* file, to provide a reference frame for the analysis. The resulting output will consist of time-series data of station positions and displacement velocities. A detailed step-by-step methodology is explained in King and Bock. The results of the time-series of coordinates of the HYEN, BTRI, and CTHO stations obtained using GAMIT/GLOBK are shown in Figs. 8, 9, and 10, respectively.

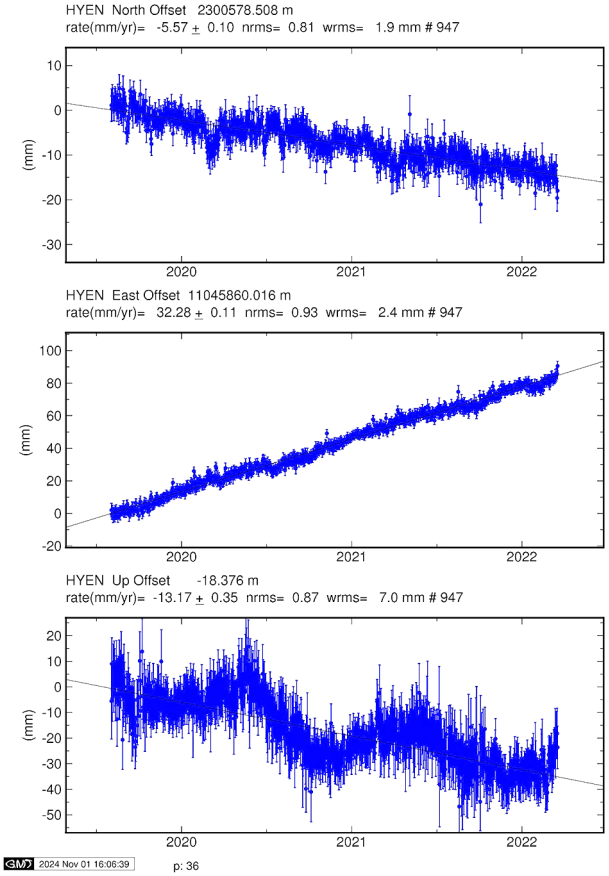

Fig. 8. Daily coordinate components of the HYEN station determined by Gamit/Globk software

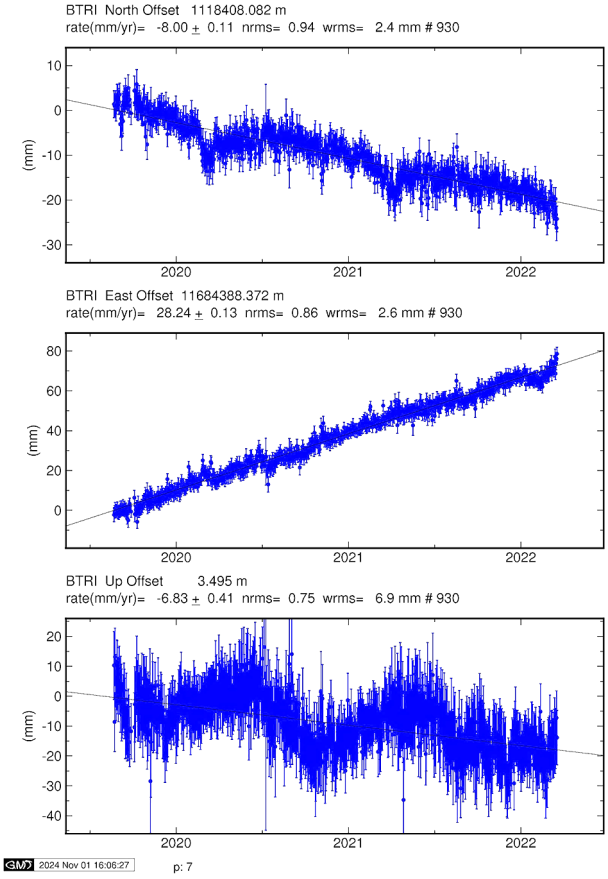

Fig. 9. Daily coordinate components of the BTRI station determined by Gamit/Globk software

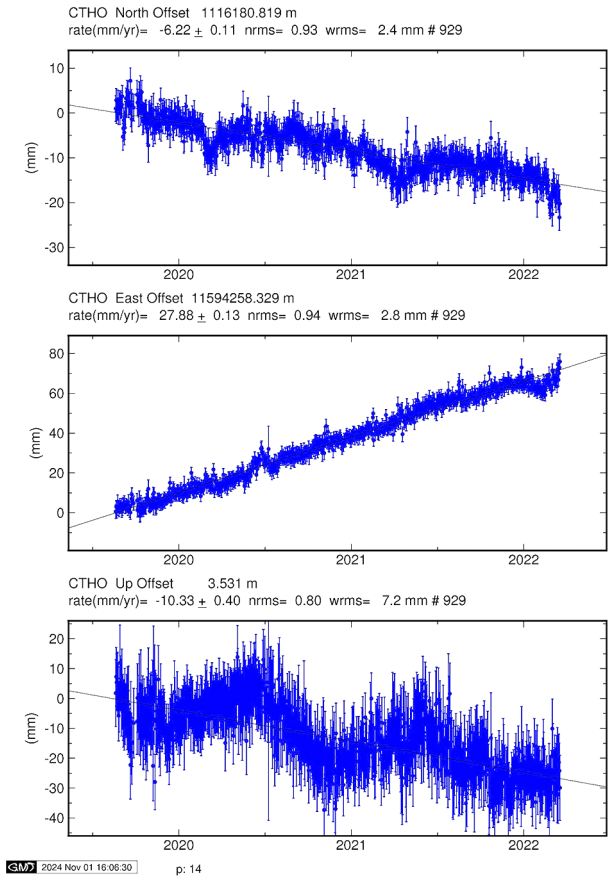

Fig. 10. Daily coordinate components of the CTHO station determined by Gamit/Globk software

Performance statistics of the model in the case of using the Adam optimization algorithm and the MSE loss function. In this case, all models use the Adam optimization algorithm and the MSE loss function; the data split ratio between the training dataset and the testing dataset is 60–40.

Adam combines the benefits of two popular optimizers, AdaGrad and RMSProp, making it adaptive to the learning rate. It effectively handles sparse gradients, converges quickly, and is less sensitive to hyperparameter tuning. However, Adam may sometimes lead to suboptimal generalization compared to other optimizers, particularly in cases with small datasets. Additionally, it can be computationally expensive due to the storage of multiple gradients [12].

The Mean Squared Error (MSE) loss function facilitates efficient convergence to minima for small errors, as the gradient diminishes progressively. It employs quadratic equations, which helps penalize models for outliers effectively. The squaring of values speeds up training; however, high loss values can lead to significant jumps during backpropagation, which is not desirable. MSE's heightened sensitivity to outliers means that substantial anomalies in the data can adversely affect model performance [27].

The data in Table 2 shows that while the SimpleRNN model demonstrates high predictive performance for data from the BTRI and CTHO stations, it is not suitable for the input data from the HYEN station, as indicated by a negative R^2 value.

The Transformer model is entirely unsuitable for predicting GNSS time series data in this case, as indicated by negative R^2 values for data from all three stations.

The Seq2seq and GRU models demonstrate very high predictive performance for data from the BTRI and CTHO stations, with maximum RMSE values of 0.82 and 0.07 mm, respectively. Additionally, the models show strong alignment with the input data (as indicated by R^2 values) reaching between 99 and 100 %. For data from the HYEN station, the Seq2seq model proves more effective than the GRU model, with an RMSE of 2.41 mm, MAE of 1.91 mm, and a model fit reaching 96 %.

The CNN (Conv1D) model demonstrates high predictive performance for data from the HYEN and CTHO stations, with maximum RMSE and MAE val-

Table 2

Accuracy evaluation results with input data being the u component

Model	Station	RMSE, mm	MSE, mm	MAE, mm	R-squared	F-score	Kapa
CNN(Conv1D)	BTRI	5.46	29.84	0.43	0.6254	1.0	1.0
	CTHO	0.70	0.49	0.59	0.9956	1.0	1.0
	HYEN	1.31	1.72	0.65	0.9895	1.0	1.0
GRU	BRTI	0.02	0.00	0.01	1.0	1.0	1.0
	CTHO	0.07	0.00	0.05	1.0	1.0	1.0
	HYEN	4.64	21.54	3.16	0.8682	1.0	1.0
Transformer	BRTI	12.43	154.57	10.71	–0.9400	0.9784	0.9569
	CTHO	41.64	1,733.95	40.81	–14.4483	0.8548	0.7097
	HYEN	261.97	68,628.35	248.33	–418.8853	0.7684	0.5368
Seq2seq	BRTI	0.82	0.68	0.77	0.9915	1.0	1.0
	CTHO	0.61	0.37	0.59	0.9967	1.0	1.0
	HYEN	2.41	5.85	1.91	0.9642	1.0	1.0
SimpleRNN	BRTI	1.59	2.52	1.19	0.9683	1.0	1.0
	CTHO	1.67	2.80	1.20	0.9750	1.0	1.0
	HYEN	17.12	293.04	11.83	–0.7929	0.7684	0.5368

Table 3

Accuracy evaluation results with input data being n, e, u components

Model	Station	RMSE, mm	MSE, mm	MAE, mm	R-squared	F-score	Kapa
CNN(Conv1D)	BTRI	0.69	0.47	0.15	0.9940	0.9946	0.9892
	CTHO	1.60	2.56	0.73	0.9772	0.9892	0.9785
	HYEN	86.55	7,491.57	63.22	-44.8353	0.5210	0.0421
GRU	BRTI	0.91	0.83	0.72	0.9895	0.9892	0.9785
	CTHO	1.21	1.46	1.08	0.9870	0.9946	0.9892
	HYEN	13.23	175.02	11.29	-0.0708	0.8368	0.6737
Transformer	BRTI	6.45	41.55	5.05	0.4784	0.83330	0.6667
	CTHO	6.34	40.27	5.08	0.6412	0.8441	0.6881
	HYEN	8.94	79.96	7.74	0.5108	0.9053	0.8105
Seq2seq	BRTI	4.48	20.10	3.44	0.7477	0.9409	0.8817
	CTHO	5.44	29.60	2.82	0.7363	0.9140	0.8280
	HYEN	9.09	82.62	4.86	0.4945	0.8895	0.7789
SimpleRNN	BRTI	4.80	23.05	4.17	0.7107	0.9194	0.8387
	CTHO	3.73	13.95	3.01	0.8757	0.9032	0.8065
	HYEN	11.59	134.31	8.09	0.1782	0.8737	0.7473

ues of 1.31 and 0.65 mm, respectively. Additionally, the fit between the input data and the model reaches approximately 100 %. However, for data from the BTRI station, the CNN (Conv1D) model proves less effective, with an RMSE of 5.46 mm and an R^2 of 0.6254, which is relatively low.

Overall, when using only the Up-component as input data to predict this same value, the Seq2seq model proves more effective than the other models. However, compared to the results from existing studies [28], the achieved results here are still not high.

The data in Table 3 shows that when using the N, E, and Up components to predict the Up-component, the performance of all models is very poor. For the HYEN station, this data is unsuitable for all models, as indicated by the negative R-squared values for the CNN (Conv1D) and GRU models. For the remaining models, the R-squared values, which indicate the fit of the input data to the model, are very low, while error metrics such as MAE and RMSE are high. However, the CNN (Conv1D) and GRU models still show excellent predictive performance for the input data from the BTRI and CTHO stations.

From the data presented in Tables 2 and 3, it can be seen that the proposed models exhibit low predictive performance. Therefore, other approaches need to be tested to improve prediction accuracy with these models.

Solutions to Enhance Model Performance. To enhance the model's predictive performance, the proposed solutions include: 1) Using the CNN (Conv1D) model with 64 filters in the hidden layer, setting the batch size to 8 – PA1; 2) Integrating the RF model with the CNN (Conv1D) model. Prediction results with these solutions are presented in the following sections – PA2.

Using the CNN (Conv1D) model with 128 filters in the hidden layer, setting the batch size to 8.

From the data in Tables 2 and 3, it can be observed that by reducing the batch size to 8, the prediction performance of the CNN (Conv1D) model is significantly improved. The maximum RMSE value decreased from 5.46 to 1.49 mm, and the model's fit with the input data increased from 63 to 97 %. In this case, the maximum RMSE value is comparable to the results in study [28]; however, the MAE value is smaller (0.10 mm compared to 1.33 mm).

The graphs representing the loss function, predicted values on the training dataset, and test dataset in PA1 are illustrated in Figs. 11, 12, and 13.

Integration of the RF Model with the CNN (Conv1D) Model. The Random Forest (RF) model is an ensemble learning method used for both classification and regression tasks. It builds multiple decision trees during training and outputs the mode of classifications or the average prediction in regression. Each tree in the forest is trained on a random subset of the data, both in terms of samples and features, which helps reduce overfitting and improve generalization.

For a regression task, the prediction of RF is given by averaging the predictions of all individual trees

$$\hat{y} = \frac{1}{T}\sum_{t=1}^{T} f_t(x), \qquad (17)$$

where T is the total number of trees; $f_t(x)$ is the prediction from the t^{th} tree; x represents the input features.

Table 4

Prediction Results with the CNN (Conv1D) Model in PA1

Station	RMSE, mm	MSE, mm	MAE, mm	R-squared	F-score	Kapa
BTRI	0.15	0.02	0.03	1.0	1.0	1.0
CTHO	0.65	0.42	0.60	0.9962	1.0	1.0
HYEN	1.49	2.23	0.10	0.9863	0.9947	0.9895

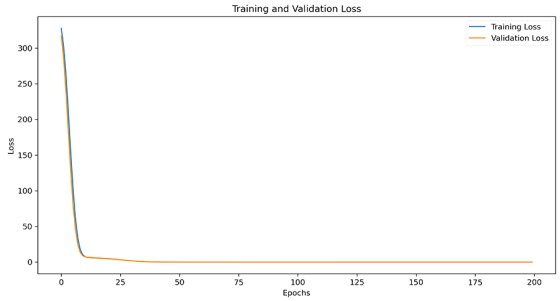

Fig. 11. Graph representing the loss function in PA1 of the HYEN station

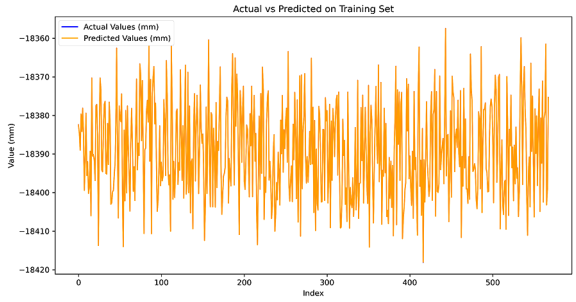

Fig. 12. Graph representing the predicted values on the training dataset of the HYEN station in PA1

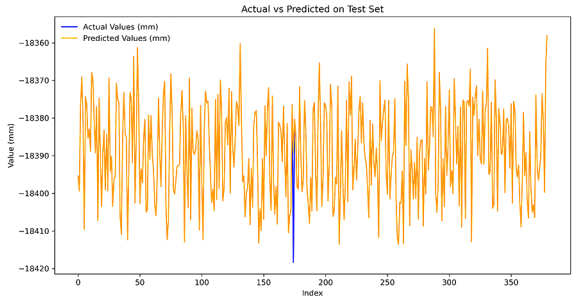

Fig. 13. Graph representing the predicted values on the testing dataset of the HYEN station in PA1

In classification, the final output is typically determined by a majority vote among the trees. The algorithm benefits from "bagging" (Bootstrap Aggregating), which provides robustness against overfitting by creating diversity among the trees. Additionally, Random Forest reduces variance compared to a single decision tree, making it highly effective for complex datasets with noisy data. However, it can become computationally expensive with large datasets due to the training of numerous trees.

The integration of Convolutional Neural Networks (CNNs), specifically the Conv1D variant, with Random Forest (RF) models produces the hybrid CNN(Conv1D)-RF model, harnessing the strengths of both for improved predictive accuracy. Conv1D layers in CNNs are effective at extracting localized patterns in sequential data, making them suitable for time-series or one-dimensional signals. However, CNNs alone can sometimes lack interpretability. RF models, known for their robustness and interpretability, utilize an ensemble of decision trees to improve accuracy and handle complex relationships, making them resilient to noise. By combining CNN's feature extraction with RF's decision-making, the CNN(Conv1D)-RF model gains the ability to classify or predict from sequential data with enhanced robustness and stability, balancing deep representation and reliable generalization.

The integrated model combining RF with CNN(Conv1D) is named CNN(Conv1D)-RF by the authors.

The prediction using the CNN(Conv1D)-RF model is conducted with three approaches, including:

1. Prediction with input data as the Up-component with a batch-size value of 16 (PA2.1).
2. Prediction with input data as N, E, and Up-component with a batch-size value of 16 (PA2.2).
3. Prediction with input data as N, E, and Up-component with a batch-size value of 8 (PA2.3).

The model performance statistics in Approach 2.1 are provided in Table 5.

From the data in Table 5, it can be seen that the performance of the CNN(Conv1D)-RF model with input data as the Up-component is very high, with maximum values of RMSE, MSE, and MAE being 0.4, 0.16 and 0.07 mm, respectively (for predictions on data from the BTRI station). The model's fit with all input data in this case reaches 100 %, demonstrating the effectiveness of the proposed model.

Figs. 14, 15, and 16 below illustrate the loss function values, predicted values on the training dataset, and predicted values on the test dataset of HYEN station in approach PA2.1.

The performance characteristics for the CNN(Conv1D)-RF model in approach PA2.2 are presented in Table 6.

From the data in Table 6, it can be seen that the CNN(Conv1D)-RF model still demonstrates high predictive performance for the data from the BTRI and CTHO stations. However, for the data from the HYEN station in this case, the predictive performance is poor.

The predictive performance of the CNN(Conv1D)-RF model in the PA2.3 approach is presented in Table 7.

From the data in Table 7, it can be seen that the predictive performance of the CNN(Conv1D)-RF model is lowest in this case when predicting data from the BTRI station, with RMSE = 0.92 mm, MSE = 0.85 mm, and MAE = 0.67 mm. The values of R-squared, Kappa, and F1-Score are all approximately 1. However, this still represents a very high prediction accuracy compared to previously published results [28, 29].

Conclusion. With the goal of identifying a suitable deep learning model for analyzing GNSS time series data, specifically the Up component, this paper evaluates the performance of five models: CNN(Conv1D),

Table 5

Performance of the CNN(Conv1D)-RF model in Approach PA2.1

Station	RMSE, mm	MSE, mm	MAE, mm	R-squared	F-score	Kapa
BTRI	0.40	0.16	0.07	0.9980	1.0	1.0
CTHO	0.09	0.01	0.06	1.0	1.0	1.0
HYEN	0.14	0.02	0.05	1.0	1.0	1.0

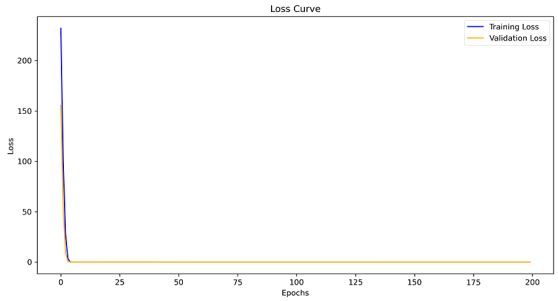

Fig. 14. Loss function values for predictions using the CNN(Conv1D)-RF model for HYEN station data in approach PA2.1

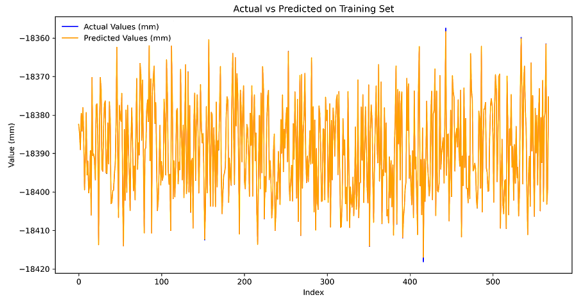

Fig. 15. Predicted values on the training dataset for HYEN station in approach PA2.1

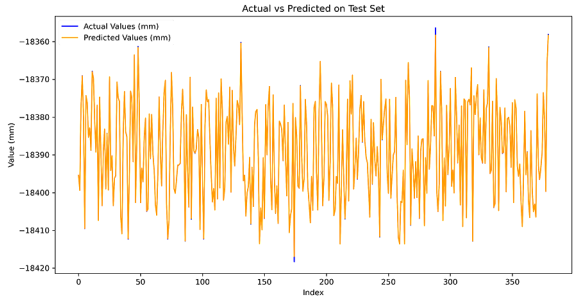

Fig. 16. Predicted values on the testing dataset for HYEN station in approach PA2.1

GRU, Transformer, Seq2seq, and SimpleRNN. The proposed models were tested based on data from three CORS stations in the VNGEONET network, established and operated by the Vietnam Department of Surveying, Mapping, and Geographic Information: HYEN, BTRI, and CTHO, with data collected from August 10, 2019, to March 18, 2022. The test results show the following characteristics:

1. When using the above models with the Adam optimization algorithm, mse loss function, and a batch size of 16, none of the models achieved the desired accuracy. In the case of the Up-component input data, the CNN(Conv1D) model showed poor performance with data from the BTRI station, and the GRU model performed poorly with data from the HYEN station. The SimpleRNN model showed negative R2 values for data from the HYEN station, indicating that this model is not suitable for the input data. The Transformer model is also not suitable for analyzing GNSS time series data. The Seq2seq model performed best in this case, with a maximum RMSE value of 2.41 mm; however, this is still higher than previously published results. With the same models as above, using input data that includes all three components include N, E, and Up, the results in poorer predictive performance compared to using only the Up-component as input data.

Table 6

Performance of the CNN(Conv1D)-RF Model in Approach PA2.2

Station	RMSE, mm	MSE, mm	MAE, mm	R-squared	F-score	Kapa
BTRI	1.23	1.50	0.54	0.9811	0.9946	0.9892
CTHO	0.12	0.02	0.07	1.0	1.0	1.0
HYEN	6.61	43.72	5.00	0.7325	0.8789	0.7579

Table 7

Predictive performance of the CNN(Conv1D)-RF model in the PA2.3 approach

Station	RMSE, mm	MSE, mm	MAE, mm	R-squared	F-score	Kapa
BTRI	0.92	0.85	0.67	0.9893	0.9946	0.9892
CTHO	0.49	0.24	0.43	0.9978	1.0	1.0
HYEN	0.36	0.13	0.27	0.9992	1.0	1.0

2. In the case of reducing the batch size to 8, when the input data consists only of the Up-component, the parameters characteristic of the model's predictive performance increase significantly. Specifically, in this case, the maximum RMSE value reaches 1.49 mm, which is comparable to previous publications. However, the MAE achieved in this case is very small (0.25 mm). When the input data includes N, E, and Up components, the model's performance decreases slightly, with the maximum RMSE value reaching 2.4 mm when predicting with the BTRI station dataset. Thus, it can be seen that a smaller batch size seems more suitable for analyzing the Up-component of the GNSS time series data.

3. The prediction results obtained from the CNN(Conv1D)-RF model indicate that the performance of the proposed new model is exceptionally high. Specifically, in the case of the lowest performance, which occurs when analyzing data from the BTRI station using the aforementioned model, the metrics characterizing the model's performance remain impressive, with RMSE = 0.4 mm, MAE = 0.07 mm, and R-squared = 99.8 %.

In the case where the input data consists of the components N, E, and Up-component with a batch size of 8, the predictive performance of the CNN(Conv1D) model decreases compared to the scenario where the input data includes only the Up-component. However, the results obtained in this case remain very high, with RMSE = 0.92 mm, MAE = 0.67 mm, and R-squared = = 98.9 %.

With the high performance achieved in predicting GNSS data over time, as demonstrated above, the pro-

posed CNN(Conv1D)-RF model shows promise for delivering effective results when applied in practical settings.

Acknowledgements. *Thanks to the Vietnam Department of Survey, Mapping, and Geographic Information (DOSM) for providing the data collected by CORS stations in the VNGEONET network to conduct this paper. Thanks to Dr. Ropesh Goyal from Department of Civil Engineering, Indian Institute of Technology (BHU), Varanasi, India for his valuable comments in the earlier version, which helped us to improve the manuscript's quality.*

Funding. *This work was financially supported by the Ministry of Natural Research and Environment in Vietnam under grant number TNMT.ĐL.2025.06.03.*

References.

1. Gill, S. K., Weston, N. D., & Smith, D. A. (2015). *NOAA guidance document for determination of vertical land motion at water level stations using GPS technology*, 18. https://doi.org/10.25607/OBP-147
2. Hu, B., Chen, J., & Zhang, X. (2019). Monitoring the Land Subsidence Area in a Coastal Urban Area with InSAR and GNSS. *Sensors*, *19*(14), 3181. https://doi.org/10.3390/s19143181
3. Pipitone, C., Maltese, A., Lo Brutto, M., & Dardanelli, G. (2023). A Review of Selected Applications of GNSS CORS and Related Experiences at the University of Palermo (Italy). *Remote Sensing*, *15*(22), 5343. https://doi.org/10.3390/rs15225343
4. Jones, C. E., An, K., Blom, R. G., Kent, J. D., Ivins, E. R., & Bekaert, D. (2016). Anthropogenic and geologic influences on subsidence in the vicinity of New Orleans, Louisiana. *Journal of Geophysical Research: Solid Earth*, *121*(5), 3867-3887. Portico. https://doi.org/10.1002/2015jb012636
5. Leveson, I. (2009). Socio-Economic Benefits Study: Scoping the Value of CORS and GRAV-D. Retrieved from https://repository.library.noaa.gov/view/noaa/55545
6. Ige, A. O., & Sibiya, M. (2024). State-of-the-Art in 1D Convolutional Neural Networks: A Survey. *IEEE Access*, *12*, 144082-144105. https://doi.org/10.1109/access.2024.3433513
7. Rizvi, S. M. H. (2021). Time Series Deep learning for Robust Steady-State Load Parameter Estimation using 1D-CNN. *Arabian Journal for Science and Engineering*, *47*(3), 2731-2744. https://doi.org/10.1007/s13369-021-05782-6
8. Weerakody, P. B., Wong, K. W., Wang, G., & Ela, W. (2021). A review of irregular time series data handling with gated recurrent neural networks. *Neurocomputing*, *441*, 161-178. https://doi.org/10.1016/j.neucom.2021.02.046
9. Bi, J., Zhu, Z., & Meng, Q. (2021). Transformer in Computer Vision. *2021 IEEE International Conference on Computer Science, Electronic Information Engineering and Intelligent Control Technology (CEI)*, 178-188. https://doi.org/10.1109/cei52496.2021.9574462
10. Ahmed, S., Nielsen, I. E., Tripathi, A., Siddiqui, S., Ramachandran, R. P., & Rasool, G. (2023). Transformers in Time-Series Analysis: A Tutorial. *Circuits, Systems, and Signal Processing*, *42*(12), 7433-7466. https://doi.org/10.1007/s00034-023-02454-8
11. Chen, Z., Ma, M., Li, T., Wang, H., & Li, C. (2023). Long sequence time-series forecasting with deep learning: A survey. *Information Fusion*, *97*, 101819. https://doi.org/10.1016/j.inffus.2023.101819
12. Goodfellow, I., Bengio, Y., Courville, A., & Bengio, Y. (2016). *Deep learning*, *1*(2). Retrieved from https://www.deeplearningbook.org/
13. Chollet, F., & Chollet, F. (2021). *Deep learning with Python. Simon and Schuster*. Retrieved from https://www.pressreader.com/books/b/deep-learning-with-python-23518
14. Siemuri, A., Kuusniemi, H., Elmusrati, M. S., Valisuo, P., & Shamsuzzoha, A. (2021). Machine Learning Utilization in GNSS – Use Cases, Challenges and Future Applications. *2021 International Conference on Localization and GNSS (ICL-GNSS)*, 1-6. https://doi.org/10.1109/icl-gnss51451.2021.9452295
15. Loli Piccolomini, E., Gandolfi, S., Poluzzi, L., Tavasci, L., Cascarano, P., & Pascucci, A. (2019). Recurrent neural networks applied to GNSS time series for denoising and prediction. *26th International Symposium on Temporal Representation and Reasoning*, 10-1. https://doi.org/10.4230/LIPIcs.TIME.2019.10
16. Kiani, M. (2020). A precise machine learning aided algorithm for land subsidence or upheave prediction from GNSS time series. *preprint arXiv:2006. 03772*. https://doi.org/10.48550/arXiv.2006.03772
17. Kiani, M. (2020). On GNSS residual position time series prediction and analysis using radial basis function networks machine learning. *Second international conference on Development of Materials engineering technology, mining, and geology*. Retrieved from https://www.researchgate.net/publication/344853610
18. Kiani, M. (2020). On the suitability of generalized regression neural networks for GNSS position time series prediction for geodetic applications in geodesy and geophysics. *arXiv preprint arXiv:2005.11106*. https://doi.org/10.48550/arXiv.2005.11106
19. Wang, J., Jiang, W., Li, Z., & Lu, Y. (2021). A New Multi-Scale Sliding Window LSTM Framework (MSSW-LSTM): A Case Study for GNSS Time-Series Prediction. *Remote Sensing*, *13*(16), 3328. https://doi.org/10.3390/rs13163328
20. Wen, T., Ou, G., Tang, X., Zhang, P., & Wang, P. (2021). A Novel Long Short-Term Memory Predicted Algorithm for BDS Short-Term Satellite Clock Offsets. *International Journal of Aerospace Engineering*, *2021*, 1-16. https://doi.org/10.1155/2021/4066275
21. Crocetti, L., Schartner, M., & Soja, B. (2021). Discontinuity Detection in GNSS Station Coordinate Time Series Using Machine Learning. *Remote Sensing*, *13*(19), 3906. https://doi.org/10.3390/rs13193906
22. Ruttner, P., Hohensinn, R., D'Aronco, S., Wegner, J. D., & Soja, B. (2021). Modeling of Residual GNSS Station Motions through Meteorological Data in a Machine Learning Approach. *Remote Sensing*, *14*(1), 17. https://doi.org/10.3390/rs14010017
23. Chen, H., Lu, T., Huang, J., He, X., Yu, K., Sun, X., Ma, X., & Huang, Z. (2023). An Improved VMD-LSTM Model for Time-Varying GNSS Time Series Prediction with Temporally Correlated Noise. *Remote Sensing*, *15*(14), 3694. https://doi.org/10.3390/rs15143694
24. Li, Z., Lu, T., Yu, K., & Wang, J. (2023). Interpolation of GNSS Position Time Series Using GBDT, XGBoost, and RF Machine Learning Algorithms and Models Error Analysis. *Remote Sensing*, *15*(18), 4374. https://doi.org/10.3390/rs15184374
25. Xie, Y., Wang, J., Li, H., Dong, A., Kang, Y., Zhu, J., Wang, Y., & Yang, Y. (2024). Deep Learning CNN-GRU Method for GNSS Deformation Monitoring Prediction. *Applied Sciences*, *14*(10), 4004. https://doi.org/10.3390/app14104004
26. Khorrami, F., Ejigu, Y. G., Näränen, J., Raja-Halli, A., & Nordman, M. (2024). Exploring Non-tidal Atmospheric Loading Deformation Correction in GNSS Time Series Analysis Using GAMIT/GLOBK Software. *International Association of Geodesy Symposia*. https://doi.org/10.1007/1345_2024_272
27. Jadon, A., Patil, A., & Jadon, S. (2024). A comprehensive survey of regression-based loss functions for time series forecasting. *Lecture Notes in Networks and Systems*, *998*, 117-147. https://doi.org/10.1007/978-981-97-3245-6_9
28. Wang, J., Nie, G., Gao, S., Wu, S., Li, H., & Ren, X. (2021). Landslide Deformation Prediction Based on a GNSS Time Series Analysis and Recurrent Neural Network Model. *Remote Sensing*, *13*(6), 1055. https://doi.org/10.3390/rs13061055
29. Gao, W., Li, Z., Chen, Q., Jiang, W., & Feng, Y. (2022). Modelling and prediction of GNSS time series using GBDT, LSTM and SVM machine learning approaches. *Journal of Geodesy*, *96*(10). https://doi.org/10.1007/s00190-022-01662-5

Емпіричне порівняння п'яти архітектур глибокого навчання для прогнозування часових рядів GNSS-даних

Тронг Гія Нгуєн[1,2], *Тін Дюк Ле*[1], *Лонг Куок Нгуєн*[*1,3]

1 – Ханойський університет гірничої справи та геології, Факультет геоматики та землеустрою, м. Ханой, Соціалістична Республіка В'єтнам
2 – Ханойський університет гірничої справи та геології, Група дослідників із геодезії та навколишнього середовища, м. Ханой, Соціалістична Республіка В'єтнам
3 – Ханойський університет гірничої справи та геології, Група дослідників із інновацій для сталого та відповідального гірничого господарства (ISRM), м. Ханой, Соціалістична Республіка В'єтнам
[*] Автор-кореспондент e-mail: nguyengiatrong@humg.edu.vn

Мета. Це дослідження має на меті запропонувати ефективну модель глибокого навчання для аналізу часових рядів GNSS-даних.

Методика. Було досліджено п’ять моделей глибокого навчання з використанням алгоритму оптимізації Adam і функції втрат MSE: CNN (Conv1D), GRU, Transformer, Seq2seq та SimpleRNN. Ефективність прогнозування моделей була оцінена на основі даних із трьох CORS-станцій у В’єтнамі: HYEN, BTRI і CTHO. Моделі були розроблені з використанням функції оптимізації Adam, функції втрат MSE й розміру пакету 16.

Результати. Загалом результати прогнозування для всіх моделей були низькими. Зокрема, модель Seq2seq досягла найнижчого значення RMSE для даних компонента Up зі станції HYEN. Модель Transformer виявилася непридатною для аналізу GNSS-даних у часовому вимірі, а модель SimpleRNN показала слабку ефективність на даних станції HYEN. Водночас запропонована модель CNN(Conv1D)-RF, що інтегрує модель RF із CNN(Conv1D), продемонструвала високу точність прогнозування: RMSE = 0,4 мм, MAE = 0,07 мм і коефіцієнт детермінації R^2 = 99,8 % при використанні лише компонента Up як вхідних даних. При включенні всіх трьох компонентів (N, E і Up) максимальне значення RMSE становило 0,92 мм.

Наукова новизна. У цьому дослідженні представлена модель CNN(Conv1D)-RF – нова інтеграція моделі RF із CNN(Conv1D) для аналізу GNSS-даних.

Практична значимість. Запропонована модель демонструє високу точність прогнозування, що свідчить про її потенційне застосування для аналізу точності GNSS-даних у різних практичних сферах.

Ключові слова: *глибоке навчання, часові ряди GNSS, машинне навчання, алгоритм*

The manuscript was submitted 21.02.25.